Cinelerra GG

The Comprehensive

User Manual

professional video editing

Video editing for ambitious users

Cinelerra-GG Infinity

The Comprehensive User Manual

November 21, 2025

https://www.cinelerra-gg.org

Cinelerra-GG – The Comprehensive User Manual

Maintained and published by Phyllis Smith (phylsmith2017@gmail.com)

Copyright © 2023 The CINELERRA-GG Community

Please report errors in the book to our bug tracker or the user forum

Graphic cover page: SAM, License CC BY 3.0

Cover design: The CINELERRA-GG Community

Illustrations: The CINELERRA-GG Community

Legal information

This manual adheres to the GNU FDL (GFDL) copyleft license for free documentation, designed by the Free Software Foundation (FSF) for the GNU Project. It provides the rights to copy, redistribute, and modify (except for "variant sections") a work and requires all copies and derivatives to be available under the same license. The GFDL was designed for manuals, textbooks, other reference and instructional materials, and documentation which often accompanies GNU software. For more detailed information please refer to GNU Free Documentation License

Disclaimer

In this capacity, the author of this manual is not responsible for the function or errors of the software described in this manual. Great care has been taken in the creation of texts and illustrations. Nevertheless, errors cannot be completely excluded.

However, the author cannot assume any legal responsibility or liability for incorrect information and its consequences. The author is grateful for suggestions for improvement and information on errors.

Trademarks

In this manual, trade names are used without the guarantee of free usability and without special marking. However, it can be assumed that many of the trade names are registered trade names or are to be regarded as such at the same time.

Links

Many links in this PDF are active. If the pointer points to a link, your PDF reader should additionally display the actual destination address. This way you can check if the target matches the link description before clicking.

Introduction

CINELERRA-GG is a software program NLE, Non-Linear Editor, that provides a way to edit, record, and play audio or video media. It can also be used to transcode media from one format to another or to correct and retouch photos. CINELERRA-GG currently runs on many Linux operating system distributions.

This manual covers the CINELERRA-GG INFINITY version. Information contained in this manual is a description of the CINELERRA-GG program usage and was obtained from various sources to include different communication channels, emails, common knowledge, and write-ups as new features were added. The origin of this manual has always been present in the CINELERRA-GG source code in the doc subdirectory as cinelerral.html and can be found at: Secrets of Cinelerra by the original author Adam Williams. Over time his manual was enhanced by Alex Ferrer with Adam's and Alex's work becoming the baseline for the CV manual and subsequently that combined work was mostly incorporated into this manual. Interestingly, starting from Secrets of Cinelerra which is about 122 full pages, to the CV manual of about 212 pages at the time it was incorporated here, this manual has now grown to over 780 pages. More information about the manual and source code contributions can be found here: Authors and Contributors.

The author of the original CINELERRA program, Adam Williams, as well as many different people have modified the CINELERRA source code over the years. William Morrow used Adam's source code as the baseline and merged all of the known changes at the time, whenever possible, into this CINELERRA-GG version and had been adding numerous features, updates, and fixes.

NOTE1: This manual is a Work in Progress: monthly parts are added to cover the new features introduced in CINELERRA-GG. The texts are continuously revised to make them clearer and more complete. In order to have a better and better product we rely on the suggestions and criticisms of our readers. Do not hesitate to let us know your opinions. For this continuous variability, we recommend that you download the manual at least once a month, during the release of CINELERRA-GG, so that its content is always up to date.

NOTE2: this manual, in the HTML version present on the website, is the base of the *Context Help* that can be called inside the program. For each window, button and plugins we can open the relative web page of the manual with `ALT/h` keys. More details in Help and Context Help.

The CINELERRA-GG software is under GPLv2+ license. A notice of this is included

in the software and shown here.

*
*CINELERRA
*Copyright (C) 1997-2012 Adam Williams <broadcast at earthling dot net>
*
*This program is free software; you can redistribute it and/or modify
*it under the terms of the GNU General Public License as published by
*the Free Software Foundation; either version 2 of the License, or
*(at your option) any later version.
*
*This program is distributed in the hope that it will be useful,
*but WITHOUT ANY WARRANTY; without even the implied warranty of
*MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
*GNU General Public License for more details.
*
*You should have received a copy of the GNU General Public License
*along with this program; if not, write to the Free Software
*Foundation, Inc., 59 Temple Place, Suite 330, Boston

CINELERRA-GG Overview

Presented briefly here is an overview of CINELERRA-GG INFINITY and information provided in this manual. The GG version of CINELERRA has been improved for *stability*, *modernized* to accommodate the *current state* of Linux software, enhanced with additional *basic features*, and enriched with *new features* imagined by dedicated users and then implemented by professional programmers.

Website https://www.cinelerra-gg.org
: The website for the cinelerra-gg software is a good place to start for information, help, and documentation. It is professionally maintained and continuously updated with more language translations ongoing.

Stability
: Software programs that *just work* are a #1 priority in order to be of use for producing quality videos. A large amount of time has been invested in debugging problems and resolving crashes. And in a continuous process to do so, a chapter on Troubleshooting is included in order to easily provide sufficient information for users to capture issues and crashes so that they can be analyzed and quickly fixed to avoid repeat problems.

Modernization
: Artistic creativity has been applied to modernize the CINELERRA-GG plugin icons for video and audio to even include the ffmpeg plugins. The Cinfinity

set of plugin icons come in square or roundish versions — your choice. In keeping up with the current expectation of users for a certain *look and feel*, 2 very modern recent theme additions, Cakewalk and Neophyte, provide alternatives to the already existing 9 themes. These 11 themes give the user the choice to get the look they like best for their own eyes.

Current and up-to-date

For today's software, included thirdparty libraries are kept up to date in a timely manner and effort is made to always use a relatively recent version of FFmpeg, if not the latest. This is a big deal because there is a whole set of separate programmers working on these libraries continuously and diligently to cover all of the old and newly created formats. Thus CINELERRA-GG programmers can be dedicated to working solely on CINELERRA-GG rather than just trying to keep up with many formats.

FFmpeg usage integration

By using FFmpeg with CINELERRA-GG, there is the advantage that users can directly convert videos via pre- and post-processing, without the need for command lines to be executed manually before or afterwards.

Import and Export formats

Listed here are only a few of the supported Import and Export formats:

- several standard native formats, such as mpeg, ac3, flac, exr and jpeg/png/ppm/tiff sequences
- FFmpeg's over 400 decoders and 150 encoders accessible from within CINELERRA-GG to include: mp4, mkv, mpeg, mov, m2ts, mp3, dvd, ogg, theora, prores, tiff, webm, flac, opus, vorbis, quicktime (div, dnxhd, jpeg, mjpeg, mp4, rle, v308, v410), h264 & h265 usage, avc, hevc, and recently released AV1 and WebP
- raw image format for over 700 supported cameras, courtesy Dave Coffin's DCraw

Standard Features

- Program window for video and audio tracks, navigation, popups, playing and seeking functions.
- Editing via track manipulation with either drag and drop editing or cut and paste editing.
- Patchbay tools: fader, gang, arm, enable output, expanders, mixers, and blending operations.
- Undo and Redo capability for many editing functions.

- Drag handle functionality for Ripple, Roll, Slip, Slide, Edge, and No Effect.
- Dynamic Keyframe support with curve, toggle, automatic, compositor, and editing capabilities.
- Proxy editing support to speed up editing for large formatted files on slower computers.
- Compositor window with masking, zooming, cropping, projector and camera capabilities.
- Viewing window for quickly viewing/playing audio, video, clips, or proxies.
- Resources window with Media, Proxy, Clips, Video and Audio Effects/-Transitions.
- User defined bins/folders for a project.
- Expandable Video/Audio effects for grouping selections such as Color Correction or Motion plugins.
- Included are over 400 Plugins – audio and video native, FFmpeg, plus expandable to include others.
- LV2 and Ladspa plugins audio support.
- Mouse over Media or Proxy folders to get a 5 second real time preview of the media.
- Single or multiple files rendering; background rendering, batch rendering, command line rendering, and use of a Render Farm to take advantage of multiple computer CPUs.
- Numerous pre-defined output formats automatically available and allowance for user-defined formats.
- Capture and Recording capability to include Broadcast TV recording, editing, and viewing.
- Hundreds of Shortcuts are defined which are easily viewed using the shell commands pulldown.
- Context Help in the program for numerous window, menu, GUI elements, buttons, and other items via the user's configured web browser.
- PorterDuff operations are available in the patchbay of the main timeline window for alpha blending.
- Color correction + 8-bit and 10-bit color space.
- Up to 8K video supported.
- ShuttlePro and ShuttleXpress functionality.
- International language support.

- Multi-screen, Multi-viewers, Multi-session, Multi-pane capability.
- Menu bar shell commands with reference to some manuals and user-definable additions.
- Video stabilization, Motion tracking, Motion Graphics.
- Blade Cut/Hard Edges, Snap/Cut editing and Snap Dragging.
- OpenGL support as well as Direct X11 speedup.
- Advanced Trim Features and Grouping.
- Asset and Title colors selection including Alpha slider adjustment.
- Scaling preference for any monitor size and HiDPI to support 4K monitors.

Innovative New Features

- Inter-View mode as invented through a collaborative effort by a user and the developer.
- Proxy settings for large formatted media with the coveted Scalar option and proxy quick switch. Includes a 1:1 proxy setting used to save CPU time with a reduced bitrate.
- Multiple cameras / Mixer Viewer with the number of cameras only limited by the resources on the user's computer (on a large system, 50 are known to work).
- Audio/Video sync via waveforms, or timecodes to include syncing with externally recorded audio.
- DVD/bluray creation, editing, and copying for non-commercial media greatly enhanced for usability.
- Title plugin virtually unlimited script size with many changeable attributes such as size, blink, color.
- Motion Graphics using the Sketcher plugin to create elements such as ellipses, rectangles and shapes for simpler motion graphics.
- Open EDL for editing clips, nested clips, and xml files while working on a Project.
- File by Reference; To be able to modify the original source during editing (which in this case, becomes destructive).
- The Vectorscope option in the Videoscope plugin allows for the use of any number of user-supplied grid patterns as an Overlay.
- Hardware acceleration with vaapi and vdpau for computers with graphics hardware meeting certain criteria.

- Bump Autos is a new type of keyframe automation that allows you to easily create and manage keyframes intervals in Speed, Fade and Camera/Projector-XYZ curves.
- One of the only Linux NLEs that handles and renders RGB, 444, 422 and 420 at 8, 10 and more-bit footage as opposed to working in only 8-bit (which introduces some artifacts like banding); both in Full Range Color and limited Broadcast Range Color.
- The Blend Algebra/Blend Program plugins provide the advanced feature of combining and modifying color pixels and transparencies of several tracks according to a mathematic algorithm written entirely by the user in the form of compact and simple piece of code. Such user defined algorithms (blend functions) are compiled, dynamically linked and immediately executed inside Cinelerra on the fly without the need to restart the application, reload the project, reattach plugins, etc.

NOTE: In CINELERRA-GG the editing workflow is different from the workflow used by other NLEs. It is less intuitive and requires us to think first about what we want to achieve. In fact, it is a workflow based on the tracks in their entirety and not on individual clips. See workflow is different! for a comparison with the Adobe Premiere Pro workflow.

Chapters Overview

Below is listed a brief overview of each chapter to help you to decide which chapters you should pay the most attention to and read more thoroughly. And which chapters are important for beginning to learn to use CINELERRA-GG. At the end of the descriptions is a list of the sections to read for beginners.

Chapter 1 Installation.

If you just want to get started using the program, you can safely skip this chapter and instead go to: `https://www.cinelerra-gg.org` and simply download a pre-built linux version for your Operating System. If you would like to do your own builds so that you always have the latest, refer to this chapter to learn how. But if you are already familiar with doing your own builds, you can just refer to this chapter when you encounter issues.

Chapter 2 Interface (the 4+ Windows).

It is important to understand the window setup in CINELERRA-GG and what each of them is used for. In addition this chapter covers some basic navigation functions that you will need to know. However, if you have used an NLE before and don't mind experimenting to learn how to execute certain functions, probably there is no need to peruse this chapter.

Chapter 3 Project and Media Attributes.

This chapter is helpful for basic understanding of setting up your session but it can easily be skipped.

Chapter 4 Load, Save, and the EDL.

The EDL is the list of changes that CINELERRA-GG would make to the original media in order to produce the desired output Since this is important to not losing your work, you should read this chapter for some basic usage concepts and for some lesser used functions that may come in handy. Besides how to Load and Save files, there is also information on using raw camera formats. Helpful hints on working with image sequences, such as a bunch of pictures from your camera all loaded at once, is a time saver. Understanding that the EDL is the list of changes that would be made to the original media in order to produce the desired output is key to ensuring that that media remains intact.

Chapter 5 Editing.

New and occasional users will find it necessary to read this chapter. However, you can decide which editing mode you prefer and concentrate on reading that section. Even seasoned users will want to read this chapter in order to take advantage of features that have been newly implemented or may be less familiar due to infrequent usage. Anyone who wants the editing capabilities that comes with use of the Shuttle Jog Wheel, will initially need to read that section to become familiar with its usage.

Chapter 6 Advanced Editing.

Many tools for the advanced editing: Proxy, Nesting; Trimming; Multi-Cam; Audio/Video sync and Transcoding. CINELERRA-GG offers many possibilities and sophisticated alternatives for editing, but they are complex tools. Here we deepen their knowledge and use.

Chapter 7 Rendering.

Minimally most users will need to read the section on single file rendering to get started; since after all the whole purpose of using an NLE is to create your own media. The end of this chapter includes some helpful specific rendering scenarios which could prove quite useful. You might also want to consider using batch rendering and background rendering for performance reasons. And, of course, render farm usage to take advantage of multiple computers can be a real boost.

Chapter 8 Keyframes.

This is a more advanced topic but extremely useful to know. Keyframes are data values that have been associated to media on the timeline that affect

the media presentation. You can go for years without knowing all of the nuances but you will most likely have a need to use and understand them sooner or later.

Chapter 9 FFmpeg Interactions.

Skip this chapter if you are just getting started. Skip this chapter if you just want to use common formats that are already set up for you. Experts may want to refer to this chapter in order to set up their own option files.

Chapter 10 Plugins.

Use this chapter as a reference to add an audio or video plugin in order to correct color or add some fancy effect. You will not want to read about each and every plugin, most of which you will never use. On the other hand, going over the section on "some specific details concerning plugins" might come in handy.

Chapter 11 Transition Plugins.

Everyone who does not know about using transitions should read this chapter because you will want to use transitions between cuts in your video for smoothly changing scenes. It is short and easy reading.

Chapter 12 Overlays Modes, Alpha Blending, and Porter Duff.

Experts can really use the information in this chapter to spiff up their editing work through some blending type techniques. A lot of information and possibilities are described here.

Chapter 13 Capturing and Recording Media.

This chapter is helpful in learning how to capture and record media from various sources. A section on using CINELERRA-GG with broadcast TV (NTSC) is especially useful.

Chapter 14 DVD and Bluray Creation.

If this is what you want to do, then read this chapter. In the case of DVD media creation it includes many examples and lots of tips for checking your methods.

Chapter 15 Shortcuts.

Everyone will want to use this chapter to speed up their editing sessions. Once you find yourself doing the same thing over and over again, you will want to know if there is a shortcut. Then look here!

Chapter 16 Configuration, Settings and Preferences.

Refer to this chapter when you want to change some setting or preference. Otherwise if a new user, you should make sure to read the first page which includes some important basic information.

Chapter 17 How some stuff works.

Only read this chapter if you are confused about a specific covered topic and how it works.

Chapter 18 Troubleshooting and Help.

Use this chapter for diagnosing a problem and find out what to report to get the best resolution or help. Context Help, using Alt/h, is explained here.

Chapter 19 Performance and other Tips.

There are a few performance tips in this chapter that may eventually prove useful.

Chapter 20 Translations.

If you want to help by providing translations for a specific language, there is a complete description of how to do this and what to do.

Chapter 21 Licenses.

Appendix A QuickStart.

Here you go! If you hate to read, just go over the quick start guide or the youtube video creation and simply be on your way.

Appendix B Developer's Section.

Some extra information that developers may or may not find useful.

Appendix C Auxiliary Programs.

This is a catch-all for any useful programs that may need to be included, mostly for analysis.

Appendix D Editing: real-world usage cases.

Our users illustrate in-depth examples of their workflows by showing their editing in real use cases.

In summary, *must* reads for a new user would be these chapters or sections:

- Chapter 2 — Interface (the 4+ Windows).
- Chapter 4 — Load, Save, and the EDL.
- Chapter 5 — Editing; read all sections except emphasize only either Cut and Paste Editing or Drag and Drop Editing to suit your purpose and then skim the editing mode that is not your preference as some operations work in either mode. Skip the ShuttlePROv2 and ShuttleXpress Jog Wheels for Editing section unless you have this jog wheel in hand.

- Chapter 7 — Rendering; minimally read the Single File Rendering section.
- Chapter 10 — Plugins; read the section on How to Use Plugins.
- Chapter 11 — Transition Plugins.
- Chapter 16 — Configuration, Settings and Preferences; read at least the first couple of paragraphs.

Contents

Introduction i

Contents xi

1 Installation 1
- 1.1 CINELERRA-GG AppImage . . . 1
- 1.2 Download Already Built CINELERRA-GG . . . 3
- 1.3 How to Build CINELERRA-GG from Developer's Git Repository . . . 9
- 1.4 Building the HTML Manual for Context Help . . . 22
- 1.5 Windows 10 with Cygwin for CINELERRA-GG Limited . . . 24
- 1.6 Android Tablet or Phone with TERMUX . . . 28
- 1.7 Pre-built Packages and Distros with CINELERRA-GG Included . . . 31

2 Interface (the 4+ Windows) 33
- 2.1 Program Window . . . 34
- 2.2 Compositor Window . . . 50
- 2.3 Viewer Window . . . 73
- 2.4 Options in both the Compositor and Viewer Windows . . . 75
- 2.5 Resources Window . . . 77
- 2.6 Other Options and Other Windows . . . 94

3 Project and Media Attributes 100
- 3.1 Audio attributes . . . 101
- 3.2 Video attributes . . . 102
- 3.3 Best practice in pre-editing . . . 105

4 Load, Save, and the EDL 108
- 4.1 EDL — Edit Decision List . . . 108
- 4.2 Supported File Formats . . . 108
- 4.3 Loading Files . . . 113
- 4.4 Saving Your Work . . . 121

5 Editing 127
- 5.1 The Patchbay . . . 128
- 5.2 Manipulating Tracks . . . 133
- 5.3 Two Screen Editing . . . 137
- 5.4 Cut and Paste Editing . . . 140
- 5.5 Drag and Drop Editing . . . 143
- 5.6 Inter-View Mode – Identifying Source Targets . . . 153

5.7 Edit Tools . . . 155
5.8 Multi-Session . . . 158
5.9 Multi-Viewer Window Support . . . 159
5.10 ShuttlePROv2 and ShuttleXpress Jog Wheels for Editing . . . 159

6 Advanced Editing 170
6.1 Proxy . . . 170
6.2 Transcode . . . 178
6.3 OpenEDL . . . 179
6.4 File by Reference . . . 182
6.5 Trimming . . . 183
6.6 Nesting clips and assets . . . 189
6.7 Multi-Camera / Mixer . . . 192
6.8 Audio/Video sync using Waveforms/Timecodes . . . 200
6.9 Subtitles . . . 209

7 Rendering 212
7.1 Single File Rendering . . . 212
7.2 Some Specific Rendering . . . 218
7.3 About Image Sequences . . . 227
7.4 Data storage formulas . . . 228
7.5 Batch Rendering . . . 228
7.6 Background Rendering . . . 235
7.7 Render Farm Usage . . . 237

8 Keyframes 249
8.1 Automation Keyframes / Autos . . . 249
8.2 Using Autos . . . 251
8.3 Speed / Fade Automation Usage and Auto Gang . . . 253
8.4 Bump autos . . . 255
8.5 Plugin Keyframes . . . 258
8.6 Default Keyframe . . . 258
8.7 Keyframe *Edit Params* for Plugins . . . 259
8.8 Generate Keyframes while Tweaking / Automatic Keyframe Mode 260
8.9 Compositor Keyframes . . . 260
8.10 More about Editing Keyframes . . . 261
8.11 Allow Keyframe Spanning . . . 261

9 FFmpeg Interactions 263
9.1 FFmpeg Early Probe Explanation . . . 263
9.2 How to Create FFmpeg Options Files . . . 264
9.3 The FFmpeg Image2 Streams . . . 272
9.4 Raw Input Opts File for Video/Audio . . . 273
9.5 FFmpeg Items of Note . . . 274

10 Plugins **275**
10.1 How to Use Plugins . . . 276
10.2 Editing Effects . . . 278
10.3 Copy/Paste Effects . . . 279
10.4 Shared Effects and Shared Tracks . . . 280
10.5 Saved Plugin Presets . . . 281
10.6 Some specific details concerning Plugins . . . 282
10.7 Audio Effects - Native . . . 290
10.7.1 AudioScope . . . 290
10.7.2 Chorus . . . 290
10.7.3 Compressor (Single Band) . . . 291
10.7.4 Compressor Multi (Multi Band) . . . 292
10.7.5 DC Offset . . . 294
10.7.6 Delay Audio . . . 294
10.7.7 DeNoise . . . 294
10.7.8 DenoiseFFT . . . 294
10.7.9 Despike . . . 295
10.7.10 EQ Graphic . . . 295
10.7.11 EQ Parametric . . . 296
10.7.12 Echo . . . 296
10.7.13 EchoCancel . . . 297
10.7.14 Flanger . . . 298
10.7.15 Freeverb . . . 299
10.7.16 Gain . . . 299
10.7.17 Interpolate . . . 299
10.7.18 Invert Audio . . . 299
10.7.19 Live Audio . . . 299
10.7.20 Loop Audio . . . 300
10.7.21 Overlay (Audio) . . . 300
10.7.22 Pitch Shift . . . 300
10.7.23 Remove Gaps . . . 300
10.7.24 ResampleRT . . . 300
10.7.25 Reverb . . . 300
10.7.26 Reverse Audio . . . 302
10.7.27 SoundLevel . . . 302
10.7.28 Spectrogram . . . 302
10.7.29 Synthesizer . . . 302
10.7.30 Time Stretch RT . . . 302
10.7.31 Tremolo . . . 303
10.8 Audio Ladspa Effects . . . 303
10.9 Audio LV2 / Calf Plugins . . . 304
10.10 Video Effects — Native . . . 307
10.10.1 1080 to 480 . . . 307

10.10.2 1080 to 540 . . . 308
10.10.3 Aging TV . . . 308
10.10.4 Alpha . . . 308
10.10.5 Auto Scale . . . 308
10.10.6 Blend Algebra and Blend Program . . . 309
10.10.7 Blue Banana . . . 322
10.10.8 Blur . . . 331
10.10.9 BoxBlur . . . 332
10.10.10 Brightness/Contrast . . . 332
10.10.11 BurningTV . . . 333
10.10.12 C41 . . . 333
10.10.13 Chroma Key . . . 336
10.10.14 Chroma Key (Avid) . . . 337
10.10.15 Chroma Key (HSV) . . . 342
10.10.16 Color 3 Way . . . 343
10.10.17 Color Balance . . . 345
10.10.18 ColorSpace . . . 345
10.10.19 Color Swatch . . . 346
10.10.20 CriKey . . . 346
10.10.21 Crop & Position . . . 349
10.10.22 DeScratch . . . 350
10.10.23 Decimate . . . 351
10.10.24 Deinterlace . . . 352
10.10.25 Deinterlace-CV . . . 352
10.10.26 Delay Video . . . 352
10.10.27 Denoise Video . . . 352
10.10.28 Difference key . . . 353
10.10.29 DotTV . . . 354
10.10.30 Downsample . . . 354
10.10.31 Edge . . . 354
10.10.32 Fields to frames . . . 354
10.10.33 Flip . . . 354
10.10.34 Foreground . . . 355
10.10.35 Frames to fields . . . 355
10.10.36 Freeze Frame . . . 356
10.10.37 Gamma . . . 357
10.10.38 Gradient . . . 358
10.10.39 HistEQ . . . 359
10.10.40 Histogram . . . 360
10.10.41 Histogram Bezier / Curves . . . 365
10.10.42 HolographicTV . . . 368
10.10.43 Hue saturation . . . 368
10.10.44 Interpolate Bayer . . . 368

10.10.45 Interpolate Video . . . 368
10.10.46 Inverse Telecine . . . 369
10.10.47 Invert Video . . . 370
10.10.48 Lens . . . 370
10.10.49 Linear Blur . . . 371
10.10.50 Live Video . . . 371
10.10.51 Loop video . . . 372
10.10.52 Mirror . . . 372
10.10.53 Motion51 . . . 374
10.10.54 Motion . . . 377
10.10.55 Motion 2 Point . . . 388
10.10.56 Motion Blur . . . 389
10.10.57 Oil painting . . . 389
10.10.58 Overlay (Video) . . . 389
10.10.59 Perspective . . . 390
10.10.60 Polar . . . 392
10.10.61 Posterize . . . 392
10.10.62 RGB-601 . . . 392
10.10.63 RGBShift . . . 392
10.10.64 Radial Blur . . . 393
10.10.65 ReframeRT . . . 394
10.10.66 Reroute . . . 396
10.10.67 Reverse video . . . 396
10.10.68 Rotate . . . 396
10.10.69 Rumbler . . . 397
10.10.70 SVG via Inkscape . . . 397
10.10.71 Scale . . . 398
10.10.72 Scale Ratio . . . 398
10.10.73 Selective Temporal Averaging . . . 399
10.10.74 Sharpen . . . 400
10.10.75 Shift Interlace . . . 401
10.10.76 Sketcher . . . 401
10.10.77 Speed PerCent . . . 404
10.10.78 Sphere Cam . . . 404
10.10.79 Swap Frames . . . 404
10.10.80 Swap channels . . . 404
10.10.81 Threshold . . . 405
10.10.82 Time average . . . 405
10.10.83 TimeBlur . . . 406
10.10.84 Timefront . . . 406
10.10.85 Timelapse Helper . . . 407
10.10.86 Title . . . 407
10.10.87 Tracer . . . 416

10.10.88 Translate . . . 418
10.10.89 Unsharp . . . 419
10.10.90 Videoscope . . . 419
10.10.91 Wave . . . 426
10.10.92 Whirl . . . 426
10.10.93 YUV . . . 426
10.10.94 YUV411 . . . 426
10.10.95 YUVShift . . . 426
10.10.96 Zoom Blur . . . 427
10.11 CUDA plugins . . . 428
10.12 OpenCV plugins . . . 429
10.13 FFmpeg Audio and Video Plugins . . . 440
10.14 Rendered Effects . . . 458

11 Transition Plugins **461**
11.1 Audio Transitions . . . 463
11.2 Video Transitions . . . 464

12 Overlays **467**
12.1 Math algorithms . . . 470
12.2 Description . . . 472
12.3 Alternative Views of Overlay Modes . . . 476
12.4 Problems with transparencies . . . 479

13 Capturing and Recording Media **481**
13.1 Record Web Media in real-time . . . 482
13.2 Digital Video Broadcasting (DVB) . . . 487
13.3 The commercial DB . . . 501
13.4 Transfer VHS/DVD Media or Video8/Hi8 Tapes into CINELERRA-GG 503

14 DVD and Bluray Creation **509**
14.1 Output Terminal Messages from Creating DVDs . . . 521
14.2 Debugging DVDs Creation . . . 524
14.3 DVD-Subtitles . . . 527
14.4 Dvd Interlaced Chroma . . . 527
14.5 DVD with LPCM or MP2 audio . . . 528
14.6 MPEG utility programs . . . 529
14.7 Creating Blu-ray Without Re-encoding . . . 530
14.8 Blu-ray Workaround for Mount/Umount . . . 532
14.9 Blu-ray from Multiple CINELERRA-GG Output . . . 532
14.10 Use Case: DVD . . . 534

15 Shortcuts **540**
15.1 Main window . . . 540

15.2 Compositor window shortcuts 551
15.3 Viewer window shortcuts 554
15.4 Resources window Keys 556
15.5 Other windows 557
15.6 Alternative Shortcuts 560

16 Configuration, Settings and Preferences **563**
16.1 .bcast5 563
16.2 Playback A / Playback B 564
16.3 Recording 570
16.4 Performance 572
16.5 Interface 574
16.6 Appearance 576
16.7 About 581
16.8 Environment Variables for Customization 582

17 How some stuff works **585**
17.1 Copy/Paste and Highlight Usage 585
17.2 Playing is Different than Seeking/Positioning! 587
17.3 Color Space and Color Range Affecting Playback 590
17.4 Automatic "Best Model" Media Load 591
17.5 Simple Animation (Festival) 592
17.6 Textbox Non-std Character / Unicode Insertion 593

18 Troubleshooting and Help **594**
18.1 Help and Context Help 594
18.2 Troubleshooting 596
18.3 What to Include in Problem Reports 597
18.4 Crash Dumps for Analysis 598
18.5 When things go wrong 599
18.6 Menu Bar Shell Commands 606
18.7 CINELERRA-GG Command Line -h 609

19 Performance and other Tips **610**
19.1 Hardware video acceleration 611
19.2 Optimized Playback – X11 Direct 623
19.3 Proxy Settings and Transcode 623
19.4 Cache size and Seconds to preroll render 624
19.5 Tips for Improving Smaller Computers Use 624
19.6 General Crash Handling Tips 625
19.7 Tips for Specific Operations 627

20 Translations **636**

21 Licenses **640**

22 Authors and Contributors **642**

Appendices **647**

A QuickStart **648**
A.1 CINELERRA-GG Quick Start Guide 648
A.2 Overview on Formats and Codecs 657
A.3 Overview on Color Management 663

B Developer's Section **670**
B.1 How Builds Really Work and Even More Options 670
B.2 Experimental Builds 673
B.3 Configuration Features 674
B.4 Thirdparty Parallel Build 677
B.5 Using the very latest Libraries 678
B.6 Find Lock Problems with Booby Trap 680
B.7 Valgrind Support Level 681
B.8 CFLAGS has -Wall 683
B.9 Prof2 – A Profiler 683
B.10 Working on AppImage 687
B.11 How to Create a new Theme 692
B.12 How Context Help works in the Program Code 694
B.13 Unique Blend plugins workflow 699

C Auxiliary Programs **704**
C.1 Using Ydiff to check results 704
C.2 Image Sequence Creation 705
C.3 Details about .bcast5 Files 706
C.4 Focusing the 4 main windows as a group 707

D Editing: real-world usage cases **708**
D.1 Workflow with OpenEDL and Nested Clips 708
D.2 Workflow with multi-cam and external audio 712
D.3 Workflow with Keyframes and Plugins 715
D.4 Create a frame for a PiP 716
D.5 Using Screen Capture on slower CPUs 718
D.6 Workflow for render yuvj (color range: jpeg) to yuv (color range:mpeg) 719

E FAQ, Known Problems and Workarounds **721**

Glossary **725**

List of Tables **738**

List of Figures 739

Index 747

CHAPTER 1

Installation

1.1 CINELERRA-GG AppImage

The main way to install CINELERRA-GG is to use the AppImage. This is updated regularly and works for every distro, since it already contains the necessary dependencies. A big advantage of using the AppImage format is that it is only 1/3 the size of the normal install, and since each release is named differently, you can keep a number of versions in a directory, and when testing from a terminal you just have to type CinGG, then hit tab, and complete it to the desired date release. A small disadvantage of using the AppImage format is that some of the options to make minor text type changes are not available and any graphics board speedups most likely will not work. In many cases, rendering has been found to take a small percentage longer than if you do your own build or use a package RPM as documented at Pre-built Packages and Distros with CINELERRA-GG Included. It is assumed that general processing will be slightly slower also.

For 64-bit systems you can choose between an image with up-to-date libraries or one that supports older libraries (named *older-distro*), which you should use only if the first image gives you problems with unsupported libs.

There are also 32-bit distros available that have *i386* as part of the filename that currently works on older distros but may not work on the newest distros (most of the popular Linux distributions such as Arch, Ubuntu, and Fedora have dropped support for this older architecture). In any case, if you are using a 32-bit Linux distro, you might want to compile your sources from git or use a precompiled binary[1].

Installing the appimage is simple, download the file from:

`https://cinelerra-gg.org/download/images/`

Some example file names are as follows - where 8 digits represent yyyymmdd and

[1] Remember that a 32-bit distro does not address more than 4GB of memory, so you may have stability and performance problems with large, high-resolution mediafiles.

i386 are 32-bit:

```
CinGG-20241231-x86_64.AppImage
  (currently based on Fedora 32, linux kernel 5.8.15, libc version 2.31)
CinGG-20241231-x86_64-older-distros.AppImage
  (currently based on Ubuntu 16.04, libc version 2.23)
CinGG-20240731-x86_64-older-distros-multibit.AppImage
  (currently based on Fedora 29 -runs on RHEL8 -linux kernel 4.19.9, libc version
    ↪ 2.28)
CinGG-20241231-alternative_shortcuts.AppImage
  (currently based on Ubuntu 16.04, libc version 2.23)
CinGG-20241231-i386.AppImage
  (currently based on Debian 9, linux kernel 4.9, use "newer" for Debian 11.0)
CinGG-20241231-i386-newer-distros.AppImage
  (currently based on Debian 11, linux kernel 5.10)
CinGG-20241120-x86_64-IntelHW.AppImage
  (currently hardware bells and whistles for Intel to include QSV/AV1/oneVPL)
```

Make the file executable with the proper execute permissions either from the GUI of the Desktop Environment used (link to the file) or from a terminal window. Make sure you are already in the directory containing the appimage:

```
\$ chmod u+x CinGG-yyyymmdd.AppImage
```

Finally start the program from a window in the directory where the image is stored:

```
\$ ./CinGG-yyyymmdd.AppImpage
```

or create a convenient desktop icon with a link to the run action, or do a *Desktop Integration* manually or with external programs. There is a description of a GUI methodology for doing so in this file on the webiste:

`https://cinelerra-gg.org/download/images/README_appimage.txt`

Most distros already have the libraries to run the appimage, but if not you may need an additional installation. For example Arch Linux needs the `libappimage` package.

```
sudo pacman -S libappimage
```

And Leap 15.3 (OpenSUSE) requires installation of the *appimage* package.

```
sudo zypper se -is appimage
```

In addition, if you are using the OpenGL video driver, you will need to install the appropriate OpenGL drivers for your Operating System graphics board because libGLU.so and other OpenGL libraries are not included in the AppImage.

Using AppImage means you do not have the installation folder and work on the files. To unpack the AppImage and get its structure in folders and files see Managing AppImage To create, edit and manage appimages see Build the CinGG.AppImage from scratch.

1.1.1 AppImage with Standard Shortcuts

In video editing it is important to learn how to use shortcuts to speed up your work. CINELERRA-GG uses shortcuts different from those considered standard in both the Linux world and video editing. For example, `"s"` is used instead of `Ctrl+S`, `"q"` instead of `Ctrl+Q`, and even the classic editing keys `J, K, L` are different. In addition, in CINELERRA-GG the keys are fixed and not customizable. A new user may have a hard time getting used to a new combination of shortcuts. To make it a little easier, an appimage containing a patch that makes use of some of the more frequently used classic key combinations is available. It can be downloaded here (note that the file contains the month and last day of the month, but you will want to go up a directory and download the latest date instead to include the current changes). A table showing the changes from CINELERRA-GG mode to standard mode can be found here: Alternative Shortcuts.

1.2 Download Already Built CINELERRA-GG

All of these images are dated 10/31/2020 and are no longer being maintained. They will still work on the version of the O/S in use at that time but will have none of the latest features. You should use the simpler AppImage instead as described previously.

If you prefer to not have to take the time to build CINELERRA-GG Infinity yourself, there are pre-built dynamic or static binaries for various versions of Ubuntu, Mint, Suse, Fedora, Debian, Centos, Arch, and Slackware linux as well as Gentoo and FreeBSD. If you do want to build it yourself so that you get the added benefit of the latest checked in changes, please reference 1.3. A Windows 10 version installation is described in 1.5. There are also 32-bit i686 Ubuntu, Debian, and

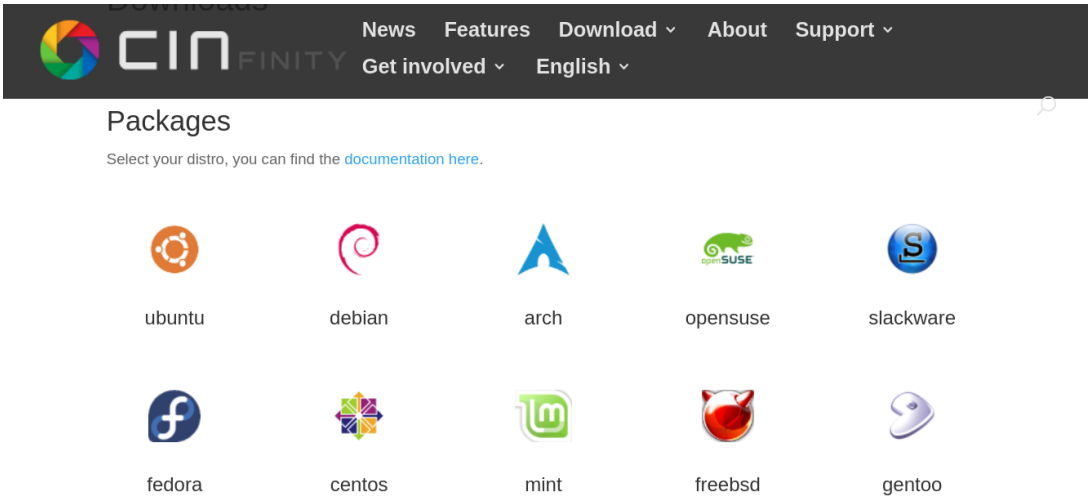


Figure 1.1: *Screencast of the website Download page for installing* CINELERRA-GG *for various O/S.*

Slackware versions available[2]. **These binaries are no longer being updated; they are stable and working but without future functionality**. They are in subdirectories of:

https://cinelerra-gg.org/download/tars

https://cinelerra-gg.org/download/pkgs

The **tars** directory contains single-user static builds for different distros. This is the recommended usage of CINELERRA-GG because all of the files will exist in a single directory. Generally all of the necessary libraries are built into the static build, but in some cases you may have to install another library that is being called for. To install the single user builds, download the designated tarball from the `./tars` subdirectory and unpack as indicated below:

```
cd /path
mkdir cin
cd cin
tar -xJf /src/path/cinelerra-5.1-*.txz # for the *, substitute your distro tarball name
```

[2]Remember that a 32-bit distro does not address more than 4GB of memory, so you may have stability and performance problems with large, high-resolution mediafiles.

Do not download the LEAP 10-bit version unless you specifically want to use h265 rendering to 10-bit instead of the more standard 8-bit. For more information see **??**.

The **pkgs** directory contains the standard packaged application for various distros. This will install a dynamic system version for users who prefer to have the binaries in the system area and for multi-user systems. In addition, performing the package install checks the md5sum in the file `md5sum.txt` to ensure the channel correctly transmits the package. There is a README.pkgs file in the `download` directory with instructions so you can *cut and paste* and avoid typos; it is also shown next.

Listing 1.1: *README.pkgs*

```
# Depending on the distro, use the instructions below and
# select the appropriate setup operations to install, update
# or remove cinelerr-gg infinity. (11/24/2019) To upgrade,
# refresh repo, then replace "install" with "update", or
# whatever.

# Email problems to cin@lists.cinelerra-gg.org. If
# repository problems, usually you can manually do an
# install by using:

# UBUNTU, MINT, DEBIAN
wget https://cinelerra-gg.org/download/pkgs/{substitute_name}/cin_5.1.<sub_name>.deb
# and install it manually:
dpkg -i cin_5.1.{substitute_filename}.deb

# ARCH
wget https://cinelerra-gg.org/download/pkgs/{substitute_name}/cin_5.1.<sub_name>.pkg.tar.
    ↪ xz
# and install it manually:
pacman -U cin_5.1.{substitute_filename}.pkg.tar.xz

# FEDORA
wget https://cinelerra-gg.org/download/pkgs/{substitute_name}/cin_5.1.<sub_name>.rpm
# and install it manually:
dnf install cin_5.1.{substitute_filename}.rpm

# LEAP, SUSE
wget https://cinelerra-gg.org/download/pkgs/{substitute_name}/cin_5.1.<sub_name>.rpm
# and install it manually:
zypper install cin_5.1.{substitute_filename}.rpm

# CENTOS
wget https://cinelerra-gg.org/download/pkgs/{substitute_name}/cin_5.1.<sub_name>.rpm
# and install it manually:
yum localinstall cin_5.1.{substitute_filename}.rpm

# GENTOO
# There are static and dynamic tarballs for
```

```
# Base Release 2.6
https://cinelerra-gg.org/download/tars/cinelerra-5.1-gentoo-20200202.x86_64-static.txz
https://cinelerra-gg.org/download/tars/cinelerra-5.1-gentoo-20200202.x86_64.txz
# download one of the above and then refer to README.txt

# FREEBSD
# There is a tarball based on FreeBSD version 12.1 at
https://cinelerra-gg.org/download/testing/bsdcin.tar.xz
# download the above and then refer to README.txt

# FEDORA
# Replace the XX in fedoraXX in the next
# line with your current O/S version number
dnf install cinelerra --nogpgcheck --repofrompath cingg,https://cinelerra-gg.org/download/pkgs
    ↪ /fedoraXX/
# dnf erase cinelerra

# CENTOS
# Python 2 has been updated for other distros to Python 3 so
# you might have to create a soft link to get the correct
# version. For help, send email to
# cin@lists.cinelerra-gg.org first create the file
# /etc/yum.repos.d/cin_gg, with the following contents:
[cin_gg]
name=cingg
baseurl=https://cinelerra-gg.org/download/pkgs/centos7
gpgcheck=0
# end of cin_gg
yum install cinelerra
# yum erase cinelerra

# UBUNTU, replace ub14 with your distro id: ub16,ub18 Some
# ubuntu apt downloads register status as working 0%
# constantly while running the package download, like ubuntu
# 14. It may take a few minutes for this step so be patient.
apt install software-properties-common apt-transport-https
apt-add-repository https://cinelerra-gg.org/download/pkgs/ub14

# UBUNTU 16/18 note -This has been known to work, but
# things change quickly:
# VIP -for the first install, the above line adds cinelerra
# to /etc/apt/sources.list but... Version 16/18 of Ubuntu
# are more strict for licensing so you will have to edit the
# file /etc/apt/sources.list to add [trusted=yes] after deb
# and before https...cin...

# For example the line should be:
# deb [trusted=yes] https://cinelerra-gg.org/download/pkgs/ub16 xenial main
# Or for ub18:
# deb [trusted=yes] https://cinelerra-gg.org/download/pkgs/ub18 bionic main
# Also, on the install you will get an error message that you
# can either ignore as cinelerra will run anyway, or else (the
```

```
# first time only) on the commnand line keyin:
# echo > /etc/sysctl.d/50-cin.conf "kernel.shmmax=0x7fffffff"
apt update
apt install cin
# to update a previous install (ignore any i386 errors as only 64
# bit version available):
apt update
apt upgrade cin
# apt remove cin

# MINT should use the same procedure as Ubuntu, but
# apt-add-repository does not seem to work, so use the GUI
# UpdateManager as follows:
# Administration -> Software Sources -> Additional
# Repositories -> Add a new repository
# (Note instead of Administration, some versions of Mint GUI
# UpdateManager might be System)
# For Mint18, add:
# deb [trusted=yes] https://cinelerra-gg.org/download/pkgs/mint18 xenial main
# For Mint19, add:
# deb [trusted=yes] https://cinelerra-gg.org/download/pkgs/mint19 bionic main
# IMPORTANT NOTE: if you get "malformed input" error, you will
# have to create a file by typing the command: sudo touch
# /etc/apt/sources.list.d/additional-repositories.list then
# wait 10 minutes or so and try using the Gui Update Manager
# again.
apt update
apt install cin
# to update a previous install
apt update
apt upgrade cin
# apt remove cin

# DEBIAN uses the same basic procedure as Ubuntu.
# The apt-add-repository varies per system so you will have to
# use your best judgement
apt install software-properties-common apt-transport-https
apt-add-repository https://cinelerra-gg.org/download/pkgs/debian8
# OR
# apt-add-repository https://cinelerra-gg.org/download/pkgs/debian9
# OR
# apt-add-repository https://cinelerra-gg.org/download/pkgs/debian10
# VIP -for the first install, the above line adds cinelerra
# to /etc/apt/sources.list but... Debian jessie/buster are
# more strict for licensing so you will have to edit the file
# /etc/apt/sources.list to add [trusted=yes] after deb and
# before https...cin...
# For example for debian9:
# deb [trusted=yes] https://cinelerra-gg.org/download/pkgs/debian9 stretch main
# For example for debian9:
# deb [trusted=yes] https://cinelerra-gg.org/download/pkgs/debian10 buster main
apt update
```

```
apt install cin
#to update a previous install
apt update
apt upgrade cin
# apt remove cin

# SUSE/LEAP/TUMBLEWEED
# (Note: you may have to zypper libavc and libiec versions if
# not already installed) cinelerra packages are unsigned so
# you will have to ignore: Package is not signed!

# openSUSE LEAP 15
zypper ar -f https://cinelerra-gg.org/download/pkgs/leap15/ cingg
zypper install -r cingg cinelerra # or cinelerra10bit for 10 bit
# openSUSE LEAP 42
zypper ar -f https://cinelerra-gg.org/download/pkgs/leap42/ cingg
# as of 42.3 SUSE there is a new requirement, so you will need
# to add:
zypper mr -G cingg
zypper install -r cingg cinelerra # or cinelerra10bit for 10 bit
# openSUSE TUMBLEWEED
zypper ar -f https://cinelerra-gg.org/download/pkgs/tweed/ cingg
# as of 42.3 SUSE there is a new requirement,
# so you will need to add:
zypper mr -G cingg
zypper install -r cingg cinelerra
#zypper remove cinelerra
# or cinelerra10bit for 10 bit.
# to update a previous install
# (assuming you enabled autorefresh as above)
zypper refresh cingg
zypper up cinelerra
# or cinelerra10bit for 10 bit.

# SLACKWARE
# substitute slk32 for slk64 and i486-1 for x86_64-1
wget -P /tmp https://cinelerra-gg.org/download/pkgs/slk64/cin-{date}-slk64-x86_64.txz
installpkg /tmp/cin... # name you used in the above line
#to update a previous install
upgradepkg /tmp/cin... # name you used in the above line
#removepkg cin

# ARCH linux
# (A loosely defined list of packages that you should install
# first is listed in this file:
# https://www.cinelerra-gg.org/download/pkgs/README.arch) first
# edit the file /etc/pacman.conf, to include the following:
[cingg]
SigLevel = Optional TrustAll
Server = https://cinelerra-gg.org/download/pkgs/arch
# end of cingg
```

```
# next run from a window the following 2 commands;
pacman -Syu
pacman -S cin

# NOTE: the first line above updates your Arch system to the
# current rolling release and the second line updates
# Cinelerra-GG based on the rolling release that was in effect
# on the last day of the month.
# Please complete the 2 steps above in order, one right after
# the other to avoid risk of a partial upgrade.

# Due to the unpredictability of when Arch libraries are
# updated, performing an install of Cinelerra at any time
# other than shortly after the last day of the month when the
# new build package is created, could lead to library
# incompatibilities. In that case, please consider using the
# Arch static tar file for installation instead. to remove a
# previous install:
# pacman -R cin
```

1.3 How to Build CINELERRA-GG from Developer's Git Repository

These are generic build instructions for building CINELERRA-GG Infinity. Known to work on Ubuntu, Mint, OpenSuse, Fedora, Debian, Centos, Arch, Slackware, and Gentoo. Compiling from git is perhaps the best way to get CINELERRA-GG on 32-bit systems[3]. It has not been tested on every single possible distro yet so you might expect to have to make some minor changes. Also works on a somewhat limited basis on FreeBSD and Windows 10 with the bsd.patch for FreeBSD and the cygwin.patch for Windows 10. As of 10/31/2020, FreeBSD and Windows 10 builds and patches are no longer being maintained so that they will work using the GIT version in use at that time but you will have to create new patches for arising problems on later GITs.

NOTE: as of May 31, 2021 when Context Help was added, to include this Context Help you will need to download the corresponding tgz file containing the HTML manual sections referenced for the Help pages. The file to download is: `https://cinelerra-gg.org/download/images/HTML_Manual-20220131.tgz` substituting for "20220131" the "yyyymmdd" representing latest release date. Then unpack to your Cinelerra/bin/doc directory so it is included in your built system. NOTE End

[3]Remember that a 32-bit distro does not address more than 4GB of memory, so you may have stability and performance problems with large, high-resolution mediafiles.

Alternatively, there are some pre-built dynamic or static binaries which are updated on a fairly regular basis (as long as code changes have been made) available at the link below.

https://cinelerra-gg.org/download/

There are 2 kinds of builds, the default system-build and a single-user build. A system build has results which are installed to the system. The majority of the files are installed in the standard system paths, but some customization is possible. The single user build allows for running completely out of a local user directory so it doesn't affect the system.

We recommend the single-user version when possible. It makes it very easy to install a new version without having to delete the older version in case you want it for backup – once you are happy with the new version, all you have to do is delete the entire old directory path. Another reason for using single-user is that if you install a new Operating System version and if you have CINELERRA-GG on separate disk space that is preserved, you won't have to reinstall CINELERRA-GG. It is also convenient for the purpose of having the ability to interrupt or to see any possible error messages, if you start the application from a terminal window command line where you will have more control to catch problems. All that said, the system builds can be useful in a university lab setting where there are possibly multiple users, or multiple versions.

There are two notable differences between standard views of CINELERRA-GG and this implementation for the system builds. Both of these can be configured during installation. The differences make it possible to have several different versions installed without having them interfere with each other.

1. application name can be set during a build but defaults to: `cin`
2. the home configuration directory can also be set and traditionally defaults to: `$HOME/.bcast5`

1.3.1 The system build

To do a system build , you should read the file `README` that is at the top level after you get the source.

- You need about 6.0 GB of disk storage to operate a build and you need to have *git* installed.

- You do not need to be **root** (or *sudo* ...) to install, except to run `bld_prepare.sh` which calls in the distro's package manager. However if there are problems with permissions you can try to compile as root.
- The *git:* step has to download many files (approx 130 MB) so allow time. When decompressed this will expand to about 530 MB.
- Run the following commands (this takes awhile):

```
# This is where you need the 6.0GB of disk space:
cd /<build_path>/
git clone --depth 1 git://git.cinelerra-gg.org/goodguy/cinelerra.git cinelerra5
# Change to the cloned directory:
cd cinelerra5/cinelerra-5.1
```

NOTE: if your system has never had CINELERRA-GG Infinity installed, you will have to make sure you have all of the compilers and libraries necessary. So on the very first build you should run:

```
./blds/bld_prepare.sh <os> # where <os> represents the
                           # Operating System of
                           # centos, fedora, suse, ubuntu, mint, debian, arch, debian-
                               ↪ older, ubuntu-older.
./autogen.sh
./configure --prefix=/usr # optional parameters can be added here
make 2>&1 | tee log # make and log the build
```

Where <os> represents the Operating System supported by CINELERRA-GG, such as centos, fedora, suse, ubuntu, mint, or debian. `bld_prepare.sh` works for Arch and Gentoo with some additional information. For Arch linux, a README file containing many more dependencies is maintained. For Gentoo, a README file lists other dependencies that have to be installed manually.

https://cinelerra-gg.org/download/README.arch

https://cinelerra-gg.org/download/README.gentoo

`bld_prepare.sh` option of debian-older and ubuntu-older is currently for older operating system versions since before about 06/2022.

- Check for obvious build errors:

```
grep "\*\*\*.*error" -ai log
```

If this reports errors and you need assistance or you think improvements can be made to the builds, email the log which is listed below to: cin@lists.cinelerra-gg.org

```
/<build_path>/cinelerra5/cinelerra-5.1/log
```

- If there are no build errors, finally just run:

```
make install
```

The "with-single-user" parameter makes it so. Check for errors before proceeding.

- If it all worked, you are all setup. Just click on the CINELERRA-GG desktop icon.

1.3.2 The single-user build

To do a single-user build, read the file `README` that is at the top level after you get the source.

1. You need at least 6 GB of disk storage to operate a build + you need to have "`git`" installed.
2. You can install it without being **root** or without using *sudo*. In case of problems you can use *sudo* to avoid permission issues.
3. The *git* step has to download many files (approx 130 MB) so allow time.
4. Run the following commands (this takes awhile):

```
# This is where you need the 6GB of disk space
cd /<build_path>/
git clone --depth 1 git://git.cinelerra-gg.org/goodguy/cinelerra.git cinelerra5
# Toplevel directory:
cd cinelerra5/cinelerra-5.1
```

NOTE: if your system has never had CINELERRA-GG Infinity installed, you will have to make sure all the compilers and libraries necessary are installed. Thus, for the execution part of `bld_prepare.sh` you must use sudo, but the other steps can be done as a normal user.

```
./blds/bld_prepare.sh <os>
./autogen.sh
./configure --with-single-user
make 2>&1 | tee log
make install
```

Where <os> represents the Operating System supported by CINELERRA-GG, such as centos, fedora, suse, ubuntu, mint, debian and arch. The "with-single-user" parameter makes it so. Check for errors before proceeding.

Then just start the application by keying in: `./cin` in the bin subdirectory OR add a desktop icon by using the appropriate directory to copy the files to and edit to correct the directory path. Below are generic directions of how to do this.

Then just start the application by keying in: `./cin` in the bin subdirectory OR add a desktop icon by using the appropriate directory to copy the files to, run as root, and edit to correct the directory path. Below are generic directions of how to do this.

```
cd /cinelerra_directory_path
cp -a image/cin.{svg,xpm} /usr/share/pixmaps/
cp -a image/cin.desktop /usr/share/applications/cin.desktop
```

After you have followed the above, in the cin.desktop file, change the `Exec=cin` line to be `Exec=<your_directory_path>/bin/cin`.

A working example of how to build in Arch as a normal user:

```
\$ git clone --depth 1 git://git.cinelerra-gg.org/goodguy/cinelerra.git cinelerra5
\$ cd /home/USER/cinelerra5/cinelerra-5.1
\$ ./autogen.sh
\$ ./configure --with-single-user --with-booby
\$ make 2>&1 | tee /tmp/cin5.log && make install
\$ mv Makefile Makefile.cfg
\$ cp Makefile.devel Makefile
```

1.3.3 8-bit build for x265 encoding

The standard versions of AppImage, the pre-compiled binary and any build you do yourself, is the *multibit* version by default, which allows rendering of the x265

codec at all supported bit-depths. If instead you prefer to compile the 8-bit only version, there is the option to do so. The 8-bit version does not have 10/12-bit encoding of x265; the 8-bit version works on 8-bit bytes which is the standard word size for computers, while the multibit version has to use 2 words to contain all 10/12 bits so you might expect rendering to be slower. Various tests show that there is little or no difference in performance between the two versions, especially on modern systems, so the standard multibit version is the recommended one. If you have a need to compile the 8-bit version first perform the following steps:

1. Move to the thirdparty directory: cd /path_to/cinelerra-5.1/thirdparty
2. Edit the file "Makefile" by changing these 3 lines to switch the # comment character like this:
   #x265.cfg_vars?=chmod +x ./configure; chmod +x ./multilib.sh;
   x265.cfg_vars?=$(call cmake_config,source)
   x265.cfg_params?= -DENABLE_SHARED=no -DENABLE_CLI=no
3. Then delete or rename the 3 patch files in thirdparty/src with names of x265-xxxxx.patchx.

Using the 8-bit version will result in an error if you use the Render formats *h265-10bit* and *h265-12bit*. Error message on the startup window is something like this:

```
[libx265 @ 0x7fa00e643880] Specified pixel format yuv422p1xle is not supported by
    ↪ the libx265 encoder.
```

1.3.4 Notable Options and Caveats

These procedures and the CINELERRA-GG Infinity software have all been run as **root** on various home laptops and desktops. This provides the best chance to ensure all works correctly and also allows for handling errors, other problems and potential crashes with the most success. Included in this section are some of the build variations easily available for normal builds.

You can, during compilation, use a patch that changes the main non-standard shortcuts of CINELERRA-GG to standard ones (`Ctrl`+S and `J`, `K`, `L`, etc.). A table showing the changes from CINELERRA-GG mode to standard mode can be found here: Alternative Shortcuts. The instructions for the build with the patch are as follows. After downloading the sources from the git repository in the usual way, you apply the patch:

```
cd cinelerra-5.1
patch -p1 -i alt_shortcuts.patch
./bld.sh
```

To see the full list of features use:

```
./configure --help
```

The default build is a system build which uses:

```
./configure --without-single-user
```

In the single-user build , the target directory is always `cin`. Because this is also the developer build, constant names are used throughout. However, you can rename files after the install is complete.

If your operating system has issues with the default install to `/usr/local`, you might have to change the location to `/usr` for a system build. Then you will have to use:

```
./configure --prefix=/usr
```

If you wish to change the default directory for a system build you will have to add the destination directory path on the `make install` line. For example:

```
make install DESTDIR=<your selected target directory path>
```

The application name can be set during installation, but defaults to `cin` so that the GG/Infinity build can coexist with other CINELERRA-GG builds if necessary. To override the default `cin` name, use:

```
./configure --with-exec-name=cinelerra
```

The home configuration directory can also be set, but default location is traditionally `$HOME/.bcast5`. For example:

```
./configure -with-config-dir=/myusername/.bcast5
```

NOTE: when you specify parameters to the configure program, it will create a `make` file as a consequence. Since in a `make` file, the $ is a special character, it must be escaped so in order to represent a $ as part of an input parameter, it has to be stuttered. That is, you will need $$ (2 dollar signs) to represent a single dollar sign.

It may be necessary on some distros which have missing or incomplete up-to-date libraries, to build CINELERRA-GG without Ladspa. To do so, use:

```
./configure --prefix=/usr --without-ladspa-build
```

Note that the with-ladspa-dir is the ladspa search path, and exists even if the ladspa build is not selected. This gives you the ability to specify an alternate ladspa system path by utilizing the `LADSPA_PATH` environment variable (that is, the default ladspa build is deselected).

Note for 32-bit 14.2 Slackware, Debian, Gentoo, Arch, FreeBSD, before running the configure, you will need to set up the following:

```
export ac_cv_header_xmmintrin_h=no
export FFMPEG_EXTRA_CFG=" --disable-vdpau"
```

Note for building 32-bit packages on hybrid 32/64 x86 systems, you may need to add the following:

```
setarch i686 (befire configure and package build)
```

NOTE: as of May 31, 2021 when Context Help was added, to include this Context Help you will need to download the corresponding tgz file containing the HTML manual sections referenced for the Help pages. The file to download is: `https://cinelerra-gg.org/download/images/HTML_Manual-20220131.tgz` substituting for "20220131" the "yyyymmdd" representing latest release date. Then unpack to your Cinelerra/bin/doc directory so it is included in your built system. The reason for not including the HTML manual in the source code so that it would already be there, is because it is very large and has its own GIT base.

1.3.5 Notes about Building from Git in your Customized Environment

Getting a build to work in a custom environment is not easy. If you have already installed libraries which are normally in the thirdparty build, getting them to be recognized means you have to install the *devel* version so the header files which match the library interfaces exist. If you want to build using only the thirdparty libraries installed in your system, just include "–without-thirdparty" to your configure script. For example:

```
./confgure --with-single-user --disable-static-build --without-thirdparty
```

Below is the list of thirdparty builds, but this list may have changed over time.

Table 1.1: *List of thirdparty builds*

a52dec	yes
djbfft	yes
ffmpeg	yes
fftw	auto
flac	auto
giflib	yes
ilmbase	auto
lame	auto
libavc1394	auto
libraw1394	auto
libiec61883	auto
libdv	auto
libjpeg	auto
opus	auto
openjpeg	auto
libogg	auto
libsndfile	auto
libtheora	auto
libuuid	yes
libvorbis	auto
mjpegtools	yes
openexr	auto
tiff	auto
twolame	auto
x264	auto
x265	auto
libvpx	auto
lv2	auto

sratom	auto
serd	auto
sord	auto
lilv	auto
suil	auto
libaom	auto
dav1d	auto
libwebp	auto
ffnvcodec	auto

The *yes* means force build and *auto* means probe and use the system version if the build operation is not static. To get your customized build to work, you need to change the probe options for the conflicting libraries from *yes* to *auto*, or even rework the `configure.ac` script. There may be several libraries which need special treatment.

An example of a problem you might encounter with your customized installation is with `a52dec` which has probes line `(CHECK_LIB/CHECK_HEADERS)` in `configure.ac`, but `djbfft` does not. In this case, `djbfft` is only built because `a52dec` is built, so if your system has `a52dec`, set `a52dec` to auto and see if that problem is solved by retrying the build with:

```
./confgure --with-single-user -enable-a52dec=auto
```

If you have a problem with libdpx (DPX image format encoding/decoding library), resolve that by changing the configure line as shown to build without it:

```
./confgure --with-single-user --without-libdpx
```

If you have some of your thirdparty libraries installed in non-standard locations or if you want to use a specific version, consider setting CFLAGS=-I/<path> to pass that location to the compiler so it knows where to find it.

With persistence, you can get results, but it may take several tries to stabilize the build. If you need help, email the `log` and `config.log`, which is usually sufficient to determine why a build failed.

If you have already installed the `libfdk_aac` development package on your computer because you prefer this version over the default aac, you will have to do the following to get this alternative operational. The libfdk_aac library is not a part of CINELERRA-GG by default because it is not license free.

```
export FFMPEG_EXTRA_CFG=" --enable-libfdk-aac --enable-nonfree"
export EXTRA_LIBS=" -lfdk-aac"
for f in ‘grep -lw aac cinelerra-5.1/ffmpeg/audio/*‘; do
  sed -e ’s/\<aac\>/libfdk_aac/’ -i $f
done
```

1.3.6 Cloning the Repository for Faster Updates

If you want to avoid downloading the software every time an update is available you need to create a local "repository" or repo. The repo is a directory where you first do a `git clone`. For the initial git clone, set up a local area for the repository storage, referred to as `<repo_path>`. The `git clone` creates a repo named `cin5` in the `/<repo_path>/` directory. This accesses about 530 MB of repo data, so the device has to have at least that available. The repo path is always a perfect clone of the main repo.

Setting up the initial clone You may want to add "`--depth 1`" before `cin5` because this will clone faster and is smaller, but has no history.

```
cd /<repo\_path>/
git clone git://git.cinelerra-gg.org/goodguy/cinelerra cin5

Cloning into "cin5"...
remote: Counting objects: 20032, done.
remote: Compressing objects: 100% (11647/11647), done.
remote: Total 20032 (delta 11333), reused 16632 (delta 8189)
Receiving objects: 100% (20032/20032), 395.29 MiB | 3.26 MiB/s, done.
Resolving deltas: 100% (11333/11333), done.
Checking connectivity... done.
```

Update an existing repo The below shows how you can get updates.

```
cd /<repo home>/cin5
git pull
```

Useful git commands Some other commands that are useful.

```
git clone git://git.cinelerra-gg.org/goodguy/cinelerra.git cin5
git pull # pull remote changes to the local version
git status # shows changed files
git clean -i # interactive clean, use answer 1 to "clean"
```

1.3.7 How to Build from a Previous GIT Version

If you have a problem with the current GIT version, you can revert to a previous working version easily. The commands to use will be similar to these next lines which are then explained in more detail. You need "history" to do this, so should not have used the "depth 1" parameter on your GIT.

```
cd /<path>/cin5 # substitute your repo path name for cin5
git log  # shows a list of versions depending on history depth specification
git checkout <version> # choose a version number as listed
```

The `git log` command produces a log file with hash values for commit keys to the level specifed if the the depth paramter was specified. The hash ids are the commit names to use when you use git checkout. Next is displayed sample output:

```
delete stray line in last checkin

commit 4a90ef3ae46465c0634f81916b79e279e4bd9961
Author: Good Guy <good1.2guy@gmail.com>
Date: Thu Feb 22 14:56:45 2018 -0700

nested clips, big rework and cleanup, sams new icons,
leaks and tweaks

commit f87479bd556ea7db4afdd02297fc00977412b873
Author: Good Guy <good1.2guy@gmail.com>
Date: Sat Feb 17 18:09:22 2018 -0700
```

For the `git checkout <version>`, you would then keyin the line below for the following results:

```
git checkout f87479bd556ea7db4afdd02297fc00977412b873
```

```
Note: checking out 'f87479bd556ea7db4afdd02297fc00977412b873'.

You are in 'detached HEAD' state. You can look around, make
experimental changes and commit them, and you can discard any
commits you make in this state without impacting any branches by
performing another checkout.

If you want to create a new branch to retain commits you create,
you may do so (now or later) by using -b with the checkout command
again. Example:

git checkout -b <new-branch-name>

HEAD is now at f87479bd... more file size icon updates,
and more to followend
```

Later to get the repo back to current, use:

```
git checkout master
```

1.3.8 Debuggable Single User Build

To build from source with full debugging symbols, first build a full static (non_debug) build as follows but instead of using `/tmp` substitute your permanent disk path if you want to keep it.

```
cd /<repo_path>/
git clone --depth 1 git://git.cinelerra-gg.org/goodguy/cinelerra.git cinelerra5
cp -a /<repo_path>/cinelerra-5.1 /tmp/
cd /tmp/cinelerra-5.1
./bld.sh
```

Then, to run as a developer in the debugger:

```
CFLAGS="-O2 -ggdb" make -j8 rebuild_all
cd cinelerra
gdb ./ci
```

When you get the gdb prompt, type the letter "r", for run, and the windows will come up. If there is a crash, the windows will freeze and typing "bt" for backtrace in the startup window after the gdb prompt, provides useful information.

1.3.9 Unbundled Builds

There are some generic build scripts included in the CINELERRA-GG GIT repository for users who want to do unbundled builds with ffmpeg already available on their system. This has been tested on Arch, Ubuntu 18, FreeBSD, Windows10 and Leap 15 (rpm) at the time this was documented. The names of the build scripts are: `arch.bld`, `bsd.bld`, `deb.bld`, `rpm.bld`, and `cygwin.bld`. These scripts are in the `blds` subdirectory. The `bsd.bld` should be used with the `bsd.patch` file in that same directory. The `cygwin.bld` should be used with the `cygwin.patch` file in that same directory.

The reason that Cin Infinity traditionally uses its own thirdparty builds (bundled builds) is because there are a lot of different distros with varying levels of ffmpeg and other needed thirdparty libraries. However, some users prefer using their current system baseline without another/different copy of ffmpeg. With different levels of the user's libraries, uncertainty, potential instability, and unknown issues may come up while running CINELERRA-GG and this will make it, for all practical purposes, impossible to diagnose and debug problems or crashes. There may be no help in these cases. You are encouraged to report any errors which potentially originate from Cin Infinity, but if the data indicates alternate library sources, please report the problems to the appropriate maintainers.

With the unbundled builds, some features may not be available and no attempt to comment them out has been made. So if you use a pulldown, or pick a render option, or choose something that is not available, it just will not work. For example, unless special options were set up by you, the LV2 audio plugins will not be available. Nor will the codec libzmpeg, the file codec ac3, or DVD creation. The old school file classes will all work, but some of the formats that come with ffmpeg may not because of the way that ffmpeg was installed on your operating system. That is because the CINELERRA-GG included ffmpeg is a known static build and is usually the latest stable/released version. For example, in the current case of Leap 15, libx264 and libx265 are not built in and this can be debilitating; you can always run `ffmpeg -formats` and `ffmpeg -codecs` to see what is available on your system.

1.4 Building the HTML Manual for Context Help

In addition to compiling your own CINELERRA-GG, you should also build an html version of the manual that is needed for Context Help in the program. The main

version of the manual is in latex to produce a pdf version of the manual and this is required to be built first as the basis for the html version. This means that you need a full latex environment, git, and the latex2html program in order to eventually create the html version. Texlive is about 1 GB; Latex2html itself has many requirements and missing any will result in failure: some requirments include Netpbm, GhostScript, dvips, etc. Latex2html must be at least version *2021.2* in order to create the html manual version from the latex.

The steps are as follows:

1. Download the manual in LaTeX:

```
git clone git://git.cinelerra-gg.org/goodguy/cin-manual-latex.git master
```

2. Included in the download is the `translate_manual` script. After modifying this file to have execute permission, run this script from a terminal window in the *master* directory where it was downloaded (be aware that this script includes several *rm* commands):

```
./translate_manual
```

   The steps that this script performs are as follows:

   - PDF production. The PDF document will be produced from the latex source in the *master* directory. Since the glossary and index are also present, it has to run the pdf build several times. The following commands in the `translate_manual` script produce the PDF document from latex source which includes invoking makeindex for the Index and Glossary.

```
pdflatex CinelerraGG_Manual.tex
makeindex CinelerraGG_Manual.idx
pdflatex CinelerraGG_Manual.tex
makeindex CinelerraGG_Manual.nlo -s nomencl.ist -o
    ↪ CinelerraGG_Manual.nls
pdflatex CinelerraGG_Manual.tex
```

     After these commands are executed you will have the manual only in PDF format. So if you only want a PDF version, you only need to run these previous 5 lines but Context Help from the program will not be available with the PDF version.

- Next, to produce HTML output the script then moves (renames) `latex 2html-init` to `.latex2html-init` (starting with dot).
- Then the script uses latex2html: latex2html is run with a unique set of parameters and some cleanup is performed. It creates the directory CinelerraGG_Manual containing all the files of the manual in html: tables, references, index, glossary, and various images.

3. After installation of the CINELERRA-GG program, place the complete unchanged directory `CinelerraGG_Manual`, as it was produced by latex2html from the manual package, into the *doc* directory of the installed Cinelerra package. This will be the directory *bin/doc/CinelerraGG_Manual* if CINELERRA-GG was built `-with-single-user`. The script ContextManual.pl will automatically be in bin/doc after the successful build of the program. It is this perl script that allows the program to access CinelerraGG_Manual to offer Context Help.

4. Optionally you can make some adjustments to the latex2html command line in the `translate_manual` script. Some variants are shown in the comments inside the script but changes may impact the usability of Alt/h hotkey from the program.

1.5 Windows 10 with Cygwin for CINELERRA-GG Limited

As of 10/31/2020, this is no longer being maintained. It should still work using an older GIT version with Windows 10 but it is possible with some effort to modify the patch file to work with the latest updated GIT.

To run CINELERRA-GG on a Windows 10 computer, you will need to have Cygwin installed on your system, along with the CINELERRA-GG static tar and a patched library: libxcb. This setup has been tested with Windows 10, version 1909, on an HP EliteBook 820 at 2.3 GHz.

This limited version provides *core* functionality at this time with the standard Windows FFmpeg executable, meaning that specific modifications in FFmpeg needed for CINELERRA-GG are not available. Limited capabilities include only a few render output formats available - for example *mov*, *qt* as *mjpeg*, and *mpeg* for videos and *avi* and *qt* as *s16le* for audio, but not *mkv* or *mp4*. This is due to the fact that several codec and utility libraries are not currently compiled to work with Windows.

Installing Cygwin

Cygwin is an environment that runs natively on Windows which allows Unix programs to be compiled and run on Windows. With cygwin installed on your Windows 10 computer, you will be able to run CINELERRA-GG. Before installing cygwin, you need to be warned that the Avast anti-virus software kills files necessary for cygwin installation and execution, so you will have to remove it and use alternative anti-virus software (the standard default already included with Windows 10 is Defender). Below are the steps for installation:

1. Download cygwin for your 64-bit computer at: https://www.cygwin.com/
2. Generally just take the defaults as they show up, but the next steps show what comes up.
3. When a warning window pops up, click *Yes*.
4. Click *Next*.
5. Choose *Install from Internet* option and then click *Next*.
6. Choose your desired directory by clicking on Browse button. Choose *All Users (Recommended)* and then click *Next*.
7. Choose the local package directory where you would like your installation files to be placed. Click *Next*.
8. Choose *Direct Connection* if you are using Internet with plug and play device. Click *Next*.
9. Choose any download site preferably "cygwin.mirror.constant.com" and then click *Next*.
10. For list of things to install, leave all set to *Default* except these to *Install* instead:

base	devel
gnome	graphics
system	video
X11	

    This install takes a long time; approximately 2 hours on an EliteBook and requires approximately 20GB storage.
11. Finally you will want to have the icons on your desktop (already default) and then click *Finish*.

Then to install the CINELERRA-GG tar files, you will need to start a cygwin console terminal from the startup menu as shown here: `Start` → `Cygwin` → `Cygwin64` Terminal

Installing CINELERRA-GG

1. Download the tar file libxcb-bld.tar.bz2.
2. Install libxcb from the tar file – installs into `/usr/local` and requires approximately 21MB storage.

   ```
   tar -C /usr/local -xJf /path/libxcb-bld.tar.bz2
   ```

   The libxcb patch repairs an error (XIOError), which stops Cinelerra.
3. Download the tar file cygcin-bld.tar.bz2.
4. Install cygcin from the tar file - this installs into home directory. Note this is cygcin *not* cygwin. You must change the `path` below to the name of the path where you downloaded the tar file.

   ```
   cd
   tar -xJf /path/cygcin-bld.tar.bz2
   ```

This creates `~/cygcin`, a user build installation of CINELERRA-GG and requires approximately 400MB storage.

Running CINELERRA-GG: You will need to start a cygwin desktop from the startup menu:

1. `Start`→ `Cygwin-X` → `Openbox`

   You should start a console controlling terminal so that you can see program logging.
2. `Start`→ `Cygwin` → `Cygwin64` Terminal

   This opens a separate window that can survive a cygwin hang and bugs. Without these logs, it is much more difficult to use.
3. Type into that console controlling window, the following:

```
export DISPLAY=:0.0
```

4. Change directories to where CINELERRA-GG is installed:

```
cd /path/cygcin (NOT cygwin)
```

5. Finally keyin:

```
./cin
```

   which starts up your 4 CINELERRA-GG windows.

The most noticeable difference from the Linux versions is that CINELERRA-GG seems to run very slowly on Windows 10. You must be very tolerant and patient to see this work. It can however exhibit astonishing speed when encoding. CINELERRA-GG has to be downgraded significantly due to lack of supported interfaces, codecs (for example h264/h265), and utilities. The only graphics driver is X11 and the only sound driver is pulseaudio. Almost all configurable omissions are applied to this build.

CINELERRA-GG build on cygwin from source code:

1. Download and install ffmpeg into /usr/local :

   download ffmpeg (currently 4.2.2)

```
cd /tmp
tar -xJf /path/ffmpeg-4.2.2.tar.bz2
cd ffmpeg-4.2.2
./configure
make -j
make install
```

2. Download and install a patched libxcb:

```
cd /tmp
rm -rf libxcb-1.13/
```

```
tar -xf /path/libxcb-1.13.tar.bz2
cd libxcb-1.13/
patch -p1 < /path/cinelerra-5.1/thirdparty/src/libxcb.patch1
    patching file configure.ac
    patching file src/xcb_in.c
./autogen.sh
./configure
make -j
make install
```

3. Download cinelerra-gg:

```
cd /build_path/
git clone git://git.cinelerra-gg.org/goodguy/cinelerra.git
cd cinelerra-gg/cinelerra-5.1
```

4. Apply cygwin patch:

```
patch -p2 < blds/cygwin.patch
```

5. Run the build with:

```
./blds/cygwin.bld
```

This produces a directory: /build_path/cinelerra-gg/cinelerra-5.1/bin which is used to create the cygcin archive.

Currently, the targets are not stripped and can be run from gdb. There is only very limited signal handler dmp file support. Running gdb from inside a desktop resident console (not a cygwin64 window) will hang cygwin (and cin) when it hits a breakpoint. You must run from an external console window to avoid this issue.

1.6 Android Tablet or Phone with TERMUX

CINELERRA-GG can be run on Android (without audio), a non-x86 mostly posix system, tablet or phone after installing TERMUX, the *terminal emulator*. You will have to do your own build using the file in Cinelerra's `blds` subdirectory, `termux.bld`. Because this is a relatively new capability and of lesser use, some additional effort

may have to be exerted on your part to get it going but it is easy to get help by contacting the mailing list. In addition, there is currently no known procedure for hearing audio.

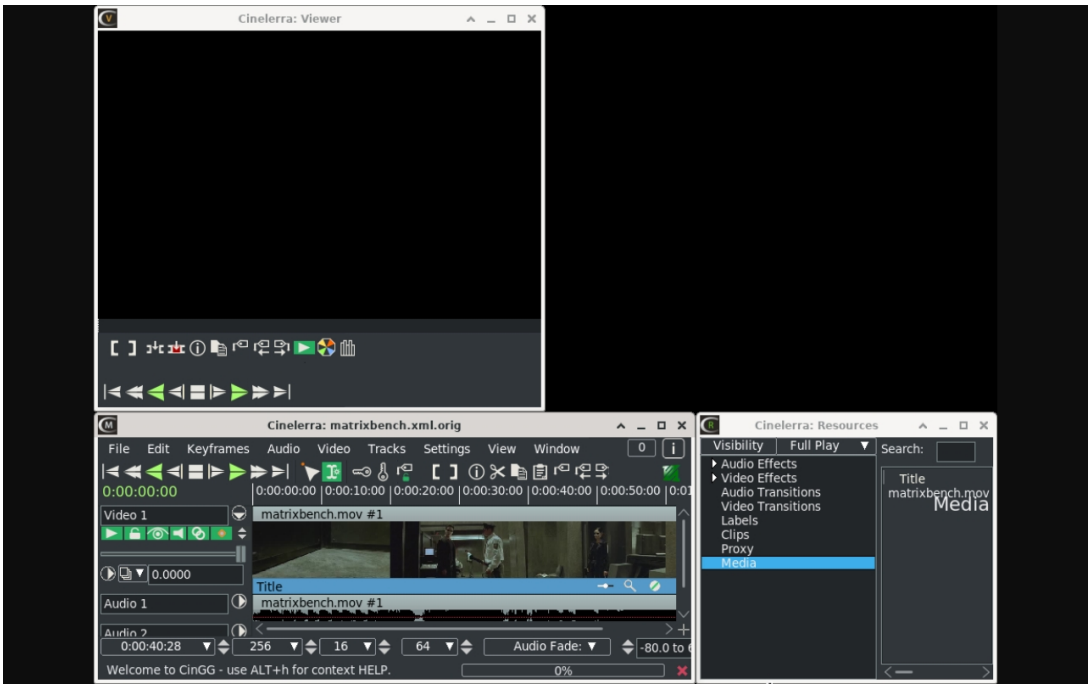


Figure 1.2: *Screencast of an Android tablet running* CINELERRA-GG *using TERMUX.*

Some requirements include;

1. Termux runs with X on Android 7+.
2. Install takes 5 GB of internal storage. In addition you can download videos, and other files with wget to one specific location at sdcard after running termux-setup-storage inside termux (it will prompt you to give access to sdcard graphically the first time used).
3. If you have empty versions of `locale.alias`, `locale.dir`, and `$PREFIX/share/X11/locale/en_US.UTF-8/XLC_LOCALE` you will have to request non-empty versions via the mailing list.
4. Some helpful information on installing the X environment is at: https://wiki.termux.com/wiki/Graphical_Environment
5. To prevent crashing when loading a video file that has audio, use the guide https://www.reddit.com/r/termux/comments/bpa8jz/pulseaudio_streaming_client/ which explains vnc/pulseaudio setup.

A little more about Audio is presented next because you will need to have this running in order to prevent a crash (even though you still will not be able to hear audio) – there does not seem to be a simple PA client in termux itself. Some information is available at: https://android.stackexchange.com/questions/205576/how-to-play-sound-from-termux-when-using-linux .

The next few lines show a successful setup/usage.

```
$ pulseaudio --start
$ ps axv
  PID TTY STAT TIME MAJFL TRS DRS RSS %MEM COMMAND
 7003 pts/28 S<s 0:00 637 532 9039 1716 0.0 /data/data/com
13684 ? S<l 0:00 0 49 123898 16616 0.8 pulseaudio --
13692 pts/28 R<+ 0:00 0 63 7500 1420 0.0 ps axv
```

```
$ pactl load-module module-native-protocol-tcp auth-ip-acl=127.0.0.1 auth-anonymous
    ↪ =116
$ PULSE_SERVER=127.0.0.1 pactl info
Server String: 127.0.0.1
Library Protocol Version: 34
Server Protocol Version: 34
Is Local: no
Client Index: 2
Tile Size: 65496
User Name: u0_a116
Host Name: localhost
Server Name: pulseaudio
Server Version: 14.2
Default Sample Specification: s16le 2ch 44100Hz
Default Channel Map: front-left,front-right
Default Sink: OpenSL_ES_sink
Default Source: OpenSL_ES_sink.monitor
Cookie: c659:c1b7
```

Now to start up CINELERRA-GG, type in:

```
$ cd (your cinelerra directory)/cinelerra/cinelerra-5.1/
$ PULSE_SERVER=127.0.0.1 ./cin.sh
```

You can even build a package version similiar to Debian, just with "`pkg search` pkg_name / `pkg install` pkg_name" instead of "`apt search/install` pkg_name"

and with "`*-static`" instead of "`*-dev/-devel` packages". For more information on this, see:

https://wiki.termux.com/wiki/Package_Management
https://wiki.termux.com/wiki/Building_packages

1.7 Pre-built Packages and Distros with CINELERRA-GG Included

There are also some special complete distribution systems available that include CINELERRA-GG for audio and video production capabilities as well as some pre-built packages in a build farm.

1.7.1 Build Farm pre-built Deb and RPM packages

Especially advantageous to the way some users prefer to keep CinelerraGG updated is the build farm for CINELERRA-GG deb and rpm packages that is maintained regularly at this location https://github.com/einhander/cin-gg-packages/releases. It will build packages on every git change in the main repo with releases corresponding to a build date, not a git commit date. Current build hosts are **Debian 12**, **Debian 11**, **Suse Leap 15.5**, **Fedora 38**, and **Ubuntu 22.04**. This build farm creates packages for the latest current versions of these distros with the potential for additional distros to be added in the future. In addition, packages for some previously built dates will still be available in case you want to revert to a specific package date. You might have to install libsuil-0-0 (library for loading and wrapping LV2 plugin UIs) but you will be informed by the installation process if so.

1.7.2 AV Linux

AV Linux is a downloadable/installable shared snapshot ISO image based on MX Linux. It provides the user an easy method to get an Audio and Video production workstation without the hassle of trying to find and install all of the usual components themselves. Of course, it includes CINELERRA-GG! Click here for the homepage of AV Linux.

1.7.3 Bodhi Linux Media

Bodhi Linux Media is a free and open source distribution that comes with a curated list of open source software for digital artists who work with audio, video, includes CINELERRA-GG, games, graphics, animations, physical computing, etc. Click here for the homepage of Bodhi Linux.

1.7.4 DeLinuxCo

DeLinuxCo is a distro derived from Manjaro (so Arch based) with DE Cinammon. It is a professional workstation, mainly oriented to the multimedia field but not only. It contains many specialized programs already configured, including CINELERRA-GG.

You can read all about DeLinuxCo here and download here.

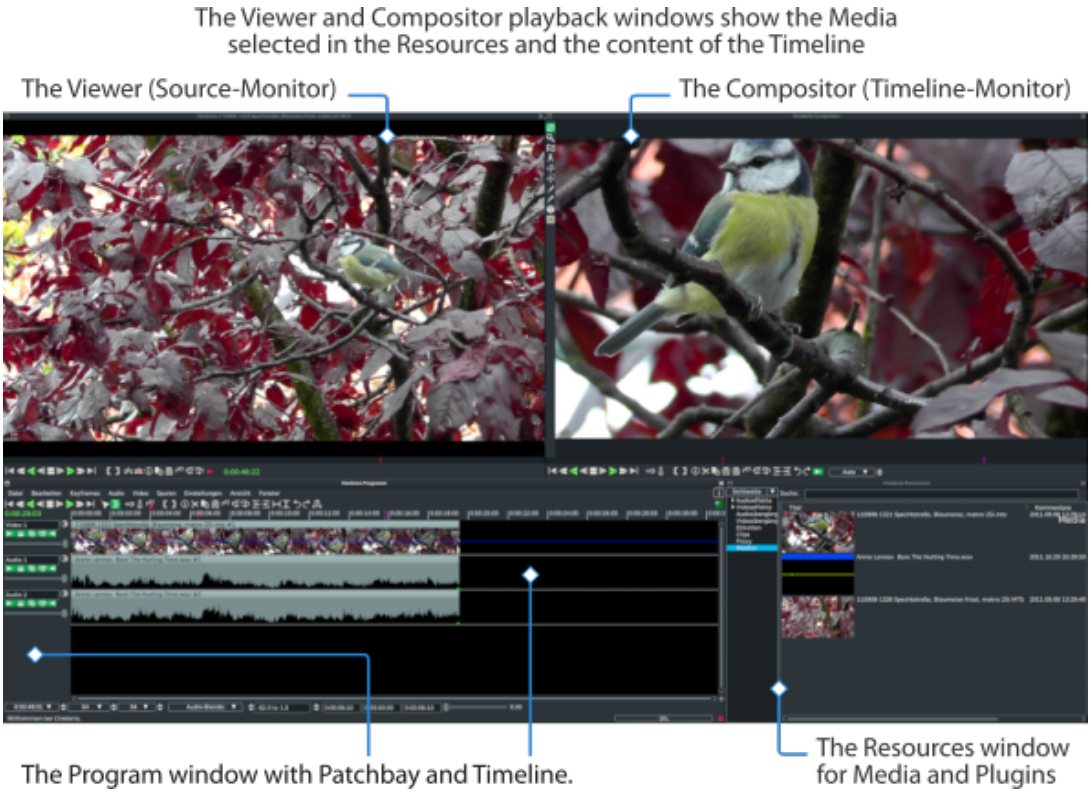
1.7.5 Elive

Elive, or Enlightenment live CD, is a non-commercial, cost-free operating system based on Debian, and it can be used either as a live CD or an Installed system. Elive uses a customized Enlightenment desktop. It is fast, user-friendly and feature-rich and CINELERRA-GG is included in the both the 64 bit and 32 bit versions.

Click Elive for more information. The CINELERRA-GG packages for the program and the manual are in the direcotry at Bookworm version 12 and Bullseye version 11 and Buster version 10 - just download the .deb files inside that directory and install via “dpkg -i “.

CHAPTER 2

Interface (the 4+ Windows)



First, it is important to know what an EDL is. When CINELERRA-GG saves a file, it saves the EDL, Edit Decision List, of your project which contains all the settings and locations of edits and pointers to the media so that the media is not modified. The EDL is described in EDL — Edit Decision List.

Second, for those users who would like to keep all 4 windows together for a better workflow, see Focusing the 4 main windows as a group.

2.1 Program Window

The main window is called the *Program* window and is often just referred to as the *timeline*. Here is where you enter the main menu operations. This timeline consists of a vertical stack of tracks with time represented horizontally on the track. It is the output of the rendering operations and this is what is saved when you run the *File* pulldown, Save command. Immediately to the left of the timeline is the patchbay. The patchbay contains options that affect each track. These options are described in great detail in The Patchbay.

The *Window* pulldown on the main window contains options that affect the 4 main windows. The first 3 options are used to display each of the windows in case one was accidentally closed. You can move or resize the windows as needed, save that particular layout, and revert to the default positions to reposition all 4 windows to the original screen configuration. On dual headed displays, the *Default positions* operation only uses the one monitor to display the windows, but as you can see in the *Window* pulldown you have more options to change that with the 2 *tile* options. Usage with dual monitors is explained in Playback A / Playback B.

2.1.1 Video and Audio Tracks and Navigation

The *Program* window (figure 2.2) has many features for navigation and displays the timeline as it is structured in memory. The tracks are stacked vertically with horizontal movement over time. There is a vertical scroll bar which allows for moving across tracks and a horizontal scroll bar for scanning across time.

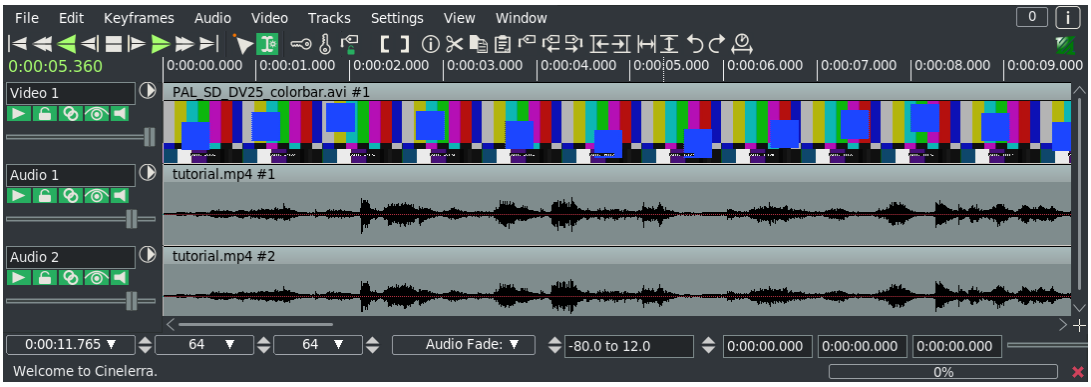


Figure 2.2: *Patchbay | Timeline with pulldowns, navigation icons, Video/Audio tracks | bottom Zoom Panel*

Video tracks represent the duration of your media, just as if you placed old-fashioned rolls of photographic film one right after the other on a table. Individual images that are drawn on each track are snapshots of what is located at that place on the timeline.

Audio tracks represent the sound media as an audio waveform, or if you change a preference setting, a rectified audio waveform. This too looks like old-fashioned digital magnetic tape laid out horizontally across a table. Using the *Zoom Panel* controls at the bottom of the timeline, you can adjust the horizontal and vertical size of the video and audio waveform displays. Each track on the timeline has a set of attributes on its left side in the patchbay which is used to control some options of that particular track.

Track navigation is performed by selecting a video or audio track and moving to a certain time in the track. Use the vertical scroll bar to scan across tracks, or even easier you can use the mouse wheel. And use the horizontal scroll bar to scan across time, or again even easier, you can use the mouse wheel with the Ctrl key.

Once you have become familiar with many of the graphical tools and pulldowns, you can switch to using more of the keyboard to navigate. Many of the key equivalences are listed in each of the pulldowns to the right of the option as a shortcut. All of the shortcuts are listed in a document for keyboard navigation (Shortcuts). This includes, for example, shortcuts like the `Home` and `End` keys to go to the beginning or end of the timeline. Another example is in the default cut and paste mode, hold down `Shift` while pressing `Home` or `End` in order to select the region of the timeline between the insertion point and the key pressed.

2.1.2 Transport and Buttons Bar

This is a short description of what is contained on the *Transport and Buttons Bar* as seen in figure 2.4. Each of the symbols has an associated tooltip as an easy reminder of its function. The usage of several is described in a little more detail in other areas of this manual.

First set of 3 symbols	Move in the reverse direction on the timeline
4th symbol	Stop play
Third set of 3 symbols	Move in the forward direction on the timeline
Next 2 symbols	Set editing mode
Generate keyframes while tweaking	Explained in 8.8
Allow keyframe spanning	Explained in 8.11
Lock labels from moving with edits	Toggle to lock or not lock labels from moving
Next 2 are In and Out pointers	Set or unset In and Out to define an area
To Clip	Create a clip of the designated area
Split / Cut	Split/Cut at insertion or selected area
Next 2 symbols	Usual Copy and Paste functions
Next 3 symbols	Label manipulation functions
Next 2 symbols	Used to move to other edits
Next 2 symbols	Fitting video to the display

Next 2 symbols	Undo and Redo last operation
Manual Go To	Explanation after this table
Set Timecode	Explained in 6.8.2
Gang Modes	Explained in 5.2.1

The *Manual Go To* menu is quite versatile as you can see in figure 2.3. The options of = represents goto the position indicated in the textbox; + indicates to goto forward the additional number in the textbox; and - means to go in the reverse direction the number in the textbox. If you use *Enter* in the textbox, the menu will stay up so that you can use it without having to continuously pop it up. If you use the OK check instead, the menu will be taken down. The pulldown arrow to the right of the textbox sets the *Time Format* for the units to be used for position changing. This lets you change the units without having to modify your Preference.

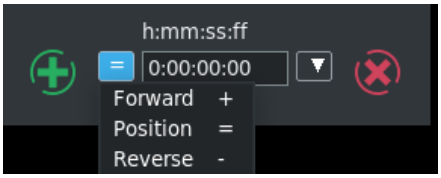


Figure 2.3: *Goto position menu*

Time Format options are:

h:mm:ss.sss	hours minutes seconds milliseconds
h:mm:ss:ff	hours minutes seconds frames
timecode	
video frames	
audio samples	
audio samples (hex)	
ssss.sss	milliseconds.xx
video frames (feet)	

2.1.3 Zoom Panel

Below the displayed tracks in the timeline, you will find the zoom panel as seen in figure 2.2. In addition to the scrollbars, these options and their values are another set of tools for positioning the timeline. In order of appearance in the zoom panel as rectangular boxes and either tumbler arrows or a up/down arrow, this next list shows each option along with its tooltip description if available. Then more details are provided in the next paragraphs.

Sample zoom	Duration visible on the timeline
Amplitude	Audio waveform scale
Track audio zoom	Height of audio tracks
Track video zoom	Height of video tracks
(type)	Automation Type
Curve zoom	Automation range minimum and maximum
Selection change	3 boxes with starting point, length, and ending point
Alpha slider	Slider bar to control alpha value for colored assets

Changing the *sample zoom* changes the amount of time displayed on the timeline so you can see your media as individual frames or as the entire length of your media. To see more frames, use a higher setting. The sample zoom value is not an absolute time reference because it refers to the duration visible on the timeline. It will change as you modify the length of the program window horizontally. You can either use the ↑ and ↓ arrows to change the sample zoom by a power of two, or use the mouse wheel on the tumblers to zoom in and out.

The next option is *amplitude* and it only affects the audio waveform size. `Ctrl-`↑ and `Ctrl-`↓ are shortcuts used to change the amplitude zoom as an alternative to the down arrow to the right of the numerical size.

The *track audio and video zoom* affects all tracks of that type and determines the height of each track. If you change the audio track zoom, the amplitude zoom will be changed also so that the audio waveforms are proportionally sized. Shortcuts, `Ctrl-Pgup` and `Ctrl-Pgdown`, change the track zoom to the next level simultaneously for all of the audio and video tracks.

Automation type is used for selecting one of the following: Audio Fade, Video Fade, Zoom, Speed, X, or Y (X and Y are for the compositor's Camera and Projector). When an auto line is present on the timeline and is being manipulated, a small square the same color as the line will be shown to the left of the Automation type when the left mouse button (LMB) is pressed. This is just an indicator to make it easy to see what is being worked.

The *curve zoom* affects the curves for the selected *automation type* in all the tracks of that type and determines the value range for those curves. Use the tumbler arrows to the left of the numbers for the minimum value and the tumblers to the right for the maximum value, or manually enter the values in the text box. Good default values for audio fade are -40.0 to 6.0 and for video fade are 0.0 to 100.0. The tumbler arrows change curve amplitude, but the only way to curve offset is to use the fit curves button on the curve itself.

The *selection start time*, *selection length*, and *selection end time* display the current

selected timeline values. When there is no selection, both the start and end time are the current position of the timeline and the selection length is 0. The *alpha slider* allows for varying the alpha value when using colors on the tracks as set in your `Preferences` → `Appearance` for `Autocolor assets`. It has no function without that flag set.

There are 3 additional pieces of information in the line immediately below the *zoom panel*. In the lower left hand corner there could be messages such as "Welcome to CINELERRA-GG" when there is no need to display a red-colored error message or a line that reads "Rendering took H:MM:SS" after a render has just been completed. Or when working with an auto, a small square the color of the last changed auto line, will be present along with its keyframe type, location on the timeline, and its current value. This is simply for easy recognition of what is being worked. The second piece of helpful information is all the way to the right which is a long rectangular box indicating the percentage completion of a render. Finally there is an X with the tooltip of "Cancel operation" used to stop an ongoing render (the cancel operation may seem slow due to the amount of data still in the buffer upon cancellation).

2.1.4 Track Popup Menu

Each Track has a popup menu. To activate the track popup menu, Right mouse click (RMB) on the track. The popup menu affects the track whether the track is armed on the patchbay or not. The Track Menu contains a number of options:

Attach Effect opens a dialog box of effects applicable to the type of track of audio or video.

Move up allows for changing the order of the tracks up; the actual code refers to this as a swap. Disarmed tracks affect the results.

Move down allow for changing the order of the tracks down; the actual code refers to this as a swap. Disarmed tracks affect the results.

Roll up moves the tracks one step up in the stack with the top track going to the bottom. This is applicable to all armed and disarmed tracks.

Roll down moves the tracks one step down in the stack with the bottom track going to the top. This is applicable to all armed and disarmed tracks.

Delete track removes the track from the timeline.

Add Track adds a track of the same media type as the one selected, audio or video, above the selected track.

Find in Resources the media file on that track at the location of the insert pointer will be highlighted in the media folder in the Resources window. If the Resources window is closed, media is found and highlighted but the Resources window is not displayed.

Show edit will point out the exact start and stop points along with the length of the current edit on that track as well as the media name, track name and number, and edit number.

User title is used to change the title name. This is really handy for files that have very long and similar names that would get cut off during edits. You can use short names to better differentiate the media. In Drag and Drop editing mode, if you select multiple edits all of those clips will have their title name changed.

Bar color allows the user to select a specific color for the title bar. This helps to more easily locate a piece of media.

Resize Track resizes the track; this is only applicable to video tracks.

Match Output Size resizes the track to match the current output size; this is only applicable to video tracks.

2.1.5 Insertion Point

The insertion point (figure 2.4) is the vertical hairline mark that spans the timeline in the program window - it can be a solid line but most of the time it will be flashing. Like the cursor on a word processor, the insertion point marks the place on the timeline where the next operation will begin. It is the starting point of all play operations and is the point where a paste operation will occur. In some cases, when rendering it defines the beginning of the region of the timeline to be rendered.

To move the insertion point, you move the mouse inside the timebar area and click with the left mouse button. You can use any place on the timebar to reposition the insertion point as long as that spot is not blocked by In/Out point or a label . In cut and paste editing mode, you can also change the position of the insertion point with a simple left mouse click in the timeline itself. When moving the insertion point, the position is either aligned to frames or aligned to samples. For best results, *Align cursor on frames* when editing a video track and *Align to samples* when editing audio. Use the pulldown `Settings→Align cursor on frames` to change the alignment by checking the box on for video and off for audio.

2.1.6 Editing Modes

There are 2 different editing modes for operations which affect how the insertion point and editing on the timeline operate. There is: *drag and drop mode* and *cut and paste mode*. The editing mode is determined by selecting the `arrow` , or immediately to the right of the arrow, the `I-beam` in the Transport and Buttons bar. In figure 2.4 you can see the green colored highlight [1] on the arrow icon indicating that you are currently in *drag and drop mode*.

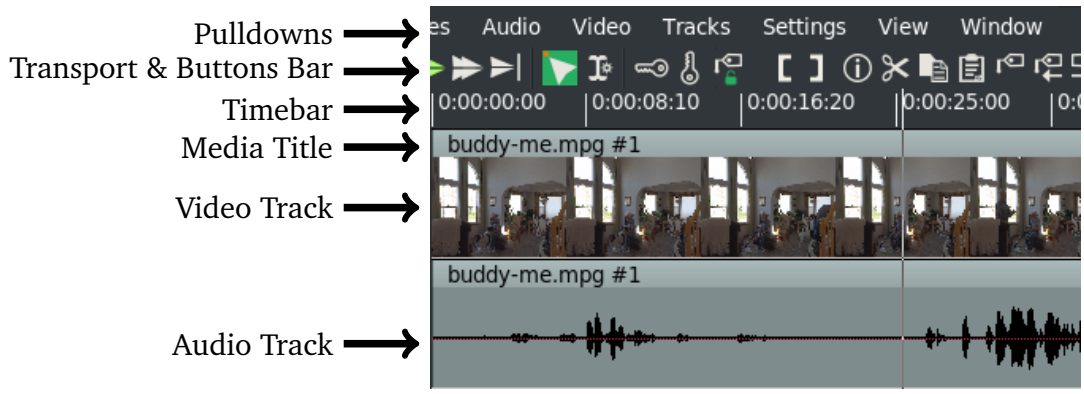


Figure 2.4: *Insertion point is at 0:00:25:10 in Hr:Mn:Sec:Frames*

With the arrow highlighted for *drag and drop mode*, a double click with the left mouse button in the timeline selects the edit the mouse pointer is over. Then dragging in the timeline repositions that edit and this can be used for moving effects, changing the order of playlists, or moving video pieces around. There are numerous methods to cut and paste in *drag and drop mode* by setting In/Out points to define a selected region or using the Copy/Paste Behavior as outlined in 5.5.1. In this mode, clicking the LMB in the timeline does not reposition the *Insertion Point*.

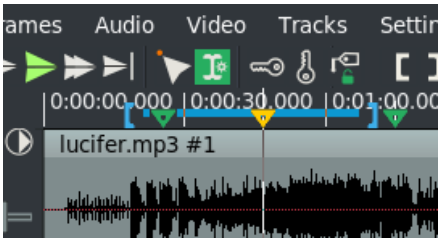


Figure 2.5: *I-beam + in/out + labels*

When the I-beam is highlighted, you are in *cut and paste mode*. In cut and paste mode, clicking the LMB in the timeline does reposition the *Insertion Point*. Double

[1] green is used in the default Cakewalk theme, but the highlight color will be different in other themes

clicking in the timeline selects the entire edit the cursor is over, i.e. that column. Dragging in the timeline with the LMB pressed down, highlights a selected region and this is the region that is affected by cut and paste operations. It is also the playback range used for the subsequent playback operation. Holding down the Shift key while clicking in the timeline extends the highlighted region.

2.1.7 In/Out Points

The In/Out points, displayed on the timebar by [and] brackets, can be set in either of the editing modes to define the selection. In the timebar, a colored bar will show between these 2 brackets to better outline the area selected. In *drag and drop mode* , they are an easy way to define a selected region.

It is important to remember that a highlighted area overrides the In/Out points. That is, if a highlighted area and In/Out points are both set, the highlighted area is changed by editing operations and the In/Out points are ignored. But if no region is highlighted, the In/Out points are used. To avoid confusion, use either highlighting or In/Out points but not both at the same time.

To set In/Out points, in the timebar move to the position where you want the Insertion point and click the In point icon or one of the [or < keys. Then move the insertion point to a position after the In point and click the] or > or the Out point icon. You can use these same icons or keyboard characters to toggle In/Out points on or off.

If you set the insertion point in another place when In/Out points are already set, that existing point will be repositioned when you click the In/Out icon or keyboard equivalent. If you click on In/Out points while a region is highlighted, the insertion point will be ignored and In/Out points will be set at the beginning and at the end of the highlighted area.

When you select either the In or Out point on the timebar, the insertion point will move to that location. Note that when the insertion point is at the exact position of an In or Out point, the bracket will change color making it easy to see that you are exactly at that spot.

If only the In point is set, when you click the In point icon the In point will be deleted. If only the Out point is set, when you click the Out point icon the Out point will be deleted. Holding the Shift key while clicking on an In/Out point, the area between the insertion point and that In/Out point will be highlighted or extended to that In/Out point if already highlighted.

An easy way to turn off the In/Out points if both are set, is to double click on

the [icon in the toolbar. If you have already set the In and Out points, and then move the insertion point anywhere to the left of the Out point, a LMB click on the [icon will move the In point to the location of the insertion point. In the same manner if you move the insertion point anywhere to the right of the In point, a LMB click on the] icon will move the Out point to that new position. However, if you move the insertion point for either the In or Out point beyond what makes sense to designate In/Out points, the bracket you clicked on will be moved to the insertion point and the other bracket will be eliminated. That is obviously because the In point has to come before the Out point on the timeline.

Some of the useful operations concerning the In/Out pointers are listed next.

Ctrl-KeyPad# if In/Out set, `KP 2,3,5,6 + Enter`, play between In/Out points

Shift-Ctrl loops play between In/Out points

Click In/Out while holding the LMB down, drags In/Out pointer where you drag to

Shift-Ctrl with a transport button (e.g. Fast Forward), loops play between In/Out points

Ctrl-t clears both In/Out points

2.1.8 Guide on timeline

It can be useful to have visual reference points on the timeline to help you position edits relative to other tracks. For example in manual synchronization of audio and video. You can set two points on the timeline with In/Out points or by selecting a range in Cut and Paste mode. Now we use `Settings` → `Loop Playback` (Shift - L) to make two vertical green lines appear at the edges of the selection. Originally this function is used to limit the playback within this range, but we can take advantage of the fact that the green lines remain even when we move the insertion point or delete the In/Out points or deselect the range. So they can be used as a visual reference when we want to move the edits precisely. To delete these guides press *Shift-L* again.

2.1.9 Labels

Labels are used in order to set exact locations on the timeline that you want to be able to easily get to. To create a label, position the insertion point at a location and click on the label icon in the Transport and Buttons bar. The new label is displayed on the timebar as a down arrow at that location as shown in figure 2.5. Whenever the insertion point is at the same position as a label, it changes color to emphasize

that it is exactly at that spot. Labels make it so you can jump back and forth to exact marked locations on the timeline. Use the lower case letter "L" as a shortcut for the label button.

You can use labels to reposition the insertion point when that label is selected. They are also especially useful for moving along the timeline to the *Next label* or *Previous label* with the buttons on the Transport and Buttons bar to the right of the Labels button. When moving along the timeline with the Next or Previous label buttons, if a label is out out of view the timeline will automatically be repositioned so that the label is visible. If you perform a *Next label* operation and there are no more, the insertion point will go to the End position. Conversely if you perform a *Previous label* operation and there are no more labels, the insertion point will go to the Home position. Keyboard shortcuts for label traversal are:

Ctrl-left moves the insertion point to the previous label.

Ctrl-right moves the insertion point to the next label.

There is a Label folder in the Resources window which has a list of every label and its exact location where the label is. The location is based on the timestamp, frame number, or sample number depending on the selected Time Format of your timebar. You can edit, delete, or goto a label by clicking the RMB on that label in the Resources window which brings up a popup menu with those options. It can be quite helpful to *edit* the label and add a text string to help identify what the label represents. In addition, RMB clicking the label symbol on the timebar brings up a textbox displaying the current text string and allowing you to change it. If a Label has been given a name, simply mousing over the label symbol on the timebar will display that string.

With labels you can also select regions:

Shift-Ctrl-left highlights the area between the insertion point and the previous label.

Shift-Ctrl-right highlights the area between the insertion point and the next label.

Double-clicking on the timebar between two labels, highlights the area between the labels.

Shift-clicking on a label, highlights the area between that label and the insertion point. If an area is already highlighted, it extends the highlighted area up to that label.

If you LMB click the label button when an area is highlighted, labels are created at each end of the highlighted area. When a label is selected, if you click on the label icon, the label will be deleted. To delete multiple labels, highlight that area, then use the `Edit` → `Clear` → `Clear labels` function to delete them all. The same precedence rules apply to this operation as mentioned earlier. That is, if both In/Out points are set and there is a highlighted area also set, the highlighted area's labels will be cleared and not those between the In/Out points.

If you enable *Edit labels* in the *Settings* pulldown menu or disable the *Lock labels from moving* button on the Transport and Buttons bar, labels will be cut, copied or pasted along with the selected area of the first armed track. In the same manner, if a selected area of media is spliced from the viewer to the timeline in a position before labels, the labels will be moved to the right on the timebar so that the label maintains its relative position to its edit. To prevent labels from moving on the timebar, enable the *Lock labels from moving* icon or disable the *Edit labels* option under the *Settings* pulldown.

2.1.10 Color Title Bars and Assets

In order to visually aid in locating clips on the timeline that are from the same media file, you can have them auto-colored or self-colored. Auto-color is an automatic system to always color the files on the timeline when loaded. Self-color is a manual coloring requiring the user to take definitive action. Use of this feature requires additional memory and cpu on every timeline redraw, therefore it is recommended that smaller computers leave it turned off.

For auto-color the color will be based on a hashed filename so that whenever you load this particular media, it will always have the same color on the title bar even if you use proxy. To enable auto-color go to `Settings` → `Preferences`, `Appearance tab` and check on `Autocolor assets`. You will see this in the Flags section as shown in Figure 16.1. It is disabled by default. Each media will have a random muted color and there could easily be close duplicates as generated by the program algorithm. There will be no total black, but some dark shades are possible.

To change a specific clip to your own chosen color, right mouse button (RMB) over that clip and an Edits popup will be displayed. Choose the option *Bar Color* to bring up the color picker and choose a color. You can also change the alpha value in the color picker and this alpha takes precedence over the current alpha slider bar value unless the color picker's alpha value is set to 1.0. The color will only change after you click on the checkmark. The *Bar Color* option works in either Drag and Drop or Cut and Paste editing mode and also works if *Autocolor assets* is not set. In Drag and Drop editing mode, if you select several clips (reference "Create a group" in

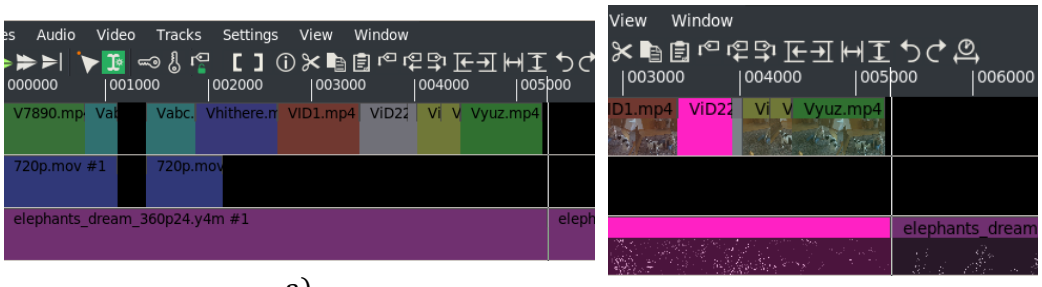
5.5.4) and then bring up the Edits popup with the right mouse button over a track, you can use the *Bar Color* option to change all of those selected to the same color.

To go back to the default colors, uncheck *Autocolor assets* in Preferences, but this does not affect the specially chosen self-colored ones as they are preserved. To change these individually or selectively, use the Edits popup *Bar Color* option and click on *Default* in the color picker window.

And that's not all! There is an *alpha fader slider bar* on the bottom of the main window on the right hand side of the Zoom Panel. With this alpha slider, you can colorize your video and audio tracks to either see only the color at 0.0 or see only the image at 1.0. This slider bar affects all colored areas of the Autocolor assets and the self-colored ones. In the case when a specifically changed edit alpha value is set in the color picker to any value except 1, the slider bar will not affect that. Once you use the slider bar, it is activated so gets first shot at any keystrokes in the main window. You deactivate this by simply clicking in a different part of the main window.

As long as we are on the subject of color, just a note that you can also change the *Highlighting Inversion color* in `Settings → Preferences, Appearance tab`. You can see this option in Figure 16.1 in the Color section. That setting defaults to white (*f f f f f f*) but sometimes this is a little bright so you can put any hex value in that suits you.

This image (figure 2.6a) shows an example of the Autocolor assets with alpha set to 0.0 so that you see only the color and no image. In this image (figure 2.6b), the alpha is set to show the image and waveforms with transluent colors. The pink media file has been self-colored rather than the autocolor to make it easy to see.



a) b)

Figure 2.6: *An example of the Autocolor assets*

2.1.11 More about Pulldowns

The main window pulldowns as pointed out in figure 2.4 are quite obvious in their meaning and usage, so here is only a summary.

File options for loading, saving, and rendering as described in other sections (4).

Edit edit functions; most of which have shortcuts that you will quickly learn (5).

Keyframes keyframe options which are described in the Keyframe section (8).

Audio audio functions such as *Add track*, *Attach effect* and *Attach transition*. The *Attach effect* is especially useful when you need the effect to be applied to all related audio tracks as a *Shared effect* and is described as an alternative method of application in section 10.4.

Video video functions such as *Add track, Default/Attach transition, Render effect*.

Tracks move or delete tracks are the most often used plus *Align Timecodes*.

Settings much of this is described elsewhere with the most frequently used to include Preferences (16), Format (3), Proxy and Transcode (19.3), as well as the others.

View for display or modifying asset parameters and values to include Fade, Speed, and Cameras.

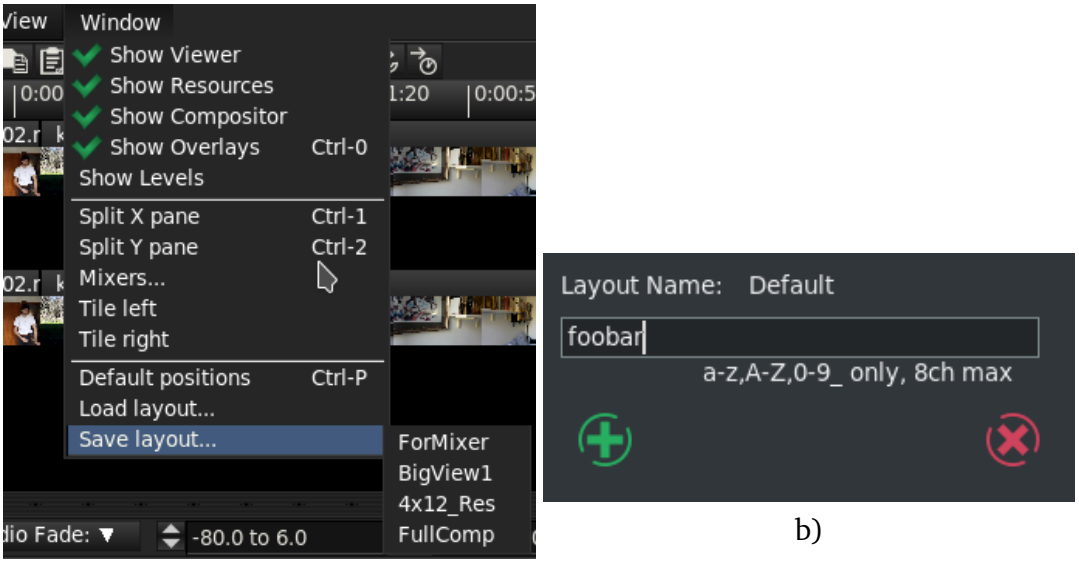
Window window manipulation functions.

2.1.12 Window Layouts

If you like to use different window layouts than the default for certain scenarios, you can setup, save, and load 4 variations. First, position your CINELERRA-GG windows where you want them to be and then use the *Window* pulldown and choose *Save layout*. Note the words *Save layout* highlighted in Figure 2.7a with 4 names shown to the right and below of that highlight. To use the default name of *Layout #*, when the popup comes up, just click the green checkmark OK on the Layout popup menu. If you would like a specific name for your layout so you can remember what its best use case is, keyin 1-8 english characters that are meaningful to you (english characters mean you can not use the German umlaut, the French accent, or the Spanish ñ). Legal characters are a-z, A-Z, 0-9, _ (the underscore character) and a limit of 8 total. If you keyin more than 8, only the last 8 characters will be used. To rename a currently existing layout, use the *Save layout* option again on the one to rename, and keyin a different name into the text box or leave blank for the default name (figure 2.7b).

The files containing the coordinates for your layouts will automatically be saved in the `$HOME/.bcast5` directory as `layout#_rc` or `layout#_8chars_rc`.

To use the desired layout, keyin the shortcut or use the *Window* pulldown and choose *Load layout* and then make your choice. It is very beneficial to learn the



a) b)

Figure 2.7: *Window Layouts*

the associated shortcut layouts (Ctrl+Shift+Fx where x is 1,2,3, or 4) because they can be executed from any of the 4 windows instead of just the main timeline window.

2.1.13 Multi-Pane Support

The main CINELERRA-GG edit window holds the Track Canvas which can be divided into 4 panes of track data: 1 or 2 vertical panes and/or 1 or 2 horizontal panes. To split the track, use the *Window* pulldown, and then click on *Split X* or *Split Y* depending on how you wish to split the track. Alternatively, the canvas pane types can be changed using keys *<Ctrl-1>* for toggle split horizontal or *<Ctrl-2>* for toggle split vertical. Or the track can be split into panes by using the *+ widget* in the lower right hand corner of the track canvas. Once the track has been divided, you can use the + widget shortcut or the drag bars to change the size of the panes.

Multi-Pane, or split screen, allows you to look at the first part of a movie at the same time as a part that is a long ways away on the timeline which would have been off the screen. By having multiple panes, you can see the 2 parts you want to look at simultaneously and drag/drop easily between the 2. Also, the *X pane split* is extremely convenient for laptop users and computer monitors with small screens since it can be used with horizontal scrolling with the mouse wheel + Ctrl. The *Y-pane split* makes it easy to see 2 simultaneous drag and drop zones when you have lots of tracks (figure 2.8).

Figure 2.8: *Shown are 4 panes that have split in X and Y the main track canvas*

2.1.14 Just Playing!

What if you are just using CINELERRA-GG to play media and listen to tunes? After loading your media, just hit the space bar to start playing and then again to stop playing. Other than that, use the transport buttons on the top bar of the Program window. Other ways to *play around* are described next.

Repeat Play / Looping Method

There are 2 methods for repeat play or looping on the timeline and 1 method for both the Compositor and the Viewer. This works in conjunction with any of the transport buttons or shortcuts in either forward or reverse as usual. The 1 exception is that the Shift key can not be used to either add or subtract audio within the repeat area.

Method 1: Shift-L on the Timeline, repeats the selection per the algorithm outlined next.

When setup, long green lines are displayed across the entire set of tracks which shows the start and end of the loop.

1. Highlighted selection repeats loop and takes precedence over all other possibilities. If the cursor is before the highlighted area, it will play up to the area and then repeat the highlighted section. If the cursor is after the highlighted section, play will start at the beginning until you get to the highlighted section and then repeat.
2. When both In and Out pointers are set, it repeats the section between [and].
3. If only one of the In or Out pointers is set, it loops the whole media.

Method 2: Ctrl+Shift+transport button on the Timeline, Viewer, and Compositor

1. Repeats entire media if no In or Out pointer set.
2. In and Out pointer set, repeats area between pointers.
3. Only In pointer set, repeats from In to end of media.

Last Play Position Memory

When you play media, the start/end playback positions are saved as if they had been made into temporary labels. They appear on the timeline as purple/yellow hairline markers representing the last start/end labels for the last playback. They can be addressed as if they are label markers using:

Ctrl← tab to the label before the cursor, that is *play start*

Ctrl→ tab to the label after the cursor, that is *play stop*

You can use these markers for re-selection. Additionally, the selection region can be expanded by *pushing* the markers using single frame playback. Use frame reverse (`keypad 4`) to push the start play marker backward, or use frame forward (`keypad 1`) to push the end play marker forward.

Another handy feature is to use the combination of Ctrl-shift-arrow (left or right) to select the media from the cursor position (red hairline) to the start or end marker by *tabbing* to the label markers. For example, tab to the beginning of the previous play region using Ctrl-left-arrow to move the cursor to the beginning of last play, then press Ctrl-Shift-right-arrow to tab to the end of the playback region. Now you can clip/play/expand or edit the previous playback selection.

Ctrl SHIFT→ tab cursor to label right of cursor position and expand selection

Ctrl SHIFT← tab cursor to label left of cursor position and expand selection

Playback Speed Automation Support

The speed automation causes the playback sampling rate to increase or decrease to a period controlled by the speed automation curve. This can make playback speed-up or slow-down according to the scaled sampling rate, as *time is multiplied by speed* (Speed × Unit_rate). For more information on changing the speed, read the section on Speed Automation 8.3.

Alternative to using Numeric Keypad for Playing

For the keyboards without a numeric keypad or if you prefer to use keys closer to where you normally type, there are alternative keys for the play/transport functions. These are listed below.

Alt + m	=	stop playback
Alt + j	=	forward single frame
Alt + k	=	forward slow playback
Alt + l	=	forward normal playback
Alt + ;	=	forward fast playback
Alt + u	=	reverse single frame
Alt + i	=	reverse slow playback
Alt + o	=	reverse normal playback
Alt + p	=	reverse fast playback

+ Shift key, single frame forward/reverse will include audio; all others will not include audio.

+ Shift + Ctrl, results in the transport function operating only between the in/out pointers.

2.2 Compositor Window

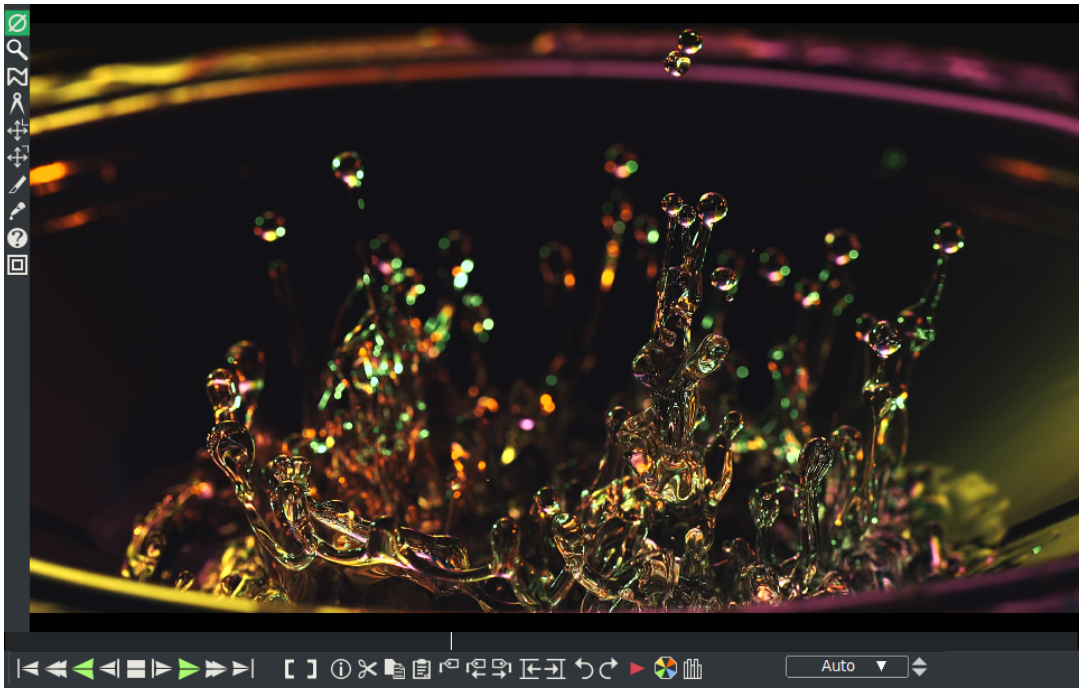


Figure 2.9: *Left hand side are the toolbar functions / bottom bar has many control functions*

The Compositor window (figure 2.9) is used to display the timeline output. Playing and moving along the timeline video in the Program window shows in the Compositor window what the current image is. Here is where many compositing operations are performed that can change what the timeline will look like and that is why it is called *canvas*. The canvas dimesions correspond, by default, to the

W/H values set in the project (*Set Format* window). Canvas size = output size = project size.

When enabled, you can simply click the LMB in the Compositor window to start and stop play. You can zoom in and out to see small details, pan with the scrollbars, lock the window to prevent changes, add masks, and make changes with the Projector and Camera function operators. These will be explained in more detail in the following sections.

2.2.1 Compositor Controls

On the bottom of the window, there are many of the same transport buttons and controls that are available in the Program window. They work the same as in the Program window and also have tooltips that are visible when you mouse over each of the icons so their use is fairly obvious. However, of particular note is the button *Click to play* which is described in 2.4.1. Next is the *Videoscope* button which is used to enable the scopes window without having to apply the filter to the tracks/edits.

Next to all of these controls all the way to the right side, there is a *zoom menu* and a *tally light*. The *zoom menu* has a pulldown with different settings that you can choose from or you can just use the tumbler arrows to the right. Generally when just getting started, you will be using the default *Auto* option. The window size is not changed, but rather the size of the video itself. In addition there are many shortcuts for zooming that you will find in the Shortcuts chapter (15).

Using a RMB click in the compositor window brings up a menu with several options . One of these is *Resize Window* which allows for resizing the entire window instead of just the video and if you highlight that, you can choose from several variations for the window size. Other available options include self-explanatory ones of *Fullscreen* with "f" as a toggle to go to fullscreen and to revert to non-fullscreen and *Zoom Auto*. As you would expect, whenever the video is zoomed so that only part of the image is visible in the window, scrollbars are automatically added as needed on the bottom, the right hand side, or both. Other options include *Reset camera* and *Reset projector* which obviously are used to reset the camera and the projector (reference 2.2.5). And *Camera/Projector keyframe* will create a keyframe at that point on the timeline for X,Y, and Z without the requirement of being in *Automatic Keyframe Mode*. More information is described in the section Compositor Keyframes. The *Hide controls/Show controls* option is great for hiding the left hand toolbar and bottom set of controls for a cleaner look.

Next to the zoom tumbler arrows, is a *tally light* that will be filled in with some color (often red or blue) when a rendering operation is taking place . This is especially helpful when loading a very large video so you know when it is finished

loading. You should pay attention to this *tally light* when performing a particularly time-consuming operation so that you do not keep executing more operations that just have to wait until completion of that CPU intensive operation. Also, you should look to see if the light is on before assuming that CINELERRA-GG is hung up.

When the window is unlocked, meaning that it is not in *Protect video from changes* mode on the toolbar, MMB clicking and dragging anywhere in the video pans the view. Panning can also be accomplished with the bottom and right hand side scroll bars when displayed.

2.2.2 Compositor Toolbar

On the left hand side of the Compositor window, there is a toolbar with several icons that provide functions for viewing and compositing the video. Each of these operational features will be described in more detail next.

Protect video from changes this option makes it possible to disable changes to the compositor output when clicking on the Compositor window. It allows for using the *Click to play* button (when enabled) for simply starting and stopping play. It helps to prevent an accidental click from making unwanted changes. When you enable this option, any of the other enabled tools will automatically be disabled.

Zoom view / magnifying glass when enabled, the *Zoom view* immediately results in the addition of a zoom slider for fine viewing. The vertically oriented *zoom slider* will be displayed underneath the last icon of the toolbar and extends to almost the end of the toolbar. The slider allows for adjusting the amount of zoom at any level between 0.01 and 100 based on a logarithmic scale.

When using the zoom slider, the number by which the view is zoomed can be seen in the textbox on the bottom controls where the % zoom is located. The zoom slider size is in the form of *times*, such as ×0.82 which indicates that the picture is zoomed to $\frac{82}{100}^{th}$ of the original size as seen in the `Settings` → `Format` menu. Once you have set the zoom to the desired size, use the vertical and horizontal scroll bars to position the view as needed. As mentioned earlier, this variety of zoom only affects the video and not the resizing of the Compositor window. After utilizing this slider bar for zooming around, you may want to switch back to *Auto* using the controls on the bottom of the window. This slider bar is also displayed when you click on the icons for *Adjust camera automation* or *Adjust projector automation*.

The Compositor window image in figure 2.10 shows the zoom slider bar with rectangular shaped slider about in the middle. Note that the magnifying glass is enabled which automatically pops-up the slider and the *Protect video*

from changes is disabled. Also note that there is a scroll bar on the bottom and right side of the image since the image at this magnification does not fit in the window. The Controls zoom textbox shows ×0.82 size.

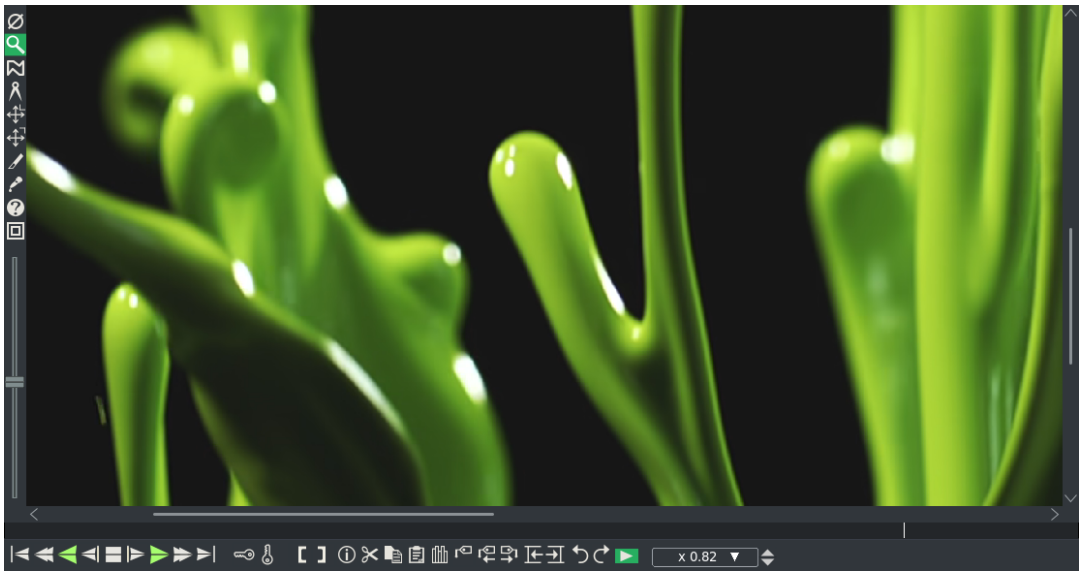


Figure 2.10: *Compositor window zoom slider bar and scroll bars*

Edit mask brings up a mask editing menu with many versatile options as described in great detail later in this section (2.2.6). You may also have to click on *Show tool info* to popup the menu depending on whether or not you dismissed that window previously.

Ruler this can be a handy tool to get the X,Y coordinates of an exact point or to measure the distance between 2 points. To use the *Ruler*, move the mouse on the video to get to the desired spot - these X,Y coordinates will be displayed in the *Current* text box. Clicking the LMB creates Point 1 and then continue to hold down the LMB so that a ruler line is created between this Point 1 and the stopping Point 2. *Deltas* is the X,Y difference between the 2 points; *Distance* is the number of pixels between the 2 points; and *Angle* is the angle in degrees of the ruler line. In Figure 2.11 you can see the Ruler menu on the right side of the Compositor window.

Holding down the Ctrl key while dragging with the LMB on one of the points, will ensure that the line is always at a multiple of a 45 degree angle. Holding down the Alt key while dragging with the LMB on any of the points, will translate the ruler line to another place on the video while maintaining its length and angle. For some desktop window managers, such as *UbuntuStudio 16.4* and *Arch*, the Alt key is already in use by the Operating System so you will have to use Alt+Ctrl instead. If you dismiss the Ruler menu, click on *Show tool info* to get the menu to popup again.

Adjust camera automation the camera brings up the camera editing tool. Enable *Show tool info* if the popup menu does not appear. More detail for usage is provided in the subsequent section 2.2.5.

Adjust projector automation the projector brings up the projector editing tool. Enable *Show tool info* to get the menu to popup again. More detail for usage is provided in the subsequent section 2.2.5.

Crop a layer or output this is a cropping tool used to reduce the visible picture area. More detail for usage is provided in a subsequent paragraph (2.2.7). There is also a Crop & Position plugin that provides a different set of capabilities described in 10.10.21. Enable *Show tool info* to get the menu to popup if it does not come up automatically.

Get color / eyedropper brings up the eyedropper used to detect the color at a particular spot. Enable the *Show tool info* if the Color popup menu does not come up automatically or if that menu was accidentally dismissed. Click on a specific color in the video output with the LMB to see the selected color. You can then use that color's value to be applied to some effects depending on how the effect handles the eyedropper. If you set the *Radius* to a larger number, a square appears around the cross cursor and an average value of the color will be the result. If you *Use maximum*, instead of an average value of the color, the result will be the maximum of all values within that square.

Show tool info this tool button is used in conjunction with the other tools on the compositor's toolbar. You only need to click on this if one of these tools popup menu does not come up or has been dismissed - Mask, Ruler, Camera, Projector, Crop, or Eyedropper tools. You can also use it when highlighted to dismiss the highlighted tool's dialog box. It is not needed for *Protect video from changes*, *Zoom view*, and *Show safe regions* since they have no dialog popup menus.

Show safe regions draws 2 outlines to display the safe regions in the video as you can see in Figure 2.11. On some particular TVs/monitors/displays, the borders of the image are cut off and that cut off section might not be as square as it appears in the compositor window. These are especially useful if the device for the output display is an older model TV. The outside largest outline is the *action safe overlay*; whereas the inside smallest outline is the *title safe overlay*.

Using the *Show safe regions* has no affect on the rendered output. The purpose of showing the borders is to make it easy to see where it might be cut off. This area outside the safe region can then be used as a scratch or vertical blanking space. Enabling the safe regions makes it really easy to see these borders so that you can make sure titles are inside the inner outline and actions are inside the outer outline.

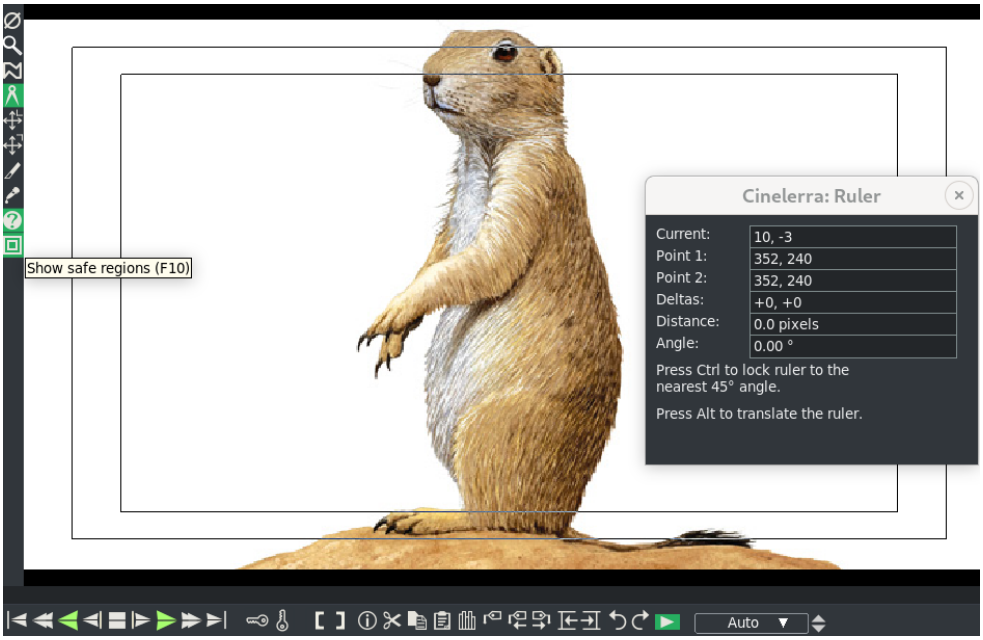


Figure 2.11: *Note the black outlines showing the safe regions. Also note the Ruler menu*

2.2.3 Compositing

Much of the editing in CINELERRA-GG involves *compositing* which is the combining of visual elements from different sources into single images. This includes such things as speeding up and slowing down the video, changing the resolution, creating a split screen, and fading in and out. Compositing operations are done on the timeline and in the Compositor window (canvas) using various operations and other compositing attributes that are available in the Resources window. When CINELERRA-GG is performing a compositing operation it plays back through the compositing engine, but when not, it uses the fastest decoder that it has.

2.2.4 The Temporary, Track and Output Sizes

This section explains a few things which help to understand Compositing - especially with relation to the camera, effects, and the projector.

The Temporary

CINELERRA-GG's compositing routines use a *temporary* which is a single frame of video in memory where graphics processing takes place. By default the size of the temporary is that of the project ($W \times H$); i.e., the output size; i.e, the canvas size. But if the tracks on the timeline have various sizes different from those of the project, then the temporary will take the size of the active track (viewport = green border). We can also change the size of the viewport using the Camera tool, as well as pans and zooms the temporary displayed in it. Effects are processed

in the temporary and as such are affected by the temporary size. In the case of the camera, its viewport is the temporary size. The size of the temporary and of the output in the compositing pipeline can be different and vary for any particular frame. However, projectors are rendered to the output and so are affected by the output size. The temporary is the basis on which the Projector tool acts to display the canvas on the Compositor window. The canvas always has the size of the project, although with the projector we can make pans and resizes of the red border inside it. When the temporary is smaller than the output, the temporary will have blank borders around the region in the output. When the temporary is larger than the output, it will be cropped.

Track and Output size

The *Track size* is used to define the temporary size with each track having a different size (viewports). It also serves to conform the input media to a chosen format (aspect ratio). So each track can have a different format (viewport). You can see or set the track size by RMB click on a track and then select *Resize Track* to resize the track to any size. Or select *Match output size* to make the track the same size as the output. Or from the Resources window, RMB on a video media and choose *Info* and then *Resize*. When a track is resized then what it looks like on the compositor changes. The relationship between the track and the project's output size makes it possible to magnify or reduce the size of a track in regards to the final output. This feature means you can create visual effects such as different aspect ratios, adjust split screens, zooms, and pans in the compositor.

The *Output size* can be set in `File` → `New` when creating a new project, or by using `Settings` → `Format`, or in the Resources window with RMB click on a video asset and choosing `Match` → `Match project size`. When you *Match project size*, you are conforming the output to the asset. To change the size and aspect ratio of the output (Projector) we have to change the whole project, which will alter all the tracks in the timeline. Once you have set the output size in 1 of these 3 ways, any newly created tracks will conform to the specified output size. When rendering, the project's output size is the final video track size where the temporary pipeline is rendered into.

To clarify, let's take an example.

If we load a media (M) into Resources and adjust the size of the project to the size of M, we will get the canvas size of the media $W_M \times H_M$). We can see in the Set Format window that the project's default values ($W \times H$) have changed to those of M ($W_M \times H_M$). The tracks on the timeline can have different sizes, but what we will see on the canvas, that is, in the Compositor window, is always the output size i.e. the size of the project. Tracks with smaller sizes will be seen with black bands; tracks with larger sizes will be cropped. Each track has its own size but

we will see it inserted in the output size. If we change the output size the tracks will not change, remaining in their original size. We can only change the size of the tracks by manually acting on each one in the ways seen before. Finally, if we create a new track in a project of size $W \times H$, it will assume the size of the project automatically.

Aspect Ratio (Theory)

The aspect ratio is the ratio of the sides of the frame (*Width* and *Height*). For example, classically broadcast TV was 4:3 (= 1.33), whereas today it has changed to 16:9 (= 1.78); in cinema we use the 35 mm aspect ratio of 1.375 (Academy aperture), academy flat (1.85 or widescreen) but even more so the super 35 mm (from 1.33 to 2.39). There are also anamorphic formats, i.e. that have no square pixels, like Cinemascope (2.39). The projection must be normalized to have an undistorted view.

From the film or digital sensors of the cameras, we can extract any frame size we want. We are talking about *viewports*, which we will examine shortly. Also important is the output of the video that will be rendered, because it is what we will see at the cinema, or on TV, or on the monitor of the PC, tablet or smartphone. Referring to figure 2.12, you can see these two possibilities: with the *Camera* tool you choose the size and aspect ratio of the source file (regardless of the original size); while with the *Projector* tool you choose the size and aspect ratio of the output. Other ways of changing the aspect ratio of assets or tracks we have seen previously (`Resize track`; `Match Output Size`; `Resize asset`). A method of changing the size of the entire project (canvas) is via the `Set Format` window. The following formula is used to vary the aspect ratio:

$$\frac{W}{H} = \text{frame aspect ratio } (\frac{pixels}{pixels})$$

For example to obtain an aspect ratio of Super 35 mm (2.35) starting from a FullHD file (1920x1080) whose base extension (1920) we want to keep:

$$\frac{1920}{H} = 2.35$$

from which: $H = 816$ pixels

At the same time as changing the *Height* parameter we also need to set `Display Aspect ratio` to 2.35. In fact, the parameters in Canvas Size are not related to those in Display Aspect ratio, unless we keep the `Auto` option checked, and we need to set both before we click on the `Apply` button. To set the aspect ratio to 2.35:1 we can choose from the drop-down menu the value 2.35 or set the value directly in the two input fields. Or again, it can be done automatically via the Auto option. Finally we can click on the Apply button to complete the calculations. Now

we have arrived at the desired result: typical Super 35 mm dimensions and aspect ratio, although starting from a 16:9 FullHD. The new canvas, however, lost the pixels of a part of the initial video (crop), to be precise $1080 - 816 = 264$ lines of pixels from top and bottom.

CINELERRA-GG allows you to vary the input and output aspect ratio in the ways indicated in the previous section: by varying the pixels of the sides (Width/Height) or by setting a multiplication coefficient (W/H Ratio; in this example: placing $HRatio = 816 : 1080 = 0.7556$) which performs the calculation automatically. If you set *W Ratio* and *H Ratio* at the same time with the same values, they work as multipliers and you get a resizing of the canvas, without altering the initial aspect ratio. If you change them to two different values or change only one of the two parameters, leaving the other at 1, you get an anamorphic video, with the pixels no longer being square (1:1) but becoming rectangular, deforming the image. If you use a non-standard aspect ratio, this has impact on other areas like titles, included stills and included non-anamorphic footage. To avoid anamorphosis, the `Display Aspect ratio` must also be adjusted at the same time, for example, with the Auto option. *Anamorphic* format is a complex field that is discussed in Andrea's paper: `https://cinelerra-gg.org/download/Anamorphic.pdf`. as well as in the Raffaella Traniello's guide: `http://www.g-raffa.eu/Cinelerra/HOWTO/anamorphic.html`.

2.2.5 Camera and Projector

In the compositor window, *Adjust camera automation* and *Adjust projector automation* are editing tools to control operation of the camera and projector. In CINELERRA-GG's compositing pipeline, the camera determines where in the source the *temporary* is copied from while the projector determines where in the output the *temporary* is copied to (figure 2.12).

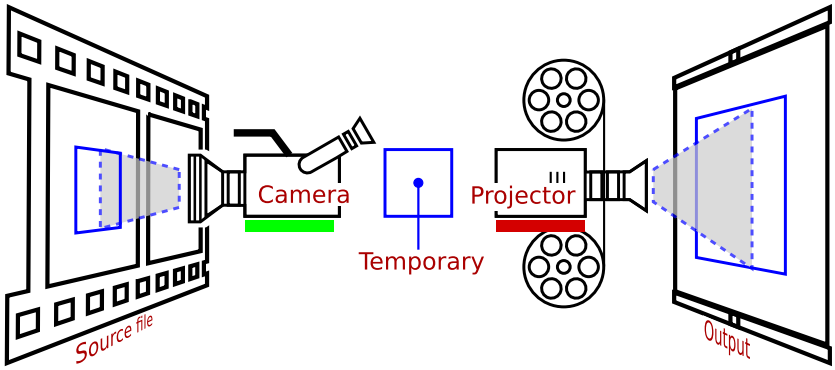


Figure 2.12: *Compositing pipeline*

In compositing, each frame can be changed using various options and plugins,

such as a color correction plugin (figure 2.13). After the image has been modified, the final image is projected to the compositor so that you now have a changed original.

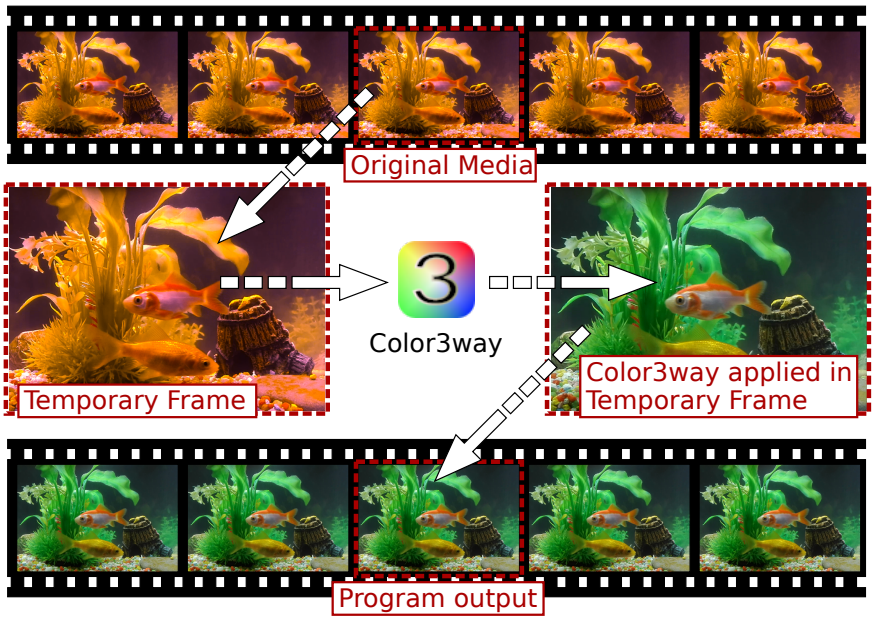


Figure 2.13: *Color3way on Temporary*

When editing the camera and projector in the compositing window, the affected track will be the first track that is armed. If there are multiple video tracks, you can select one specific track for editing with a LMB click on the *Arm track* icon of the desired track. This is called "solo" the track. To reverse this solo-ing, LMB click on the icon again.

Projector Compositing

The purpose of the *projector* is to composite several sources from various tracks into one output track. The projector alignment frame is the same as the camera's viewport, except that it shows where to put the contents of each temporary on the output canvas. To get into projector editing mode, click on the *Adjust projector automation* icon in the Compositor toolbar. You will then see red border lines surrounding the image and 2 diagonal lines criss-crossing in the middle, displayed in the video window. The red outline indicates the size of the frame that will be sent to the Output. You can easily drag the box with LMB, moving the frame in x and y directions. When moving along the $z-axis$ (i.e. the zoom, with SHIFT+Drag) the box exactly follows the movement and the size of the frame. After you position the video with the projector, you may next want to *Adjust camera automation*.

The Viewport

The *viewport* is a window on the camera that frames the area of source video. The size of the current track is used for the initial size of the viewport. A smaller viewport, for example (640×480), captures a smaller area; whereas a larger viewport of (800 × 600) captures a larger area. If the captured area is larger than the source video, the empty spaces will be automatically filled with blanks. To change the size and aspect ratio of the viewport (Camera) of a single track, right-click on the track in the timeline and choose Resize Track. Here we can vary the height and base of the viewport in pixels or choose the multiplication coefficient for each side (Scale). With OK we will see the change in the Compositor window with the new dimensions reflected in the green box. We can have different size viewports for each video track on the timeline. To go back, reset the viewport to the original value. After the viewport is defined, the camera needs to be placed right above the area of interest in the source video. Operations to control the location of the camera are as follows:

1. In the compositor window you should see the selected track.
2. LMB click on the *Adjust camera automation* to bring up the editing menu and the green and yellow colored outlines.
3. Use the LMB to drag the video over the display in the compositor window to the desired placement.

When you drag over the viewport in the compositor window, it looks like you are moving the camera with the mouse. The viewport moves in the same manner.

Camera Compositing

Select the camera button to enable camera editing mode. In this mode, the guide box shows where the camera position is in relation to past and future camera positions but not where it is in relation to the source video. The green box is the Viewport; at the beginning it coincides with the size of the source frame. If we move the viewport by dragging it with LMB (moving it in x/y), the green box remains fixed to the original size but the frame is moved to the new position. A yellow frame will appear along the edges of the frame to indicate the displacement with respect to the green box; this behavior differs from that seen for the Projector. Even if we click on the $z-axis$ (SHIFT + Drag, equivalent to the zoom), the frame narrows or widens, moving behind the yellow frame.

Camera and Projector Menu

The camera and projector have shortcut operations that do not appear in the popup menu and are not represented in video overlays. These are accessed in the *Show tool info* window. Most operations in the Compositor window have a tool window which is enabled by activating the question mark icon (figure 2.14).

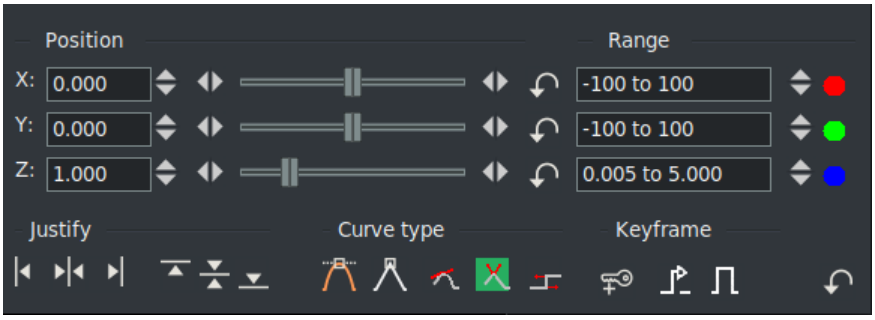


Figure 2.14: *Camera and Projector tool*

In the *Position* section you can change the X, Y and Z coordinates. By either tumbling or entering text directly or by using the slider, the camera and projector can be precisely positioned. There is also a reference to the color of the curve as we see it on the timeline. You can also define the *range* of action which by default is [-100 to 100]. By pressing the Reset button for each coordinate, or the global Reset button, the range is automatically brought to the project size value (HD; 4k; etc), which are usually the most useful limits. Note that the range can also be changed in the *Program* window, in the *zoom bar*, where there are similar input fields to enter the chosen limits.

In the *Justify* section we can use automatic positioning in the 6 standard coordinates: Left, Horizontal, Right, Top, Center and Bottom.

The translation effect allows simultaneous aspect ratio conversion and reduction but is easier to use if the reduced video is put in the upper left of the *temporary* instead of in the center. The track size is set to the original size of the video and the camera is centered. The output size is set to the reduced size of the video. Without any effects, this produces just the cropped center portion of the video in the output.

The translation effect is dropped onto the video track. The input dimensions of the translation effect are set to the original size and the output dimensions are set to the reduced size. To put the reduced video in the center subsection that the projector shows would require offsetting out X and out Y by a complicated calculation. Instead, we leave out X and out Y at 0 and use the projector's tool window. By selecting left justify and top justify, the projector displays the reduced image from the top left corner of the *temporary* in the center of the output.

In the *Curve type* section we can choose between various interpolation algorithms that determine the curve type :

smooth: bezier interpolation which are flat at the endpts

linear: piecewise linear curve.

tangent: bezier interpolation with collinear endpts in a specified line.

free: piecewise bezier, if there is such a thing.

bump: has 2 values, one viewed from the left/right, discontinuous.

In the *Keyframe* section we can create new keyframes and set them as Bump autos (*Right/Left edges* and *Span* buttons). For further details see Bump autos.

Reset to Default

In the compositing window, there is a popup menu of options for the camera and projector. Right click over the video portion of the compositing window to bring up the menu:

Reset Camera: causes the camera to return to the center position.

Reset Projector: causes the projector to return to the center.

Use Case: Interaction Between Camera And Projector [2]

1. Start by shrinking the projector to $z = 0.500$ ($\frac{1}{4}$ of the original frame).
2. The next step is to switch to the camera and note that the green box has assumed the size of the projector, i.e. the red box. The value of z of the camera is always equal to 1.000 (default) but the frame is $\frac{1}{4}$ of the original frame, i.e. it has the size of the projector that has $z = 0.500$. This is the current viewport size.
3. You enlarge the room bringing $z = 2.000$. You can see that the dimensions of the viewport (green box) do not change, remaining the same as those of the projector. However, the frame has been enlarged and this variation is indicated by the enlargement of the yellow box. Let's remember that this follows the changes made with the camera tool.

[2] Example provided by Sam. The relative video is located at: https://youtu.be/ZQaLZiYY1lg

4. We can drag the room so that we can center the frame to our liking. The movement of the yellow box shows well the variation compared to the green box.

5. Finally, if we want, we can switch to the projector tool to move the output frame to the position we want with respect to the size of the source. Of course, we can also work on the z, which in the example is at $z = 0.500$, if we have decided to change the size of the output.

2.2.6 Masks

Masks can be used to accomplish various tasks but basically are used to select an area of the video to be displayed or hidden. They can be used in conjunction with another effect to isolate the effect to a certain region. Another usage is where you slightly delay one video track copy and unmask an area where the one copy has interference but the other copy does not. Or use a mask when color correction is needed in one part of a frame but not another. A mask can be applied to just a small section of a color corrected track while a plain track shows through. Removal of boom microphones, license plates, people and airplanes via mask is a very common usage.

The order of the compositing pipeline affects how masks are done. Usually masks are operated on the temporary, after the effects but before the projector. Because of the way this works, multiple tracks can be bounced to a masked track and projected with the same mask.

The compositing pipeline graph has a masking stage (figure 2.15).

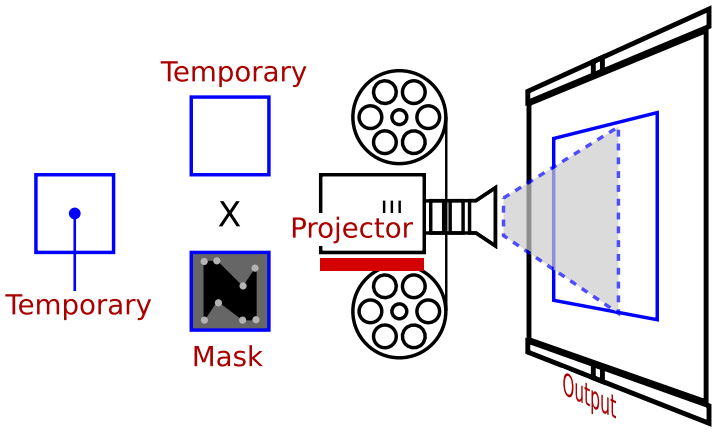


Figure 2.15: *Compositing pipeline with mask*

Compositing pipeline with masks

The Mask popup menu can be overwhelming upon first encounter. However, if you follow the next few steps you can create a single simple mask without having to understand every possible parameter. Be sure you are working in a Format that includes the Alpha channel, YUVA or RGBA, because you need to have transparency for masking to show or will end up with only a black area.

1. To define a mask, in the Compositor window click on the *Edit mask* icon to get the popup Mask menu. If the menu does not come up, click on the *Show tool info*.
2. On the video, LMB click on the place where you want to start a mask.
3. Then LMB click on another spot of the image to create each new point of the mask. Once you have at least 3 points, lines will be drawn between them, but you can just create as many more points as you need and the lines will be redrawn to cover all points. When you create each point of the mask a straight line curve is expanded, altering the shape of the mask. The mask position will always be in the same position on each image of the video unless you enable *Generate keyframes while tweaking* on the Program window Transport and Buttons bar. Then when enabled you can move a mask over time.
4. For a mask to be seen or not seen, there must be another video track under the track that you are viewing in the compositor. An easy way to see the masked area is to just add an empty track below the target track and drag the *Gradient* plugin to a highlighted area on that track.
5. You can move existing points to new locations by simply using the LMB at a point to drag that point to a different location.
6. The mask can be translated as a single entity by Alt-dragging the mask. For some desktop window managers, such as *UbuntuStudio 16.4* and *Arch*, the Alt key is already in use by the Operating System so you will have to use Alt+Ctrl instead.
7. To create curved instead of straight lines between the points, use Ctrl-drag on a specific point. Using Ctrl-drag activates bezier handles (control points) to create these curves between the points. For example, on a mask with just two points, you can create a romantic heart mask.

There are a lot more operations you can do using the Mask menu as shown in figure 2.16. Detailed description is provided here next. Note that the Mask window is separated into various sections to make it easier to locate the area of interest.

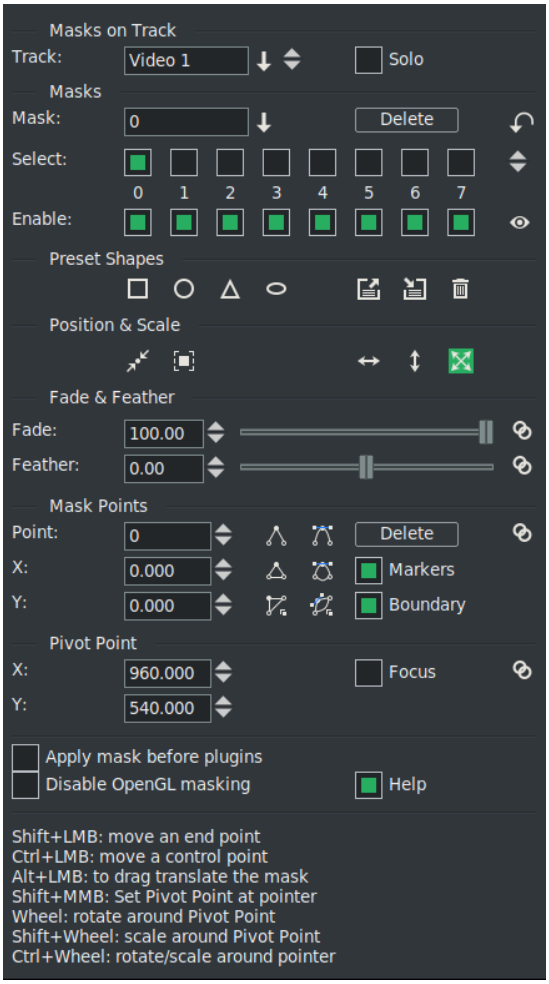


Figure 2.16: *Mask options window*

Masks on Track section

The *Track*: textbox displays the different video tracks for your session which will be initially set to the first armed video track or will be left blank if there are no armed tracks. A pulldown to the right of the box brings up the names of all of the video tracks allowing you to change to which track the masking applies. You can also just use the tumbler to easily mouse up/down to get to the desired track. In the pulldown list, any track that has a red colored text name is disarmed so that you can not change it. A track that contains masks has yellow colored text for easy identification. Only when there are no masks on the track, do you have the default text color. This textbox is display only and you can not type into it.

The *Solo* button in the Masks on Track section of the Mask window is very handy when working with masks on different tracks. It displays just that track so that you see only the track you choose, as well as the tracks behind it to show the mask

part. The Solo button is just a convenience to prevent having to mouse over to the patchbay.

Masks section

The *Mask*: textbox will show you the mask numbers of 0—7 or the 8 ascii character name that you have used to designate each mask number. There is a pulldown on the right side to easily switch to another mask.

The *Delete* button is used to delete the mask number/name that is selected. The symbol to the right with tooltip of *Delete all masks* can be used to delete all of the current video track masks.

The *Select*: row of checkboxes is used to indicate which mask is currently displayed for that video track in the Compositor. Numbers that are colored yellow are active masks for that track. A tumbler to the right allows for quickly changing the mask number displayed.

The *Enable* row of masks makes it so you can enable all or none of the masks, making it possible to look at no masks or at one mask without interference from the other masks. The symbol that looks like an `eye` can be used to easily check all or none as the tooltip *Show/Hide mask states*.

Preset Shapes section

There are 4 shapes that are automatically available for usage as masks – square, circle, triangle, and oval. In addition, the next 3 symbols in this section are for the purpose of loading, saving, and deleting your own customized shapes. The first symbol, *Load* preset, will bring up a list of your previously saved presets. Clicking on *Save* preset brings up a popup window allowing you to provide a name used to identify the preset you want to save, along with a pulldown to see the names of your other saved presets. Clicking on *Delete* preset also brings up a textbox with a pulldown to choose which one to delete. There is a file, called `mask_rc`, in `$HOME/.bcast5` that records your custom masks.

When you click *Load* preset, keep in mind that it will write the mask number that you have selected so if you already have a mask at that location, it will write over it – just *Undo mask* under the main window *Edit* pulldown (shortcut 'z'or 'Ctrl-z') to revert to the previous if you made this mistake.

Position & Scale section

Center mask button allows for quickly centering a mask on the video track. *Normalize* mask button makes it easy to normalize the size of the mask based on the

scale of the video - about 1/4 the window size in the X and Y directions. Normalization makes the mask fit in the space to avoid having it wander off of the window, then you can move it around easily before resizing. The next 3 symbols concern the direction to *drag translate* a mask using the `Alt+LMB` thus making it easy to preserve the current *X* or *Y* value when desirable. For some desktop window managers, such as *UbuntuStudio 16.4* and *Arch*, the Alt key is already in use by the Operating System so you will have to use Alt+Ctrl instead.

`xlate/scale x` - drag translate constrained in the *X* direction

`xlate/scale y` - drag translate constrained in the *Y* direction

`xlate/scale x/y` - drag translate in both directions; this is the default and after using the other 2 options, you should reset to this to avoid future confusion while dragging.

Fade & Feather section

The `Fade:` textbox is used to type in a fade value; the tumbler to the right of the textbox allows you to increase or decrease that number; and the slider bar makes it quick to adjust the fade value. The fader goes from −100 on the left to +100 on the right for negative to positive. Default value is +100. The fade slider includes a sticky point at 0 so that it is easy to get to 0 without going too far or not quite far enough – that way you don't have to keep jiggling to get there.

In addition there is a *Gang fader* symbol to allow for having all of the masks fade in unison. The symbol is surrounded by a green colored background [3] when it is in effect. If you have multiple masks with different modes, a decision had to be made on what value to use – it uses the maximum transparency value of the background to determine the operations results. To understand how this works, here is a summary:

Note1: The area outside the mask is referred to as the background.

Note2: The operational result is based on the maximum transparency value of that background.

Case 1, Positive Fade: When the fade for all of the masks is positive, affecting the area inside of the mask, all of the background colors are at a transparency value of zero. So the largest transparency value is 0,and all masks are drawn with opaque backgrounds, depicted as one would expect.

[3] green is used in the default Cakewalk theme, but the background color will be different in other themes

Case 2, Negative Fade: When the program computes the background color for any number of masks that includes negative mask(s), it uses the largest transparency number as the determining factor for the background. Only 1 of the masks can be largest, and wins for the background transparency result.

Feather: works in a similar manner to a *gradient Fade* aligned on the mask boundary but is a logical function instead of a mathematical function so will be faster. The *Gang feather* symbol also works in a similar fashion and is surrounded by a colored background when it is in effect.

Mask Points section

This section is used to change to a different mask number and manipulate the masks you have created.

The *Point*: textbox provides the ability to change which point number for the current mask that you want to work on. It has a tumbler to allow for quickly switching the point number. The *X:* and *Y:* boxes below reflect the current values and allow for modifying the X/Y coordinates and these too have tumblers. The *Delete* button will allow for deleting the selected point number.

The next 6 symbols in 2 columns represent *Smooth* and *Linear* buttons. Smooth buttons use an algorithm based on the previous point and the next point to create a curved line. The smoothing operation takes three points, A, B, C, and arranges the slope at B to be AC as it moves to the next point for that mask.

smooth point → smooth a single point.

smooth curve → smooth all points on a mask edge curve.

smooth all → smooth all enabled masks on this track.

Linear buttons of *linear point*, *linear curve*, and *linear all*, perform the inverse of the smooth functions. The control point vectors on the bezier endpoints are set to zero magnitude.

In addition there is a *Markers* and a *Boundary* checkbox which come in handy to turn off the display of the points and the outline of the mask. Turning off *Markers* is very useful when you have a lot of control points that clutter the display and make it more difficult to see the actual mask. A helpful feature is available by disabling *Markers* and enabling *Boundary* which results in all masks being displayed in the viewer even if they are not enabled; for example you can then see mask 0, mask 1 . . . at the same time.

A *gang* symbol on the right hand side of this section, tooltip of *Gang points*, is another useful feature that makes it easy to drag a mask to an exact coordinate using the *X* or *Y* textbox for numerical input or the associated tumblers. This works like the `Alt+LMB drag` translate but provides the ability to be precise.

Pivot Point section

The *X:* and *Y:* coordinates mark the value of the current *Pivot Point* used for rotation, scaling, and translation. You can either directly key in numerical values or use the tumblers to change the values as long as the *Focus* checkbox is checked.

The *Focus* checkbox is used in case you want to set a different point in the Compositor for pivoting instead. And the *Gang* symbol for rotate/scale/translate means that these operations will be performed on all points of the enabled masks. The gang symbol is surrounded by a colored background when it is in effect. When performing a rotate operation on a mask with the mouse wheel, *acceleration* is in effect – this means the faster you wheel, the more space is covered so that you do not have to wheel dozens of time to make a full rotation. Then when you wheel around slower, you can fine tune the result. Note that in order to be able to rotate/scale around pointer, the Focus checkbox must be unchecked.

Other sections

Finally there are the *Apply masks before plugins* and *Disable OpenGL masking* self-explanatory checkboxes.

Note: Not all OpenGL software can support the current masking methods. If your opengl implementation does not support Shader Version 4.3 or has trouble with this (it is relatively new to opengl at the time this was implemented), then this checkbox will allow you to use the software masking to avoid any potential issues. Normally, OpenGL is probed for the shader version and will automatically use the software implementation if required.

The *Help* checkbox can be enabled in order to see a list of the keys used to perform various operations. If you use Masking infrequently, these are a valuable reminder to which key combinations to use. Currently they are as follows:

Shift+LMB	move an end point
Ctrl+LMB	move a control point
Alt+LMB	to drag translate the mask
Shift+MMB	set Pivot Point at pointer
Wheel	rotate around Pivot Point
Shift+Wheel	scale around Pivot Point
Ctrl+Wheel	rotate/scale around pointer

Key Alternatives

Note: For some desktop window managers, certain keys may already be in use by the operating system, so you will either have to redefine them in your desktop or use different key combinations. For example, at least some desktops used with *UbuntuStudio 16.04* and *Arch* field the `Alt` key, thus requiring alternative key combinations to be needed. Below are some of these alternatives.

LMB	move/create an end point (to move the end point the pointer must be above the point)
Shift+LMB	move an end point (the pointer may be near the point, not above it)
Ctrl+LMB	move/create a control point
Alt+Ctrl+LMB	to drag translate the mask
Shift+Key Delete	to delete the mask
Shift+MMB	Set Pivot Point at pointer
Alt+Wheel	zoom in/out the screen (also available in Ubuntu16 but does not exist in all distros)

Focus checkbox = unchecked:

Wheel	rotate around Pivot Point
Shift+Wheel	scale around Pivot Point
Ctrl+Wheel	rotate around pointer
Ctrl+Shift+Wheel	scale around pointer

Focus checkbox = checked:

Wheel	rotate around Pivot Point (“Custom focus point”)
Shift+Wheel	scale around Pivot Point (“Custom focus point”)

2.2.7 Cropping

Cropping is used to reduce the visible picture area by changing the output dimensions, width and height in pixels, and the X, Y values. An example of cropping and the crop menu is seen in figure 2.17. The easiest way to use cropping is to click with the LMB at the spot to begin cropping and while holding down the LMB, drag the mouse. This creates a rectangular cropping area. To change the size/location of that area, click on any of the 4 corner points with the LMB and drag. While dragging, you will see the X1, Y1 coordinates and W for width, H for height, in the Crop tool popup menu automatically change numerical value to reflect the current position. For precise locations, you can keyin exact values into those textboxes instead of using the mouse. Once you have the crop area defined as you want it, then click on the *Apply* button to have the actual cropping take affect.

There are 3 choices of crop methods to choose in the menu pulldown on the bottom right side.

1. Reformat - Reformat Session crops and changes the Format for the entire session. Because the Format is changed, this is applied to all tracks in the project. The part of the image outside the rectangle will be cut off and the projector will make the video fit. The `Settings → Format` window will show the new project Width and Height values and the projector tool window will show the new X, Y values. Track size remains unchanged. You can undo the cropping by entering the original project dimensions in the `Settings → Format` window for the Width and Height. You will also have to use the Projector tool in the Compositor toolbar to *Ajdust projector automation* by clicking on the Reset icon.

2. Resize - Resize Projector; to undo this, enable *Adjust projector automation* and do a Reset.

3. Shrink - Resize Projector and Camera; to undo this, enable each of the *Adjust projector and camera automation* tools, one at a time, and do a Reset in the menus.

An important note here is that the original aspect ratio will be maintained so if your frame is rectangular (as many are) and you "crop" by surrounding the region of interest with a square, the cropped area will be more than you marked in order to keep the aspect rectangular shape. The Resize and Shrink options are applicable to all video tracks except the disarmed ones. This is in contrast to the Reformat option, as mentioned previously, which applies to all tracks even if disarmed because it changes the Format for the session. One last note of interest, this cropping is keyframable.

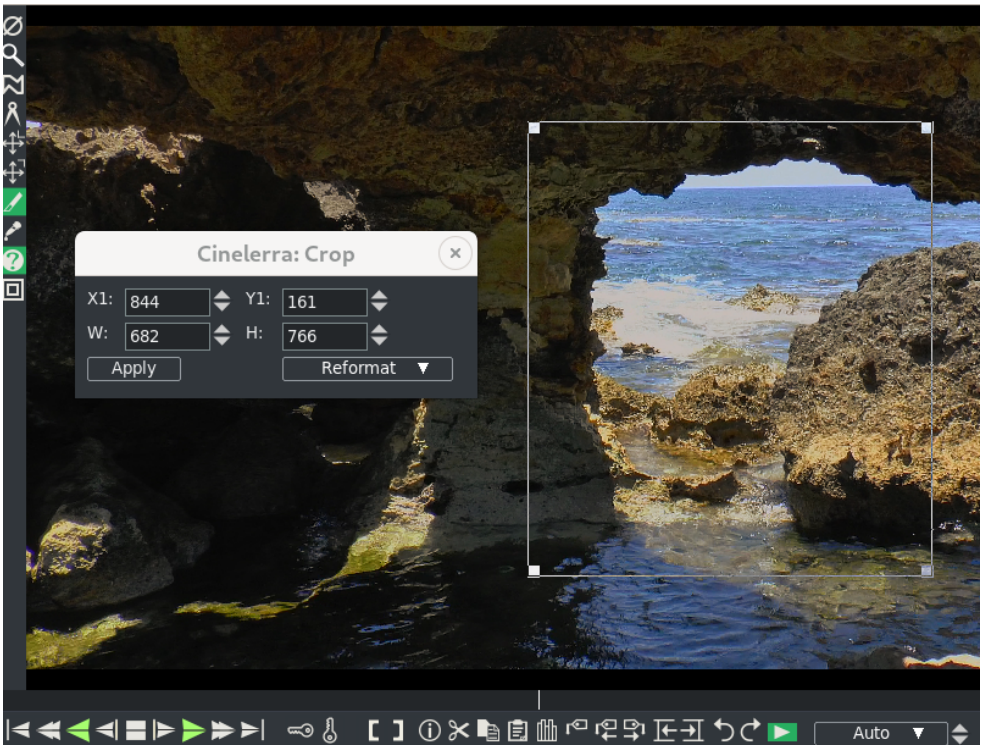


Figure 2.17: *Crop menu and outlined crop rectangle on the right side*

An easy to follow step by step usage of the cropping tool is outlined next.

- Enable the crop tool in the compositor window to display the Crop popup menu.
- Click-drag in the video to define the crop area which draws a rectangle on the video.
- Click-drag in the video to start a different rectangle instead.
- Click-drag on a corner of the rectangle to reposition that corner.
- Alt-click in crop rectangle to translate the rectangle to a different position without resizing.
- The crop popup menu allows text entry of the top left coordinates $(X1, Y1)$ and width and height (W, H) that define the crop rectangle.
- Choose one of the 3 options of Reformat, Resize, or Shrink.
- When you have the rectangle where you want it, click on the *Apply* button in the menu to actually perform the crop operation.

2.3 Viewer Window

The Viewer window (figure 2.18) is convenient for previewing your media and clips. It can also be used for editing with cuts and then paste operations into the timeline or to create a clip. There are transport buttons to use in the same manner as in the Program and Compositor windows or you can quickly move through the media by dragging with the LMB in the timebar above the transport buttons.

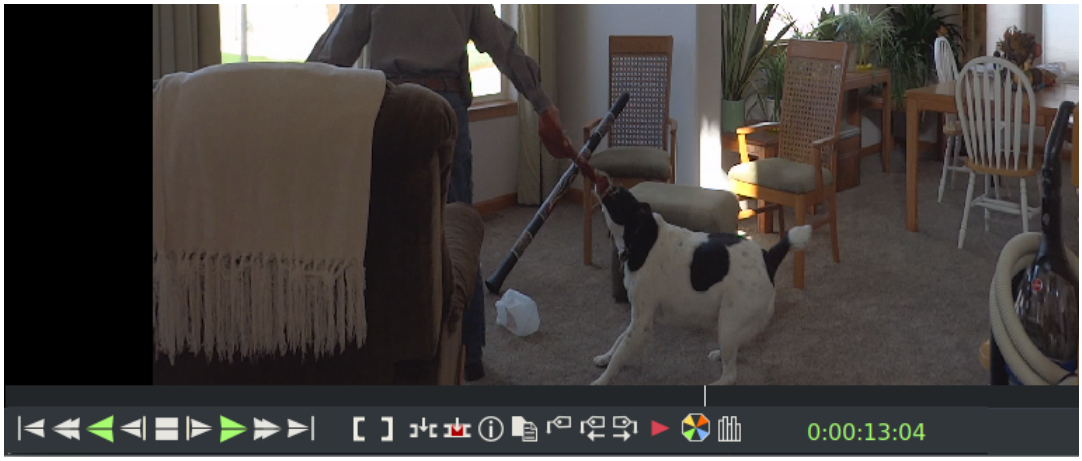


Figure 2.18: *Viewer Window - the red arrow "Play" button is left of the Videoscope button*

In order to view media in the window, you have to load it as follows:

1. In the Resources window, highlight the Media folder or the Clip folder.
2. Drag a file from the folder to the Viewer.
3. **Or** double LMB click on a media asset.
4. **Or** highlight an asset, RMB to bring up choices, and click on *View* or *View in new window*.

Note that you can have multiple Viewer windows open with different or even the same media asset. After the media is loaded you can use the transport buttons to play, rewind, stop, and so on, or for fast previewing drag with the LMB anywhere on the timebar slider. There is also the Videoscope button which is to used to enable the scopes window without having to apply the filter to the tracks/edits. A few more options available in the Viewer window can be accessed with a RMB click on the display . These functions are listed next.

1. Switch to a fullscreen display by choosing *Fullscreen*. To switch back, click with the RMB on the display again and choose *Windowed*.

2. Change the display size by choosing the *Zoom* function to select a zoom level of 25%, 33%, . . . 300%, or 400% of the original media size.

3. To remove the current media from being displayed, choose *Close source*.

The Viewer uses the project's output size format settings that is in effect to display the media instead of the original asset's format. Operations performed in the Viewer affect a temporary EDL or a clip rather than the timeline. By default, the Viewer window is automatically available but if it gets accidentally closed you can open it again by using the pulldown `Window` → `Show Viewer` to bring it back up. More details for editing in the Viewer window with the Two Screen Editing method is explained in 5.3.

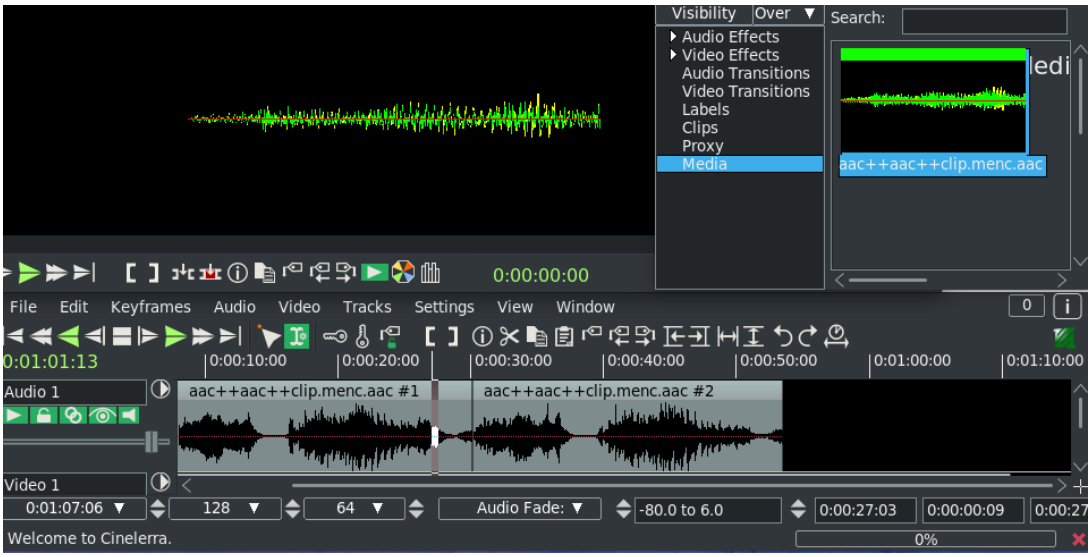


Figure 2.19: *Viewer window at the top displaying same 5 seconds as seen in the Resources window thumbnail. At the bottom of the screen is the audio loaded on the timeline.*

You can also use the Viewer to listen to media that consists only of Audio. This is a quick way to listen to the audio to see if it is what you would like to add to a timeline audio track for your project. To do this, you simply drag the audio file from the Resources window in the same manner as a video file. The viewer was designed to "view" images rather than play audio so in order to make it obvious that audio media is loaded to the viewer, a waveform is displayed that is the same waveform as shown in the Resources window thumbnail when in the *Display Icons* mode. This waveform only represents the first 5 seconds of the media and will not change or move while playing in the Viewer window. But you can play the entire piece of media in the window and as you do so, you see the play cursor line move along and the timestamp reflect the actual position. You can also create clips. The entire waveform can only be seen on an actual audio track on the timeline. An example of what this looks like is shown in figure 2.19.

2.4 Options in both the Compositor and Viewer Windows

The next sections describe capabilities that are available in both the Compositor and Viewer windows.

2.4.1 Click to Play in Viewer and Compositor

In both the Viewer and Compositor windows, there is an arrow on the right hand side of the other buttons in the edit panel as shown in figure 2.18. The "play" button can be toggled on/off via this arrow, which has the tooltip of *Click to play* . When enabled the arrow is white surrounded by green and when disabled the arrow is red.[4] The purpose of enabling this capability is to make it really easy to play the media in the window by just using the left mouse button to start or stop the play. The entire main canvas surface becomes a big play button! Although the default is initially off, a good reason to enable this, at least temporarily, is so that you can quickly review your video before a render.

left click forward play or stop forward play if already playing

middle wheel single frame forward or back

middle click reverse play or stop reverse play if already playing. Note that some 3 button mice do not accommodate a middle click for reverse but you can find out by testing from a terminal window with the command `xev`.

2.4.2 Timebar + Preview Region Usage in the Compositor and Viewer

The Viewer and Compositor each have a timebar control area with a red indicator line below the video output. The timebar shows the whole time covered by the resource. When a video resource is loaded in the main window and you move in the compositor, the insertion pointer in the main window will reflect those movements. But in the viewer only that specific media is shown and there is no corresponding movement on the timeline.

Figure 2.20: *The mouse cursor, above the green colored 'play forward' transport button, is on the timebar. Further to the right we see the red "indicator line".*

[4] the color and the look will be different for themes other than the default theme of Cakewalk

Both the Compositor and Viewer support labels and in/out pointer which are displayed in the timebar. And as with the movements, when you use the labels or in/out pointer in the compositor timebar, the result will also be reflected in the main window timebar. Along with that, of course, when you move to a label or in/out pointer in the compositor, the insertion point in the program window will go to that position.

The timebar in the compositor and the viewer covers the whole length of the resource loaded in there. Moving the mouse with the LMB pressed moves the position in the resources as indicated by the indicator line. A complete mouse movement between left and right edges of the window moves the indicator line by default along the whole timebar. With long resources it can be difficult to precisely locate sections in the resource. To make this easier, you can limit the effect of the same window-wide mouse movement to a smaller area of the timebar, this area is defined as the preview region. This region can also be moved along the timebar.

For instance, if you need to get a few 10 second clips from an 60 minute video, you could define a one-minute preview region, move it approximately to a place where a clip needs to be taken, and easily select with the mouse in and out points within the region, because the whole window mouse movement now covers 1 minute instead of 60 minutes.

To create and use a preview region, hold down the right mouse button inside the timebar on either end of the timebar close to the edge until you see the resize pointer. While continuously holding the right mouse button down, drag the arrow away from the end towards the middle of the timebar until you have the desired area outlined. The slider will be a dark red color while the selected preview region will remain the same initial black color. There are either a left or right resize pointer and you can click and drag in either direction to expand or shrink the region.

Figure 2.21: *A left-facing arrow on the right side of the reddish slider bar is used to drag the bar.*

Figure 2.22: *Here you can see the right-facing arrow used to drag the other end of the slider bar. The black area is the actual preview area.*

You can slide the preview zone left or right by holding the right mouse button over the preview zone where you will see a fat double headed arrow. The selected area will move left or right as you drag and still retains the same size.

Figure 2.23: *Note the double-headed fat arrow in the black preview area used to move the selection over.*

Settings:

1. If no preview region is set, increasing the length of the media on the timeline by inserting media or appending, has no effect on the non-selected preview region. That is, you will not see the reddish slider bar suddenly mysteriously appear.
2. If the preview region is set, when you replace the current project or file, the preview region is automatically disabled.
3. If the preview region is set, when you append data or change the size of the current project, the preview region may appear to either move, shrink, or grow depending on the new length of the media on the timeline.
4. To disable the preview region, you will have to drag both the right and the left blue slider bars completely to their corresponding end so that there is no longer any visible red slider.

A good method for taking advantage of the preview region is described here. On the main track canvas, scroll to the beginning of the area of interest. When you do that, you will see in the compositor the red indicator line of that location. Now in the compositor window, right mouse drag from the left side of the edge of the timebar to create the dark red slider bar line up to the red indicator. Back in the main track canvas, move to the location of the area you want to end looking and again you will see the red indicator line in the compositor. Use the right mouse drag from the right to stop at that end point. Using this method is often easier than continuous usage of the single frame move which can be tedious.

One last interesting item of note – sometimes you may wish to see just a little more that is outside the preview region and you can do so! You can actually move outside the compositor or viewer window space and view more, at least until you hit the end of the screen space.

2.5 Resources Window

Effects, transitions, labels, clips, proxies, user bins, and media assets are accessed here. Most of the resources are inserted into the project by dragging them out of

the resource window. Management of resource allocation is also performed here.

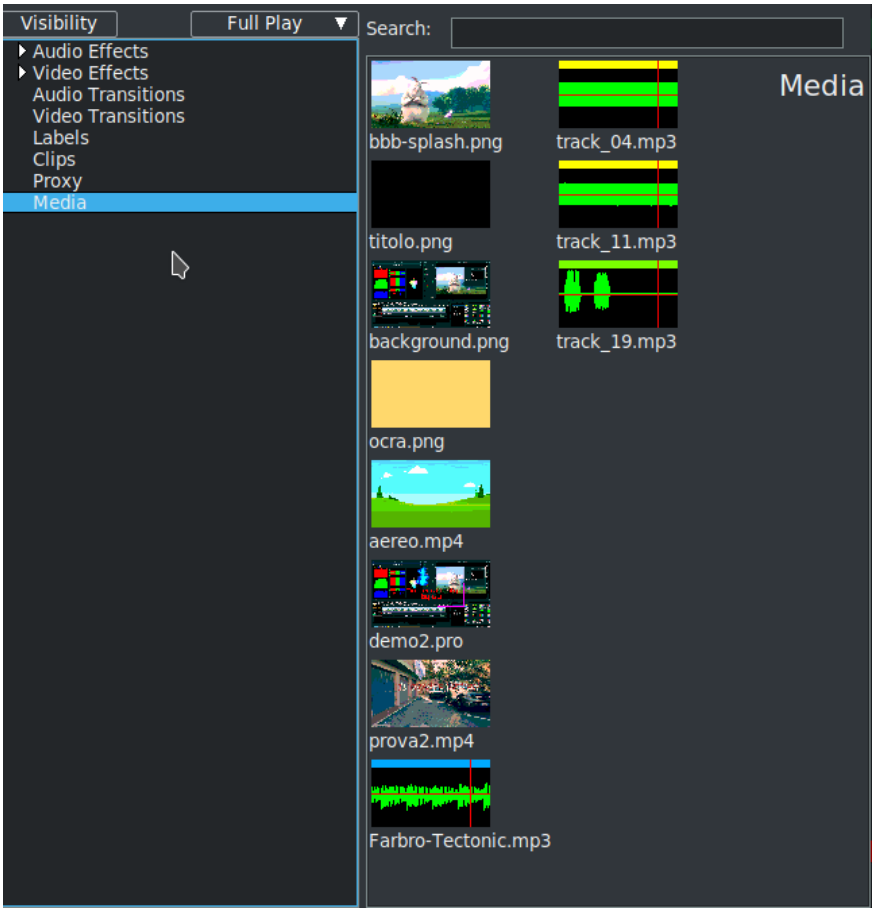


Figure 2.24: *Folders are in the first column with contents of that folder on the right hand side*

The resources window is divided into two areas (figure 2.24. One area lists folders and another area lists the folder contents . Going into the folder list and clicking on a folder updates the contents area with the contents of that folder. The folders can be displayed as icons or text. There are several variations for displaying the contents; select *Display text*, *Display icons*, *Display icons packed*, *Display icons list* as types of display for the assets or plugins. Use the letter "V" to easily scroll through the choices and see which you prefer. You can also get to these options from the menu by a right mouse click in the window.

A *Search* option is available for any of the folders in the Resources window (and when using *Attach effect* on the main track canvas for the Plugins). As you type in characters a match is made with that substring. Names that do not match are filtered out making it a lot easier to find the item you are looking for. The characters can be any where within the phrase and it does not matter if upper or lower case.

Other options you will see if you *right mouse click in the folder* which brings up the menu are described next .

Load files for convenience to load files same as from the main window so you do not have to move the mouse so far in case you have multiple monitors.

Display text/icons as described previously for format variations preference.

Select options are All, Used, Unused, and None. This gives you the capability to have a set of the contents highlighted for ease of use so you can see what is or is not loaded, or unset the highlight.

Sort items to sort the contents of the folder alphabetically. Especially helpful if you accidentally did a drag and changed your mind or dropped suddenly so that the assets no longer look nicely aligned.

Copy/Paste file list use to easily copy a set of files or paste a set of files between CINELERRA-GG and other programs or operating system windows.

Snapshot/Grabshot use to take a quick snapshot or to grab a specific area on the screen. These functions are described in detail in section 2.5.7).

Using the right mouse click to bring up a menu in the folder area , you can also switch from Display text to Display icons, Sort items and create, delete and manipulate user defined folders/bins. Select Folder to create a user Folder or modify an existing folder.

If you *right mouse click on a highlighted/selected resource*, several options are available depending on whether the resource is an effect or transition or a piece of media. You can highlight several for some options so that it is applicable to all of them, such as Info. Those listed immediately below are the available choices for media assets .

Info provided basic Asset information; details are described later in this section.

Display text/icons same as mentioned previously.

Sort same as mentioned previously.

Rebuild index if you switch from/to using ffmpeg/native for media loading, you should rebuild indexes. Or if you get hangs on media or strange looking tracks, you might want to rebuild indexes.

View use this option to bring up the media in the Viewer window.

View in new window in order to not overwrite your current viewer window, you can open any number of viewer windows to simultaneously view multiple media.

Open mixers when you record with multiple cameras setup, you can work with them most easily using the mixer mode. This is described in detail

Paste

Match if you need to change your media parameters you can choose from the following: Match frame rate, Match project size, Match all

Remove use to Remove the asset from the project or with caution, to Remove from disk permanently.

In the case of Effects or Transitions , a right mouse click on a highlighted selection leads to an *Info* button which gives a short 1 line description of what the effect/-transition can be used for. For Labels , choices are *Edit*, *Label*, and *Go to*. For Clips , *Nest* and *UnNest* as described elsewhere are available.

2.5.1 Info Asset Details

The asset *Info* window also can be used to display detailed information about the selected/highlighted media file – available for any loaded media of type mpeg or ffmpeg. This is extremely helpful in determining what type of media it is, size, resolution, format, and type/number of audio streams. It is especially useful for multiple program streams. You can have the info window popped on several of your assets simultaneously.

Detail opens an additional window with more information.

Sample rate and *Frame rate* allows you to impose a different sampling or Fps of the audio and video assets.

Another option is *Resize* button, which allows you to change the size (in pixels) of the frame of the asset.

Asset's interlacing is the type of interlacing the asset has: If the file is (H)DV type, recognition and configuration is done automatically. All other media types will be set unknown. So we have to manually set the interlacing [5].

[5] From Igor ubuntu's mail

Figure 2.25 shows the *Detail* box to click on the left side and a simple, typical output in the Asset Detail window on the right side. Also, note the highlighted media in the Resources window.

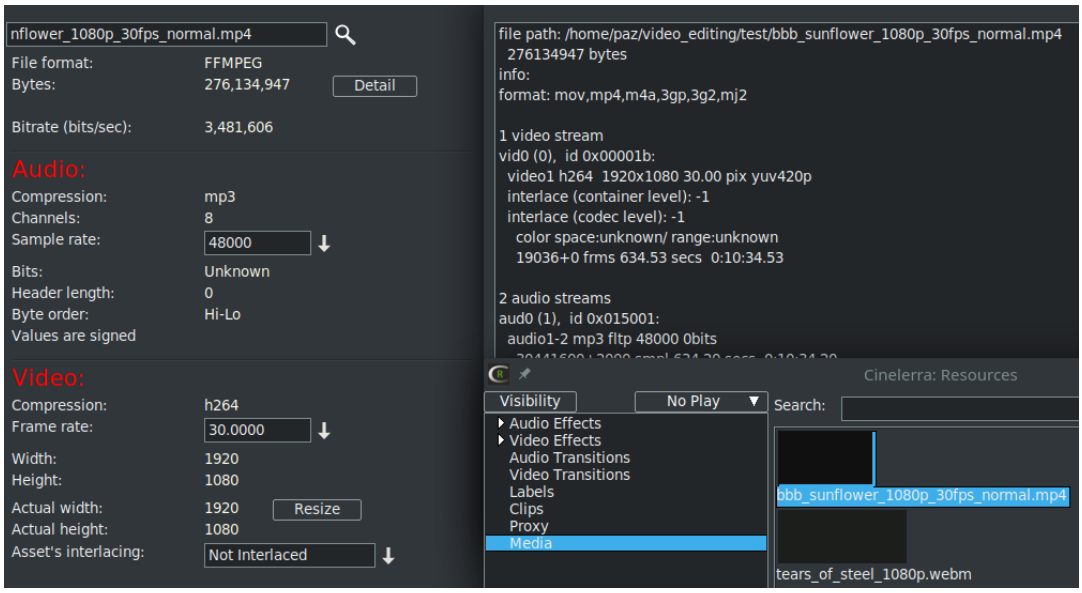


Figure 2.25: *Info windows with the 'Detail' box*

2.5.2 User Folders/Bins

Creating folders that are more specific to a particular project is helpful in better organizing your work. This can be done by utilizing the files already loaded to the *master* Media or Clips folders in the Resources window. The general rule is you can only drag clips to a ClipUserBin and you can only drag media to a MediaUserBin. No subfolders can be created within the new folders. Further below, after we populate the new folder, we will see how to use logical operators or other filters to automatically sort the media.

Below are steps illustrating an easy way to set up a new folder.

1. In the Resources window (figure 2.26), in the location of the Video/Audio effects and Media folders, bring up the *Folder...* popup by clicking the right mouse button. Highlight, then click *New Media* or *New Clips*. *Modify folder* can be used to change the name of a folder. *Delete folder* can be used to delete a folder.

2. In the *New folder* popup as shown below (figure 2.27), type in your folder name in the textbox. Click OK.

3. Select the *master* Media folder to see which files are currently loaded, figure 2.28. Highlight the files there that you want to copy to your new folder (named *Photos of Garden*). Drag the files to the left and when you see the *Photos of Garden* folder become highlighted, then drop there. You can drag and drop any of the media from the *master* Media at any time. It flashes when the drop is successful.

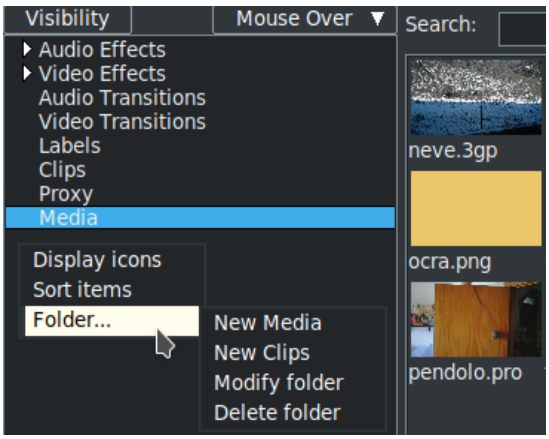


Figure 2.26: *New folder menu*

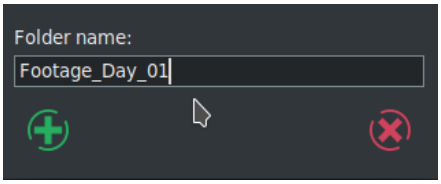


Figure 2.27: *Folder name*

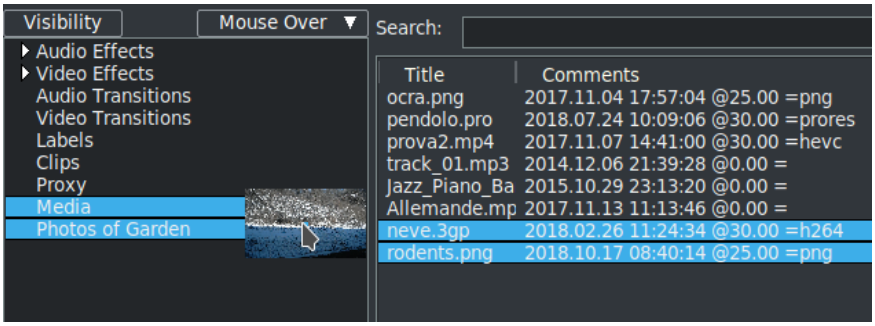


Figure 2.28: *The 'master' Media folder*

2.5.3 Modify Folder: sorting the media

In CINELERRA-GG all the virtual mediafiles loaded in the *Resources* window are collected in the `Media` folder without any kind of organization and sorting other

than alphabetical. The Media folder is considered a *master* folder because it is the basis for every other new folder we create. Another *master* folder is `Clips`, and in fact we can create new clips folders as well. If we do editing of large projects containing numerous mediafiles, a rational and selective organization of the source material becomes necessary. In fact, having the sources organized in folders, where they can be found and used easily, makes subsequent editing much faster, as well as minimizing confusion and the possibility of errors. This step, which is fundamental in the pro environment, is called *Logging* and also includes *screening* individual mediafiles in the Viewer window; *renaming* them rationally; and adding labels and descriptions in the *metadata*.

We start by creating one or more new folders (MediaUserBin). These will be empty and we should populate them with the mediafiles contained in the master Media folder. There are two options: manual *Drag and Drop* or automatic populating by setting *filters* (Smart Bins).

Drag and Drop: you drag mediafiles into the new folder one by one or in selected groups. The starting master folder can only be `Media` (or `Clips`, if we have created a new clip folder). Dragging is done by preserving the absolute path of the sources. If you open Modify Folder window you will notice that each imported mediafile occupies a different row characterized by its own path and name. See figure 2.29. The `Value` field containing paths can contain only 63 characters, and it may happen to find the names cut off. The full path and name can be found by right-clicking on the row, but this is inconvenient and time-consuming. The best solution is to drag the mediafiles while holding down the `Shift` key; this will return the relative path instead of the absolute path, with space gain.

Here is the filter that was generated with the original drop →

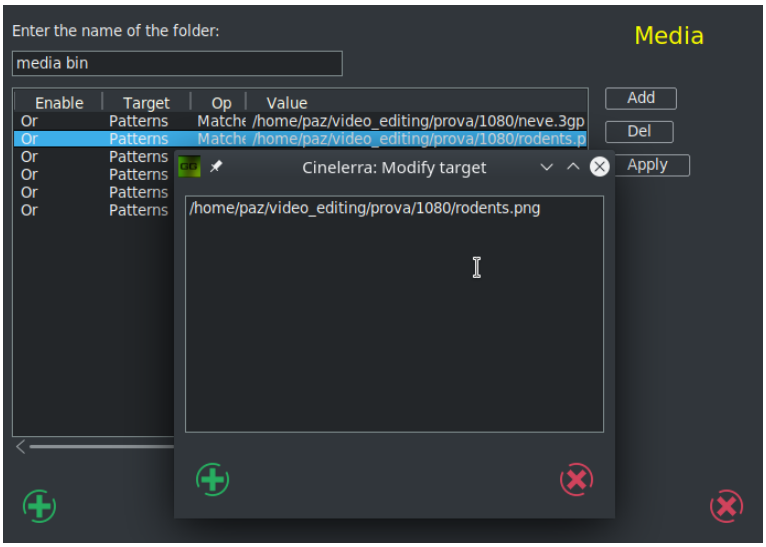


Figure 2.29: *Modify Folder*

Filters: you do automatic populating using filters (Smart Bins mode). You set the values of the `Enable`; `Target`; `Op` and `Value` fields to make up the required filtering. The folder will populate automatically by pulling from the `Media` master folder. The filter is the result of one row, but we can do more sophisticated filtering by adding more rows. The process will start from the first row propagating to subsequent rows always following the pattern from top to bottom. See figure 2.30.

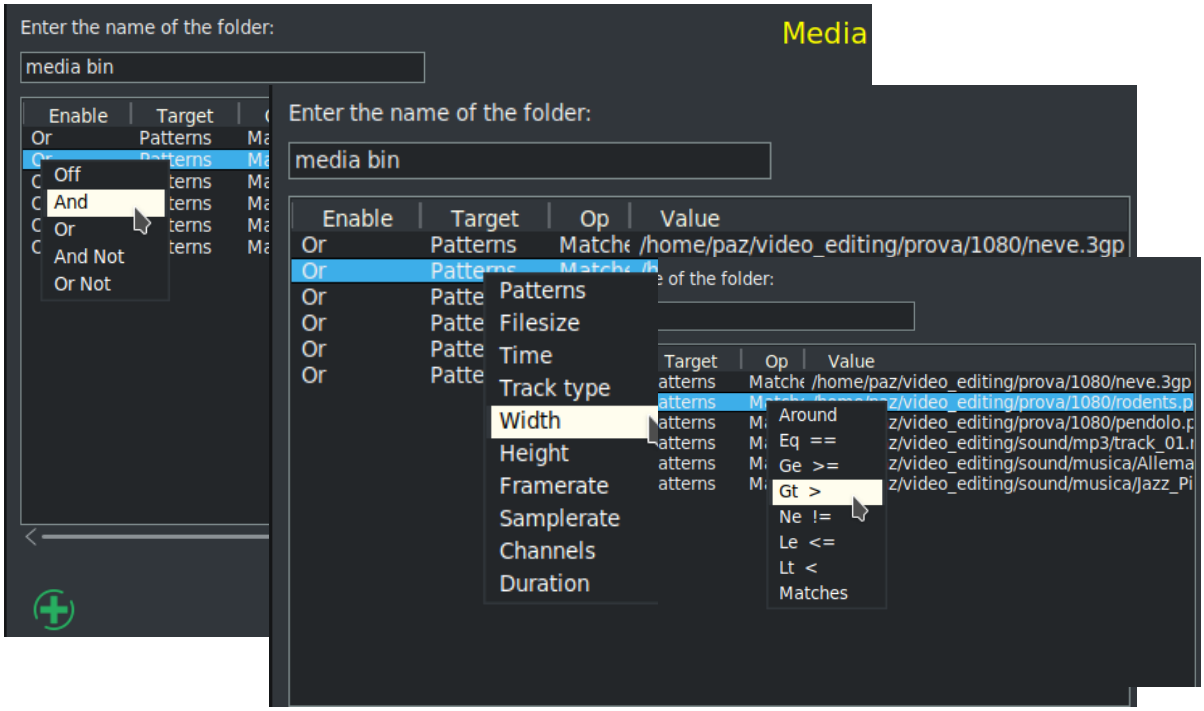


Figure 2.30: *The available choices for each field. Value field missing.*

Let's look in detail at all the options of Modify Folder.

When you bring up the Modify folder window, if you already have files in that folder, you will see filters that were generated automatically when you did a Drag and Drop. See figure 2.29.

If you click on the *Value* portion of that filter when it contains filenames that were dragged over, the entire set of files that are covered by the filter rules pops up. Now you can highlight a target filename that you would like to remove, and just erase that line and check the green checkmark for OK in that window plus OK in the "Modify folder" window (or click Apply there).

To delete the entire set of files listed on the filter rule, highlight the rule line and hit the *Del* button.

To add a new filter rule, click on the *Add* button which will automatically add a

default line after any current lines. The default line will be a line that matches everything in the master Media folder which is *Or Patterns Matches **. Click the right mouse button on the current field underneath the column header to see the choices available for each column.

Modifications will not be in effect until you click on the green arrow *OK* button or click on the *Apply* button. But once you hit Apply, clicking on the red X button will not undo your changes. The filter/search rules are applied in the order listed in the Modify folder window. You can change the order of the filter rules by highlighting the rule you want to move and then drag and drop to a new location.

Information about the columns and rules for the search filters in the Modify folder window follows.

Column headers:

Enable – this column is used to designate the state of that filter rule

Off disable the filter

And narrow your search; all of your search terms must be present

Or broaden your search to include more values

And Not exclude terms that do not contain the given value from your search results

Or Not include terms that do not contain the given value from your search results

Target – this column designates which media asset attribute to look at

Patterns each line contains a filename filter, matches the file path

Filesize number of bytes in a file

Time date file was created

Track Type track type of video, audio, or audio video (for both)

Width Format width

Height Format height

Framerate Video framerate

Samplerate Audio samplerate

Channels Number of audio channels

Duration Playback time in seconds – it uses the largest of audio or video if contains both

Op – boolean operators used to narrow or broaden the relationship between your search terms

Around about this value; use *+radius* for a search range: [target–radius... target+radius]

Eq equal to

Ge greater than or equal to

Gt greater than

Ne not equal

Le less than or equal

Lt less than

Matches exactly matches for strings

Value — the characteristic you are looking for with expressions that can be written with the following:

Number (decimal points are allowed and will be converted to a standard form):

inf representing infinity

#[TtGgMmKk] — where # represents a number and the characters mean:

inf	=	infinity
T	=	1099511627776
t	=	1000000000000
G	=	1073741824
g	=	1000000000
M	=	1048576
m	=	1000000
K	=	1024
k	=	1000

Scalar: Number
Number+Number

Date time:

date	=	year/month
date	=	year/month/day
time	=	hour:minute
time	=	hour:minute:second
date_time	=	date time

	day	=	#day | #days
	week	=	#week | #weeks
	month	=	#month | #months
Duration:	year	=	#year | #years
	delta	=	secs
	delta	=	mins:secs
	delta	=	hours:mins:secs

Around time: date time+duration

Around length: duration+duration

Table showing the allowed usage:

```
target: | eq ge gt ne le lt matches around
++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
    ↪
patterns | <---- strcmp ---------> + filter + nearest
file_size | <---- arithmetic -------+------> + radius
mod_time | <---- arithmetic -------+------> + radius
track_type | <---- member test ------+--------+------>
width | <---- arithmetic -------+------> + radius
height | <---- arithmetic -------+------> + radius
framerate | <---- arithmetic -------+------> + radius
samplerate | <---- arithmetic -------+------> + radius
channels | <---- arithmetic -------+------> + radius
duration | <---- arithmetic -------+------> + radius
```

where in the above, the filter can be:

filter	=	list
filter	=	token
list	=	[token]
list	=	[token]list
string	=	<chars>|<empty>
token	=	string
token	=	string*token

Examples with some caveats first:

1. *Or* generally includes or adds whereas *And* generally excludes or subtracts.

2. The filters only work on media in the folder; if there is no media, then there is nothing to search.

3. The examples below are not meant to be executed as a list of filters in Modify folder, they are just single line examples to indicate what can work.

4. Sort is by filename base name (directory path not included automatically) except when the *Around* operation is used and then it is sorted by that Target distance first and then filename.

Table 2.2: *Examples*

Enable	**Target**	**OP**	**Value**	**meaning**
Or	Patterns	Matches	*	all files from the Media Folder are included
And Not	Filesize	Lt	160000000	no files that are less than 160MB in size
Or Not	Time	Ge	2018/07/30 06:13:00	files not greater than or equal date
And	Duration	Eq	01:00	files included must have 60 secs. Duration
Off	Samplerate	Ne	44000	off for now, but may want to include later
And	Framerate	Around	24+1	files included all have 24 to 25 framerate
Or	Patterns	Matches	[*.mp4]	all files with the extension of mp4
Or	Time	Around	2018/08/02 06:00:00 + 02:00:00	files at 4 AM to 8 AM

2.5.4 Vicons & Aicons – aka Video Icons / Audio Icons

Vicons are video icons. Aicons are audio icons. By default the Resources window will play the first 5 seconds of video or audio waveform looped in the area occupied by the media icons (figure 2.31). This is enabled for the Media/Proxy folders in icon mode when the mouse pointer is inside the Resources window.

The waveform in the figure 2.32 is displayed in the Resources window in the color green for the 3 audio tracks. There is a colored bar on the top of each a-icon where the color is based on the Color Spectrum – the smaller the time duration, the redder the color; then as the time duration goes up, the color goes up so that you will go to green, then yellow, then blue, then really dark blue, then purple for the audio files 1 hour and over. There are various other colors between these

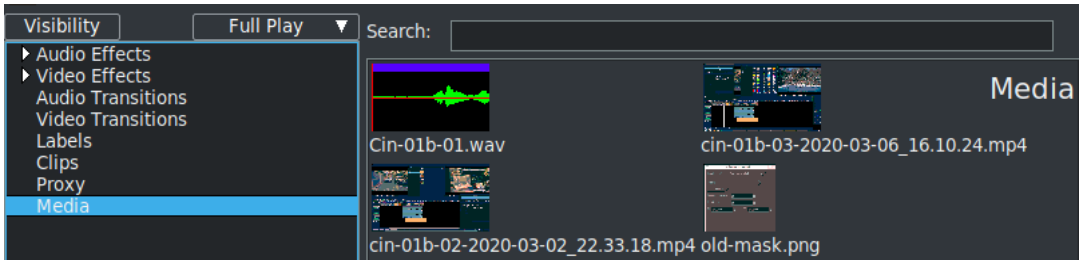


Figure 2.31: *Note "Full Play" mode and Vicons and Aicons in Media folder*

colors same as that seen in the color spectrum in the screenshot below. Colors are utilized from the hue wheel in the counter-clockwise direction. Note that the horizontal line in the middle of the a-icon is yellow/red representing the 2 audio tracks and is only red for mono.

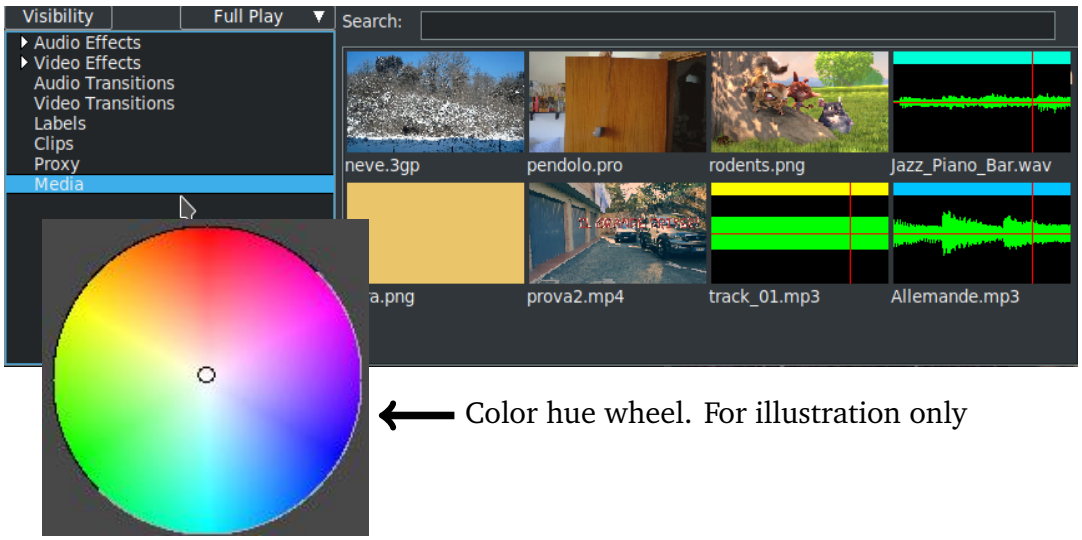


Figure 2.32: *Draw Vicons | Screenshot display various audio file lengths; red is shortest.*

Note that if in `Settings → Preferences` under the Appearance tab, you have unchecked *Use thumbnails in resource window* you will only have default icons and none of the above capabilities.

2.5.5 Resources Window Preview Mode

Preview mode can be used to pop up a window which draws the vicons/aicons thumbnails in a larger size. Preview or *draw vicons* mode is a helpful feature of CINELERRA-GG that lets you see and/or hear the first 5 seconds of the video for identification purposes. The Preview mode/playback toggle is to the right of the Visibility label as seen in the screenshot above. Preview mode is available for the Media, Proxy, Media User Bins, and Clips but clips are only 1 image.

When *Preview/draw vicons* is enabled/active, if you click on one of the video icons or an audio waveform icon, a view pops up that increases the size to 4 times the surface area larger. This makes it easier to see or hear if it is the media you are looking for in case you have many similar media files. To conserve memory, the video is stored 8 bits per pixel which results in low image quality while the audio is 16 bit. The reason for playing 5 seconds of a video for a vicon is that until the first I-frame, the media frequently does not decode properly. In other words, a lot of media does not begin at the *beginning* point and will not be properly rendered until enough data has been read to assemble a picture. You can increase the thumbnail size, clarity of pixels (memory size) and color mode but it takes a lot more memory. Change these values in `Settings` → `Preferences`, Appearance tab, right hand side of the Layout section – be aware that when you click OK, your session will re-initialize. You can also temporarily increase the preview mini-window by use of the mouse wheel up or down for media but not for clips.

There are 4 options for the preview mode.

1. *Full Play* is the default mode. This means all of the media will automatically play when the mouse is in the Resources window and you can use the left mouse button to click on specific media to see it pop up in a larger view. Audio only files do not play the audio until the icon is clicked on and the waveform aicon pops up into the 4x larger mode. *Full Play* includes the *Mouse Over* capabilities as described below as well as the Inter-View *Src Target* functions.

2. *No Play* mode is especially useful on smaller computers and for users who find the constant loop play to be somewhat distracting.

3. *Mouse Over* mode is activated by a single click on one of the vicons/aicons and deactivated with another single click over any of the icons. Once activated, whenever you just move the mouse over an icon, it automatically pops up the increased size preview. The first time in your session that you enable this feature, it may take a few seconds to load all of the icon previews into memory so be patient and just wait. *Mouse Over* mode makes it quick and easy to preview without having to drag the media to the viewer. You can still drag the media same as without preview enabled.

4. *Src Target* mode gives easy access to the Inter-View source target available by using the middle mouse button on media. There are 2 advantages to this mode – there is no 5 second play loop taking up cpu time and the popup allows for the use of the letter "`f`" on that popup to have it go to fullscreen mode. *Src Target* mode in any scenario never plays sound as that is nonsensical usage. After the initial click to pop media in this mode, you also have the *Mouse over* feature.

For any of the options, but not *No Play*, you can temporarily turn off that option by clicking on the top "Preview mode" option selected button using the middle mouse. This helps to avoid having the thumbnail get in the way of dragging or other functions. When you do, a line will be drawn through the current preview mode so that you are aware that it is in *No Play* mode until you click it again.

Note that if in `Settings` → `Preferences` under the Appearance tab, you have unchecked *Use thumbnails in resource window* you will only have default icons and no active previews.

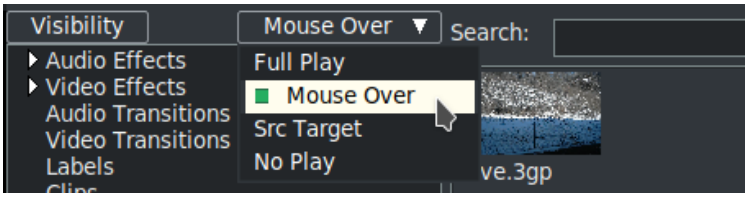


Figure 2.33: *The location of the Preview/Draw Icons mode.*

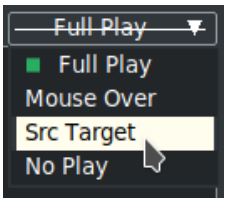


Figure 2.34: *Note the line through the mode.*

2.5.6 Moving clips/media from/to Resources window

If you have several media files loaded into the Resources window of one instance of CINELERRA-GG and want to load some of the same ones into another instance or just want a listing to save in a file for later use, you can do this with these set of steps:

Copy or paste a list of files in the Media Resources window:

1. create a highlighted selection of the desired media files in the media Resources window
2. right click on an unused portion of that window to bring up the popup menu
3. select the *Copy file list* item and a file list box will appear that contains the full path filenames
4. wipe the textbox using your standard copy/paste method to put the list of files in the copy buffer
5. in another CINELERRA-GG instance, choose the *Paste file list* of the media Resources window
6. paste the list of files, again using your standard paste method, into the new file list box; press OK

7. the status bar of the main window will be updated as the file list is loaded to the media folder (the purpose of displaying the status is simply to show that the load is progressing normally).

Obviously this *Paste file list* feature means you can create a list of files outside of CINELERRA-GG using an editor, highlight the names using your mouse, and then use *Paste file list* to load them into the media Resources window.

It is important to note that in the steps above, the Operating System cut and paste capabilities are in use for steps 4 and 6 as opposed to CINELERRA-GG's c/v shortcuts. Since the procedure varies among the distros, you will have to adapt to your specific one. For example, a usage for ubuntu consists of:

4. Ctrl-c to copy the list of files; open gedit; Ctrl-v to paste the list of files into gedit
5. Ctrl-c or the standard way using the right click to copy this list from gedit
6. Ctrl-v paste the list of files into the new file list box, and press OK

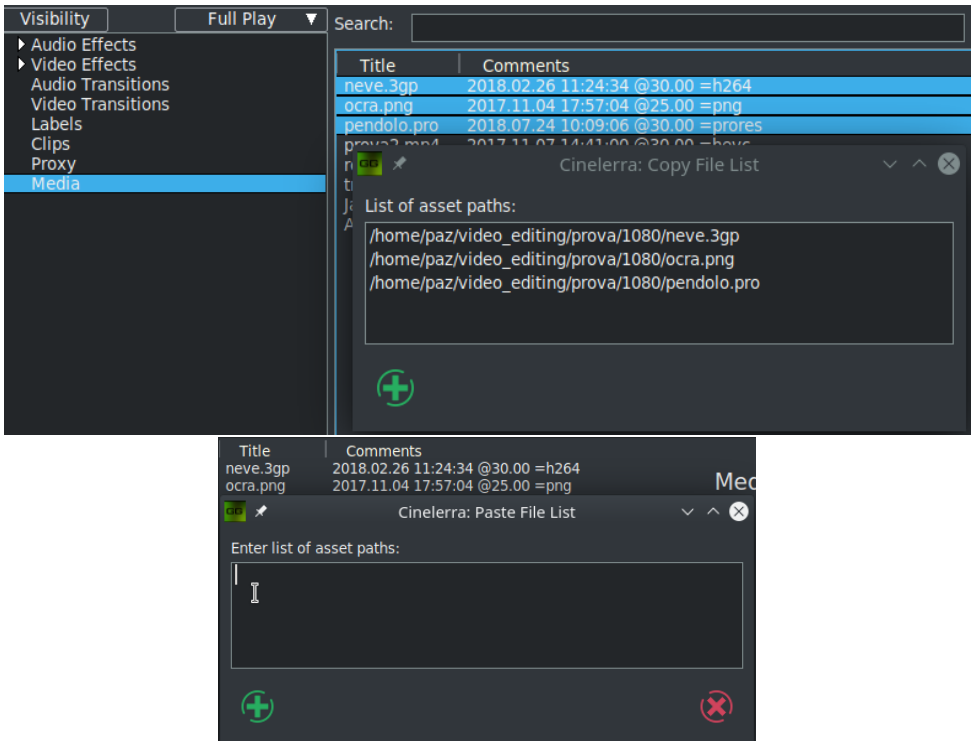


Figure 2.35: *Example of copy file list*

In the Figure 2.35, one instance of CINELERRA-GG has 3 items in the Media area highlighted that were copied to the file list. Note how it includes the full pathname.

In this screenshot on another instance of CINELERRA-GG, there are only 2 items in the media but the *Paste file list* box is ready to have the items inserted via the standard text box paste method. When that is done, the additional 3 media files will be available on this other instance too.

Another possible usage of this capability:

1. Right Click on the Clips Resources window and use the *Paste Clip* option to paste the Copy selection as a clip.
2. Similarly, by highlighting a clip in the Resources window and selecting its copy popup menu item using the right mouse button, that copy buffer can now be loaded onto the timeline.

2.5.7 Snapshot / Grabshot

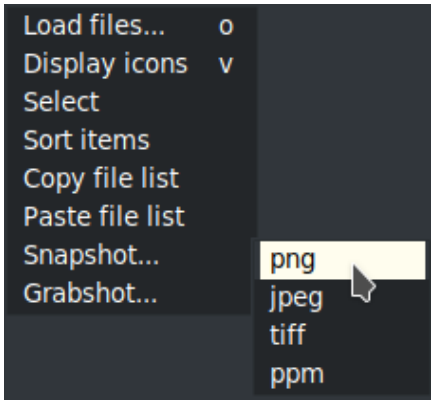


Figure 2.36: *Snapshot menu and choices*

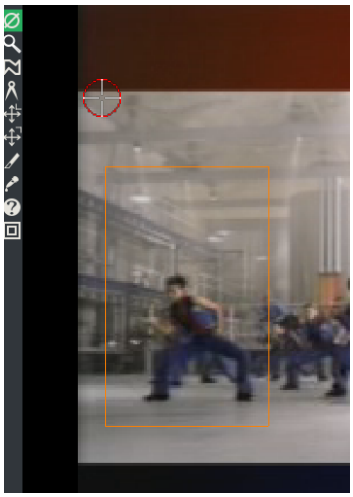


Figure 2.37: *Grabshot reticle & orange box*

To take a snapshot, perform the following steps:

1. set your timeline insert marker where you want the snapshot – this frame shows in the compositor
2. right click in an empty spot in the media folder and the popup shows snapshot as the 7^{th} item down

3. highlight that and the submenu comes up allowing you to choose png, jpg, ppm or tiff

The snapshot shows up in the Media folder. It is saved by default in `/tmp` as

`snap_date-time.ext` BUT you can change the default directory path in `Settings` → `Preferences` → `Interface tab` in the right hand side of the Editing section.

Grabshot is the last menu item. A red circle reticle can be moved to the area to grab; use left mouse drag to surround an area; and right click to grab.

2.6 Other Options and Other Windows

2.6.1 Transport Controls

Transport controls are useful for navigation and for playing media. Each of the Viewer, Compositor, and Program windows has its own transport panel. The controls generally all contain a colored tooltip when you mouse over the control, providing a hint of their function and shortcuts for usage.

The transport panel is controlled by the keyboard as well as the graphical interface. For each of the operations it performs, the starting position is the position of the insertion point in the Program window and the slider in the Compositor and Viewer windows. The ending position is either the end or start of the timeline or the end or start of the selected region if there is one.

The orientation of the end or start depends on the direction of playback. If it is forward the end position is the end of the selected region. If it is backward the end position is the start of the selected region. The insertion point moves to track playback. When playback stops, the insertion point stays where playback stopped. Thus, by playing back you change the position of the insertion point. The keyboard interface of either the numeric pad or alternative keys has more speeds with the addition of *Forward Slow* (2) and *Reverse Slow* (5). Hitting one of the programmed numeric keys, 2 for example, starts playback. Hitting that same key a second time, stops it. The shortcuts section of this manual as well as a Shell Command available from the CINELERRA-GG main window has a listing of each of the keys.

When using frame advance functions the behavior may seem odd. If you frame advance forward and then frame advance backward, the displayed frame does not change. This is because the playback position is not the frame but the time between two frames. The rendered frame is the area that the playback position

crosses. When you increment the time between two frames by one and decrement it by one, you cross the same frame both times and so the same frame is displayed. There is an option in `Settings → Preferences, Appearance tab` to *Always show next frame* that may help make this clearer for some users.

The transport behavior changes if you hold down Ctrl when issuing any of the transport commands. This causes the starting point to be the In point if playing forward and the Out point if playing backward. If playing forward, the Out point becomes the ending point and if playing backward, the In point becomes the ending point. If no In/Out points are specified, the behavior falls back to using the insertion point and track boundaries as the starting and ending points.

The transport behavior also changes if you hold down the Shift key along with KeyPad 1–6. If normally audio is included in the play, it will be removed and if normally audio is not included in the play, it will be added.

2.6.2 Zoombar

The compositor has zoom capability. The pull-down menu on the bottom of the compositor window has a number of zoom options. When set to Auto the video is zoomed to match the compositor window size as closely as possible. When the video is zoomed bigger than the window size, you can use scrollbars to scan around or if the zoom icon is enabled, the middle mouse button can be used to zoom in or out the video.

The zoom toggle also causes the Compositor window to enter zoom mode. In zoom mode, clicking in the video output zooms in while shift-click in the video output zooms out. If you have a wheel mouse, rotating the wheel zooms in or out too. Zooming in or out with the zoom tool does not change the rendered output. It is merely for scrutinizing video or fitting it in the desktop. Playing video on the compositor when zoomed to any size other that 100%, the original size, requires CINELERRA-GG to do extra processing steps. This could affect performance on slower systems.

2.6.3 Show Overlays

Color Coded Keyframe Curves are a big feature in the *Show Overlays* window because by changing the colors to suit the user, it helps to remove confusion from multiple curves on the track canvas. They can be viewed from the pulldown menu of `Window → Show overlays` but they will operate the same as when used from the *View* pulldown menu. The *Color Coded Keyframe Curves* have distinct colors associated with each type for ease of identification. By clicking LMB on the *Color Ball* to the right of any keyframe type in the *Show overlays* menu you have the ability

to change the colors to whatever works best for your video. The color ball changes made will be retained across sessions.

There is a line separating the first 4 items, which are just non-automation type settable values as opposed to *auto* keyframe types. The color is not changeable for the 4 items of Mode, Pan, Mask, and Hard edges which simply display their symbol icon.

Figure 2.38 displays the Show overlays popup with all of its options and color coded types such as yellow for Speed and blue for Camera Z. Upon clicking on the associated *color ball* to the right of any keyframe type, for example *Fade* in this screenshot, the color wheel palette window pops up so that you can manipulate the color as desired.

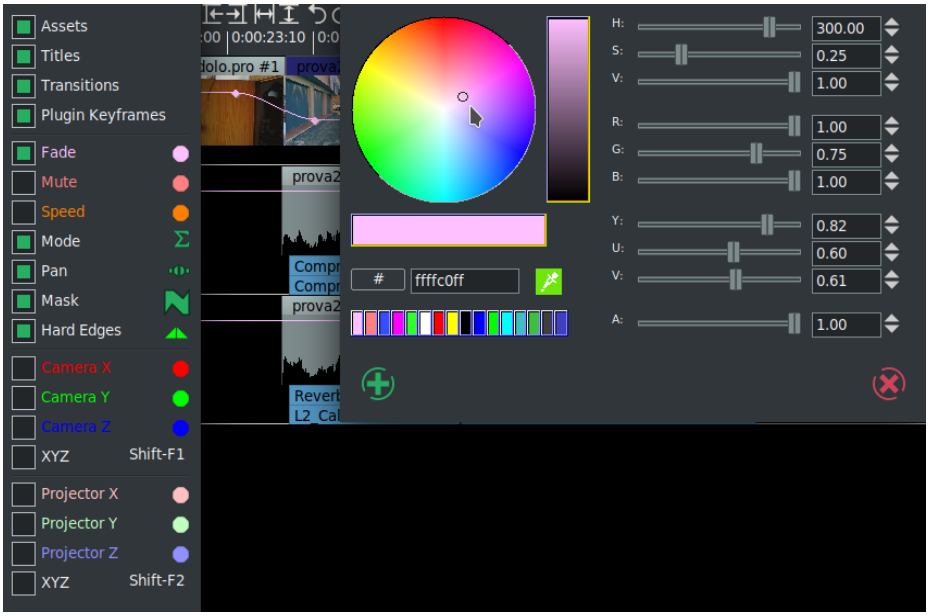


Figure 2.38: *Show Overlays window on the left with the Color ball window to the right to set color*

Figure 2.39 shows several color coded lines for different keyframes along with the Fade slider for manipulation. The slider is in the same color as the color coded keyframe type line which is the same color as in the *Show overlays* window.

Overlays Window Nuances:

The Overlays window is an alternative to the main track canvas *View* pulldown, and thus the order is mostly maintained to match each other. To make it easier to get a quick temporary look at a specific option, there is a shortcut of Shift-LMB (left mouse button) that can be used as opposed to having to uncheck everything that is currently checked and then having to recheck them on when done. Here

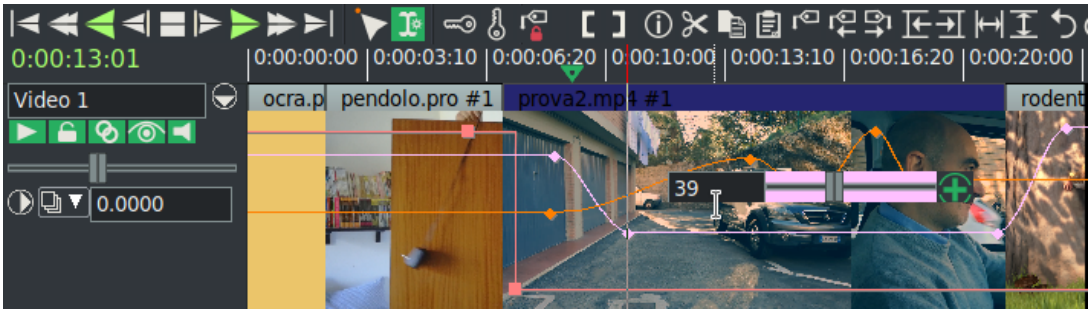


Figure 2.39: *Lines are colored here on the timeline as designated in Show Overlays*

is a list of how they work. Keep in mind that if the Expander on the patchbay is enabled, you still see the track.

- Shift+LMB (left mouse button) in the Overlays Window on a checkbox will turn off all other checkboxes except for the one you are on. Then this named box will have outline for a *hot* spot.
- Shift+LMB on this *hot* spot will return to *cool* of the previous settings with all of the previous checkboxes checked again.

- Shift+LMB on a non-*hot* spot will simply check or uncheck a box and there is no previous state.
- This all works in conjunction with the *View* pulldown menu which, of course, has no hot spots.
- Caveat #1 - Shift+LMB on the top 4 choices of Assets, Titles, Transitions, Plugin Keyframes will turn off all of the checkboxes below because it makes sense to do so.
- Caveat #2 - Shift+LMB on the Autos will not turn off Assets, Titles, Transitions, or Plugin Keyframes because you need to be able to see what is going on.
- Caveat #3 - XYZ toggle on/off of Camera and Projector are not affected.

2.6.4 Sound Level Meters Window

An additional window, the levels window, can be brought up from the *Window* pulldown. The levels window displays the output audio levels after all mixing is done. The visible range of the sound level meters is configurable in `Settings` → `Preferences, Interface tab` under the Operations section.

Figure 2.40: *Original Settings — cool spot*

Figure 2.41: *Note Titles box hot spot*

Figure 2.42: *Cam/Proj XYZ toggle to fine tune*

Sound level meters can be toggled in the viewer and compositor windows with the show meters button. They also appear in the patchbay when the track is expanded and in the recording monitor when audio is being recorded.

The sound levels in the levels window, compositor, and viewer correspond to the final output levels before they are clipped to the sound card range. In the record monitor they are the input values from the sound card. In the patchbay they are the sound levels for each track after all effects are processed and before down-mixing for the output. Most of the time, audio levels have numerical markings in dB but in the patchbay there is not enough room.

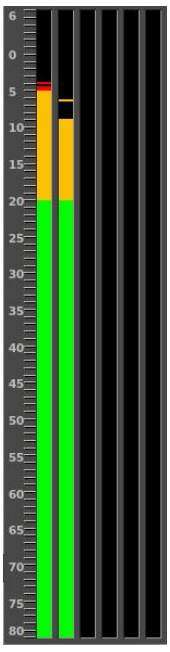


Figure 2.43: *Sound Level Meters Window*

The sound level is color coded as an extra means of determining the sound level. Even without numerical markings, the sound level color can distinguish between several ranges and overload. Look at the color codings in a meter with numerical markings to see what colors correspond to what sound level. Then for meters in the patchbay in expanded audio tracks, use the color codings to see if it is overloading.

Be aware that sound levels in CINELERRA-GG can go above 0 dB. This allows for not only seeing if a track is overloading but how much information is being lost by the overloading. Overloading by less than 3 dB is usually acceptable. While overloading is treated as positive numbers in CINELERRA-GG, it is clipped to 0 when sent to a sound card or file.

CHAPTER 3

Project and Media Attributes

When you play media files in CINELERRA-GG, the media files have a certain number of tracks, frame size, sample size, and so on. No matter what attributes the media file has, it is played back according to the project attributes. So, if an audio file's sample rate is different than the project attributes, it is resampled. Similarly, if a video file's frame size is different than the project attributes, the video is composited on a black frame, either cropped or bordered with black.

The project attributes are adjusted in `Settings → Format` (figure 3.1) or can be created in `File → New`. When you adjust project settings in `File → New`, a new empty timeline is created. Every timeline created from this point on uses the same settings. When you adjust settings in `Settings → Format`, media on the timeline is left unchanged. But every timeline created from this point uses the same settings.

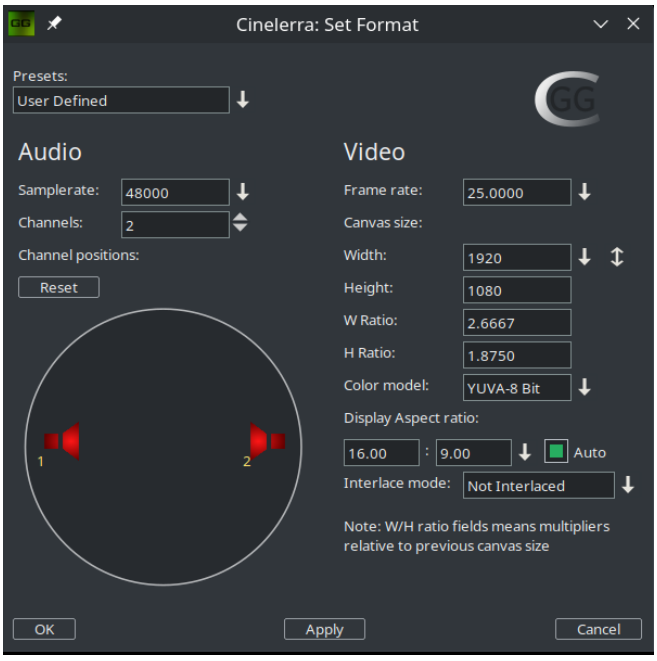


Figure 3.1: *Set Format window - note the Audio Channel positions*

In addition to the standard settings for sample rate, frame rate, and frame size

(canvas size), CINELERRA-GG uses some less traditional settings like channel positions, color model, and aspect ratio. The aspect ratio refers to the display aspect ratio (DAR).

Edit decision lists , the EDL stored in XML, save the project settings. Formats which contain media but no edit decisions just add data to the tracks. Keep in mind details such as if your project sample rate is 48 kHz and you load a sound file with 96 kHz, you will still be playing it at 48 kHz. Or if you load an EDL file at 96 kHz and the current project sample rate is 48 kHz, you will change it to 96 kHz.

The New Project window has some options that are different than the Set Format window as you can see by comparing figure 3.1 above with this figure 3.2. Mostly notably is the field for a directory path and a Project Name.

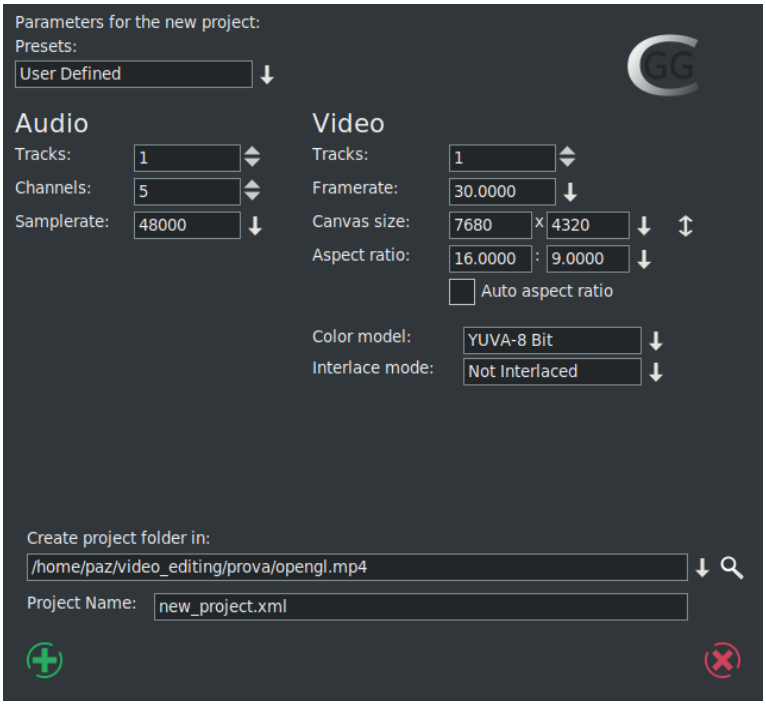


Figure 3.2: *New Project dialog window*

Explanation of the various fields is described next.

3.1 Audio attributes

Presets: select an option from this menu to have all the project settings set to one of the known standards. Some of the options are 1080P/24, 1080I, 720P/60, PAL, NTSC, YouTube, and CD audio.

Tracks: (in New Project menu only) sets the number of audio tracks for the new project. Tracks can be added or deleted later, but this option is on the New Project menu for convenience.

Samplerate: sets the samplerate of the audio. The project samplerate does not have to be the same as the media sample rate that you load. Media is resampled to match the project sample rate.

Channels: sets the number of audio channels for the new project. The number of audio channels does not have to be the same as the number of tracks.

Channel positions: the currently enabled audio channels and their positions in the audio panning boxes in the track patchbay are displayed in the channel position widget in the Set Format window. You can see this display on the left side in figure 3.1 above. Channel positions are not in New Project window.

The channels are numbered. When rendered, the output from channel 1 is rendered to the first output track in the file or the first sound card channel of the sound card. Later channels are rendered to output tracks numbered consecutively. The audio channel positions correspond to where in the panning widgets each of the audio outputs is located. The closer the panning position is to one of the audio outputs, the more signal that speaker gets. Click on a speaker icon and drag to change the audio channel location. The speakers can be in any orientation. A different speaker arrangement is stored for every number of audio channels since normally you do not want the same speaker arrangement for different numbers of channels.

Channel positions is the only setting that does not affect the output necessarily. It is merely a convenience, so that when more than two channels are used, the pan controls on the timeline can distinguish between them. It has nothing to do with the actual arrangement of speakers. Different channels can be positioned very close together to make them have the same output.

3.2 Video attributes

Tracks: (in New Project menu only) sets the number of video tracks the new project is assigned. Tracks can be added or deleted later, but options are provided here for convenience.

Framerate: sets the framerate of the video. The project framerate does not have to be the same as an individual media file frame rate that you load. Media is reframed to match the project framerate.

Canvas size: sets the size of the video output . In addition, each track also has its own frame size. Initially, the New Project dialog creates video tracks whose size matches the video output. The video track sizes can be changed

later without changing the video output. We have: Project size = $Width \times Height$, pixels = canvas size = output size .

W/H Ratio Sets the ratio of the new canvas size (Width, Height) from the old (previous) canvas size (Width, Height).

$$\text{W Ratio} = \frac{W_f}{W_i} \qquad \text{H Ratio} = \frac{H_f}{H_i}$$

with W_f/H_f: final Width and Height; W_i/H_i: initial Width and Height.

The new canvas size is recalculated based upon a certain factor in the `W Ratio`, `H Ratio` fields. A practical use-case: the current resolution is 640 × 480, and for some reason you want Width to be 1.33 times bigger. You don't have to calculate what 640 × 1.33 is; you type 1.33 into the `Width` input instead, and CINELERRA-GG calculates it for you. W/H Ratio works as a local calculator. Warning: if you vary W/H Ratio without adjusting Display aspect ratio, we may get non-square pixels resulting in anamorphic frame distortion.

Display aspect ratio: sets the aspect ratio; this aspect ratio refers to the display aspect ratio (DAR). The aspect ratio is applied to the video output (canvas). It can be convenient to vary the size of the canvas in percentage terms, instead of having to calculate the number of Width x Height pixels. The aspect ratio can be different than the ratio that results from the formula: $\frac{h}{v}$ (the number of horizontal pixels divided into the number of vertical pixels). If the aspect ratio differs from the results of the formula above, your output will be in non-square pixels.

Auto aspect ratio: if this option is checked, the `Set Format` dialog always recalculates the Aspect ratio setting based upon the given Canvas size. This ensures pixels are always square.

Color model: the internal color space of CINELERRA-GG is X11 sRGB without color profile. CINELERRA-GG always switches to sRGB when applying filters or using the compositing engine. Different case for decoding/playback or encoding/output; the project will be stored in the color model video that is selected in the dropdown. Color model is important for video playback because video has the disadvantage of being slow compared to audio. Video is stored on disk in one colormodel, usually a YUV derivative. When played back, CINELERRA-GG decompresses it from the file format directly into the format of the output device. If effects are processed, the program decompresses the video into an intermediate colormodel first and then converts it to the format of the output device. The selection of an intermediate colormodel determines how fast and accurate the effects are. A list of the current colormodel choices follows.

RGB-8 bit Allocates 8 bits for the R, G, and B channels and no alpha. This is normally used for uncompressed media with low dynamic range.

RGBA-8 bit Allocates an alpha channel to the 8 bit RGB colormodel. It can be used for overlaying multiple tracks.

RGB-Float Allocates a 32 bit float for the R, G, and B channels and no alpha. This is used for high dynamic range processing with no transparency.

RGBA-Float This adds a 32 bit float for alpha to RGB-Float. It is used for high dynamic range processing with transparency. Or when we don't want to lose data during workflow, for example in color correction, key extraction and motion tracking. Note: even if CINELERRA-GG outputs fp32, exr/tiff values there are normalized to 0-1.0f.

YUV-8 bit Allocates 8 bits for Y, U, and V. This is used for low dynamic range operations in which the media is compressed in the YUV color space. Most compressed media is in YUV and this derivative allows video to be processed fast with the least color degradation.

YUVA-8 bit Allocates an alpha channel to the 8 bit YUV colormodel for transparency.

In order to do effects which involve alpha channels , a colormodel with an alpha channel must be selected. These are RGBA-8 bit, YUVA-8 bit, and RGBA-Float. The 4 channel colormodels are slower than 3 channel colormodels, with the slowest being RGBA-Float. Some effects, like fade, work around the need for alpha channels while other effects, like chromakey, require an alpha channel in order to be functional. So in order to get faster results, it is always a good idea to try the effect without alpha channels to see if it works before settling on an alpha channel and slowing it down.

When using compressed footage, YUV colormodels are usually faster than RGB colormodels . They also destroy fewer colors than RGB colormodels. If footage stored as JPEG or MPEG is processed many times in RGB, the colors will fade whereas they will not fade if processed in YUV. Years of working with high dynamic range footage has shown floating point RGB to be the best format for high dynamic range. 16 bit integers were used in the past and were too lossy and slow for the amount of improvement. RGB float does not destroy information when used with YUV source footage and also supports brightness above 100 %. Be aware that some effects, like Histogram, still clip above 100 % when in floating point. See also 17.3, 17.4 and A.3.

Interlace mode: this is mostly obsolete in the modern digital age, but may be needed for older media such as that from broadcast TV. Interlacing uses two fields to create a frame. One field contains all odd-numbered lines in the image; the other contains all even-numbered lines. Interlaced fields are stored in alternating lines of interlaced source footage. The alternating lines missing on each output frame are interpolated.

A tip on how to eliminate Letterbox/Pillarbox black bands in the Set Format window can be found here: How to remove letterbox/pillarbox bands.

3.3 Best practice in pre-editing

CINELERRA-GG supports the simultaneous presence in the Timeline of sources with different frame sizes and frame rates. However, audio/video synchronization problems may occur due to their different timing.[1] Plugins that rely on the timing of each frame, for example *Motion* and *Interpolate* plugins, may have problems when used at the same time with engines which increase frame rate. Frame rate per definition cannot be increased without either duplicating some frames or generating them in some intelligent way. But to work reliably, the *Motion* plugin requires access to all actual frames. These kinds of plugins (and also the rare cases of audio/video desync) explicitly require the *Play every frame* option.

There is no problem as long as the source fps, project fps, and destination fps are identical. In most cases, high frame rates such as 120 or 144 or any fps, will be just fine for *Motion* provided that source footage all has the same frame rate.

But when *project* and *source* frame rates are different (or *project* and *rendered* fps), then the CINELERRA-GG engine has to either duplicate (interpolate) some frames or throw some away. Because of this, the audio tracks and the timeline get out of sync with such accelerated (or slowed down) video. And to make *Motion* plugins reliably calculate interframe changes, you have to ensure the consistent frame numbers and frame properties.

Generally, best practice is to perform the following sequence of preparations for video editing.

1. Motion stabilization, and maybe some other preparations, to improve the quality of the source video is best done under the properties identical to the properties of the original video; it may be different codec, but same frame size and same frame rate. For stabilization you can use ffmpeg command line plugins called *vidstabdetect* and *vidstabtransform*.

2. To have a workflow at the highest quality it may be convenient to convert the sources into image sequences (e.g. OpenEXR). Especially if we want to exchange files with other Color or Compositimg programs that preferably use image sequences.

[1] credit to sge and Andrew Randrianasulu

3. Uniform the color models. It is convenient to unify the color models of the sources because they would give different and inconsistent results with each other once displayed in the Compositor window.

4. If we intend to do some color correction or compositing with VFX, it is convenient to do some de-noising on the sources to make their pixels more homogeneous and suitable for post processing. De-noising is a heavy operation for the system so it may be convenient to do it in pre-editing.

5. If you need to alter the frame rate, for example because different source clips have different frame rates, then recode all the necessary clips to the same future project frame rate. Here frame sizes can still have different sizes, but frame rates should be all the same. If you need to change frame rate of some restricted part, particularly when smooth acceleration/deceleration is needed, it can be done in timeline. But if frame rate has to be changed only due to different source fps, it is better to do it during the preparation stage.

CINELERRA-GG does not have color management , but we can still give some general advice on how to set color spaces:

1. Profiling and setting the monitor:
   source: *sRGB* → monitor: *sRGB* (we get a correct color reproduction)
   source: *sRGB* → monitor: *rec709* (we get slightly dark colors)
   source: *sRGB* → monitor: *DCI-P3* (we get over-saturated colors)

   source: *rec709* → monitor: *rec709* (we get a correct color reproduction)
   source: *rec709* → monitor: *sRGB* (we get slightly faded colors)
   source: *rec709* → monitor: *DCI-P3* (we get over-saturated colors)

2. It would be better to set the project as RGB(A)-FLOAT, allowing system performance, because it collects all available data and does not make rounding errors. If we can't afford it, starting from YUV type media it is better to set the project as YUV(A)8, so as not to have a darker rendering in the timeline. On the contrary, if we start from RGB signals, it is better to use RGB(A)8. If we don't display correctly on the timeline, we'll make adjustments from the wrong base (metamerism) and get false results.

3. Having correct color representation in the Compositor can be complicated. You can convert the imput *YUV color range* to a new YUV color range that provides more correct results (i.e. MPEG to JPEG). The `Colorspace` plugin can be used for this conversion.

4. Among the rendering options always set the values
   `color_trc=...` (gamma correction)

`color_primaries=...` (gamut)
`colorspace=...` (color spaces conversion, more depth-color);
or
`colormatrix=...` (color spaces conversion, faster).

These are only metadata that do not affect rendering but when the file is read by a player later they are used to reproduce the colors without errors.

For more tips on how CINELERRA-GG processes colors on the timeline see Color Space and Color Range Affecting Playback, Automatic "Best Model" Media Load and Overview on Color Management.

CHAPTER 4

Load, Save, and the EDL

The media files loaded into the Resources window are called *assets* and are virtual references to the real media files on disk. The original sources are not changed and so we have non-destructive editing.

There are many supported file formats that can be loaded and rendered to, that is to say, imported and exported. The format of the file affects what CINELERRA-GG does with it. Some file formats are very slow to display on the timeline, especially video which is highly compressed. Drawing video thumbnails, picons, on the timeline can be very slow. (You can disable picon drawing for these files with the *draw media* toggle in the patchbay to speed up operations).

4.1 EDL — Edit Decision List

When CINELERRA-GG saves a file, it saves the EDL, Edit Decision List, of the current project but does not save any media (Non-Destructive Editing). Edit decision lists, more commonly referred to as the EDL, are generated by CINELERRA-GG for storing projects. The EDL contains all the project settings and locations of every edit. Instead of media, the file contains pointers to the original media files on disk. EDL files are specific to CINELERRA-GG.

The EDL files generally have an extension of `.xml`. The purpose of the EDL is to separate the media from all of the editing operations so that the original media remains intact. When the `.xml` file is loaded, changes to the attributes of the current project are made based on the EDL. Edit decision lists are text files which means they can be edited in a text editor. EDL and XML are used interchangeably.

4.2 Supported File Formats

There are basically 2 kinds of supported file formats, native and ffmpeg. With the addition of ffmpeg, the majority of the supported file formats you will be using comes via this thirdparty package. There are hundreds of ffmpeg file format and codec combinations. This set of possibilities includes qt (quicktime), avi (audio-

video interleave), mp4, mp3, mov, mpeg, m2ts, ts, wmv, mts, mpg, flv, mkv, webm, webp, ProRes and many more.

The other supported formats, referred to as *native*, include the following: AC3, Apple/SGI AIFF, Sun/NeXT AU, FLAC , Microsoft WAV, Raw DV and PCM, MPEG Audio and Video.

Still Images: JPEG/EXR/GIF/PNG/PPM/TGA/TIFF. Note: even if CINELERRA-GG outputs fp32, exr/tiff values there are normalized to 0-1.0f.

MPEG Files: What is an MPEG file? A very common file format is MPEG because it works with many cameras and televisions. Mpeg2 video, an elementary codec stream for mpeg files, is the most common format. To read this format you need to decode the mpeg stream. You can read and write mpeg natively. Mpeg video encoding is done separately from mpeg audio encoding when using the native file format, meaning that 2 passes are required and then they have to be muxed together. However, if using ffmpeg it is rendered in only 1 pass. DVD uses MPEG as does NTSC and Pal.

4.2.1 Working with Still Images

Still images are played from 1 to any number of times, over and over; they have no duration. You can load still images on video tracks just like you do for any video file. When loaded on the track, use the down arrow on the timeline so you can see the single frame. To extend the length of the image, drag its boundaries just as you would do with regular video media. You can drag the boundaries of a still image as much as you want. Images in CINELERRA-GG have the ability to be dragged to an infinite length. Alternatively, you can define the initial length of the loaded images. The parameter is set in the Images section of the `Settings` → `Preferences` → `Recording` window.

Unless your original material comes from a digital source using its best resolution (like a digital camera), the first thing you might have to do before you can use it is to somehow capture the assets into a usable digital medium. For old photos, paper maps, drawings or diagrams, you can scan them into a file format like PNG, TIF, TGA or JPG files by using a digital scanner.

Rendering a video to a single image causes the final image file to be overwritten for every timeline position. The rendered file is a single still image of the last frame of the video.

4.2.2 Timelapse Sequence of Images, and Animation

The next areas covered in this section are the following: Filelist format such as jpeglist and Image2ffmpeg.

In order to be reasonably fast to use, you will most likely want to prepare them by creating a script and then load by utilizing this file generated script.

4.2.2.1 Filelist format

An image sequence is a series of ordered still pictures; for example a bunch of camera shots, frames of an animated scene, or series of frame shots. These can be loaded as multiple files. For timelapse sequences, as the size of camera images increases to 70 megabytes and beyond, and more images can be stored on a memory stick, more cache, memory, and system resources (such as file descriptors) are used by CINELERRA-GG to load the images when you use the *concatenate tracks* or *paste at insertion point* strategies. It is very time consuming and resource consuming when each of the image files is loaded and concatenated as edits, and it also plays super poorly. Here is an alternative to the usual *load*. This technique may also be useful for just a bunch of pictures.

File lists formats can be utilized in some way for the following list of types of *Sequence files* The first line of the sequence list file identifies the list codec.

PNGLIST = *.png	PPMLIST = *.ppm	TGALIST = *.tga	TIFFLIST = *.tiff
EXRLIST = *.exr	CR2LIST = *.cr2	JPEGLIST = *.jpg	GIFLIST = *.gif

Using the example of jpeg's, the jpeg list sequence file type is the easiest and fastest way to access a sequence of jpg images as a single asset. First build a jpeglist sequence file and name it something like jpeglist.sh. There is an example script of how to do this in the Auxiliary Programs section of the Appendix (C.2). Once the jpeglist.sh file is built and made executable you can then run it similar to this line:

```
./jpeglist.sh /<same_path>/outputfile.jpgs /<same_path>/inputfilesDSC*.jpg
```

The <path> must be the same on both outputfile and inputfiles so that they can be successfully loaded in CINELERRA-GG. Since this creates outputfile list as a single asset, the memory demand and access time is much lower. When you load the outputfile in CINELERRA-GG, you might need to set *Try ffmpeg last* if ffmpeg does not work with that particular type of sequence files.

An example output file from running this script residing in the directory where `DSC*.jpg` files exist is shown below. To use this, turn off ffmpeg probes first, and open `outputfile.jpgs` using File → Load files. Of course, you can edit this file to change frame rate or names of files to include. The width and height should always be the same size for each included file and the file type must be the same. For example, if you include a JPEG file in a GIF list, you will get undesirable results.

Listing 4.1: *Example: outputfile.jpgs*

```
JPEGLIST
# First line is always JPEGLIST and all files must be JPEGs
# Frame rate:
29.970030
# Width -all files must be the same size in width:
6016
# Height -all files must be the same size in height:
4016
# List of image files follows
./DSC04948.jpg
./DSC04949.jpg
./DSC04950.jpg
./DSC04951.jpg
...(files in between)
./DSC04997.jpg
./DSC04998.jpg
```

Another advantage of using image sequences of exr, jpeg, png, ppm tiff, and tga is that if you then render to anoher of the same sequence type, a direct copy speed up can be used for any of the unmodified frames.

4.2.2.2 Image2ffmpeg

Image2file format is an alternative method to open an image sequence via ffmpeg. To do this, create 2 files in the same directory as the `DSC*.jpg` files named: `DSC0%04d.opts`, and `DSC0%04d.jpg`. `DSC0%04d.opts` should contain the following lines which have to be modified to fit your exact requirements for duration, start_number, and frame_rate.

Listing 4.2: *Example DSC0%04d.opts*

```
loglevel=verbose
threads=auto
format=image2
codec=mjpeg
start_number=4948
frame_rate=29.97
duration=17.36
```

Example of the contents of the file `DSC0%04d.jpg` would be just a single line as: JPEG In this case, make sure *Try ffmpeg first* is enabled, and load `DSC0%04d.jpg`. This will access the media using ffmpeg which is slower so be patient.

4.2.3 Raw Image Format of Some Digital Cameras & Probe Order

Note: requires some expert knowledge. Raw digital camera images are a special kind of image file that CINELERRA-GG can load. Dcraw, as used by CINELERRA-GG, is Dave Coffin's open-source computer program which reads many raw-image formats typically produced by many earlier and current digital cameras. Currently over 700 of the types of cameras it recognizes are listed at:

`https://www.dechifro.org/dcraw/`

For example, included are numerous models of Canon, Fuji, Kodak, Nikon, Olympus, and Panasonic as well as many others. Because ffmpeg tries to load *any and every* file if *Try Ffmpeg first* is enabled, it will make an attempt to load Raw Camera files first before any other file driver gets the chance. In addition, there is the possibility that dcraw could conflict with the standard TIFF format, since it might be seen as format type *tiff-pipe*. Therefore it is necessary to specifically enable CR2 and either move it to the top or disable *FFMPEG_Early* and enable *FFMPEG_late* in the *Probe Order* as described in another section (4.3.3 and 9.1). These changed settings will be retained across CINELERRA-GG sessions in `.bcast5`. Usage of Raw Camera mode generally requires more in depth knowledge of your specific camera.

The first screenshot in figure 4.1 as in `Settings → Preferences → Playback A` Tab, shows the default checked settings of *Interpolate CR2 images* and *White balance CR2 images* which display the raw images in a way that you expect. However, you may want to uncheck them to ensure that no program manipulation has modified your images so that you can add plugins or make your own modifications. Unchecked indicates that the images are as closest as possible to unadulterated raw.

The second screenshot showing CR2 for Raw Camera highlighed/enabled in the Preferences Probes' screen.

The final screenshot showing the Resources Asset Info displaying the File format as Raw Camera.

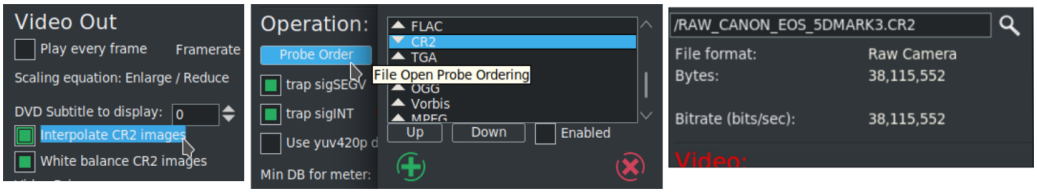


Figure 4.1: *Screenshots for RAW images*

4.3 Loading Files

All data that you work with in CINELERRA-GG is acquired either by loading from disk or recording from a device. This section describes loading. To bring up the Load window go to the *File* pulldown and choose Load Files (figure 4.2). Next *Select files to load*, and click ok (the green checkmark) or *Apply*. When you use the Apply button, the Load window remains active for easily loading more files later. Depending on the setting of the Insertion Strategy list box, your file will be either loaded directly on the Program window or in the Resources Media window. If the file is a still image, the project's attributes are not changed and the first frame of the track becomes the image. CINELERRA-GG usually builds an index file if one does not already exist, in order to speed up drawing. You can edit and play the file while the index is being built.

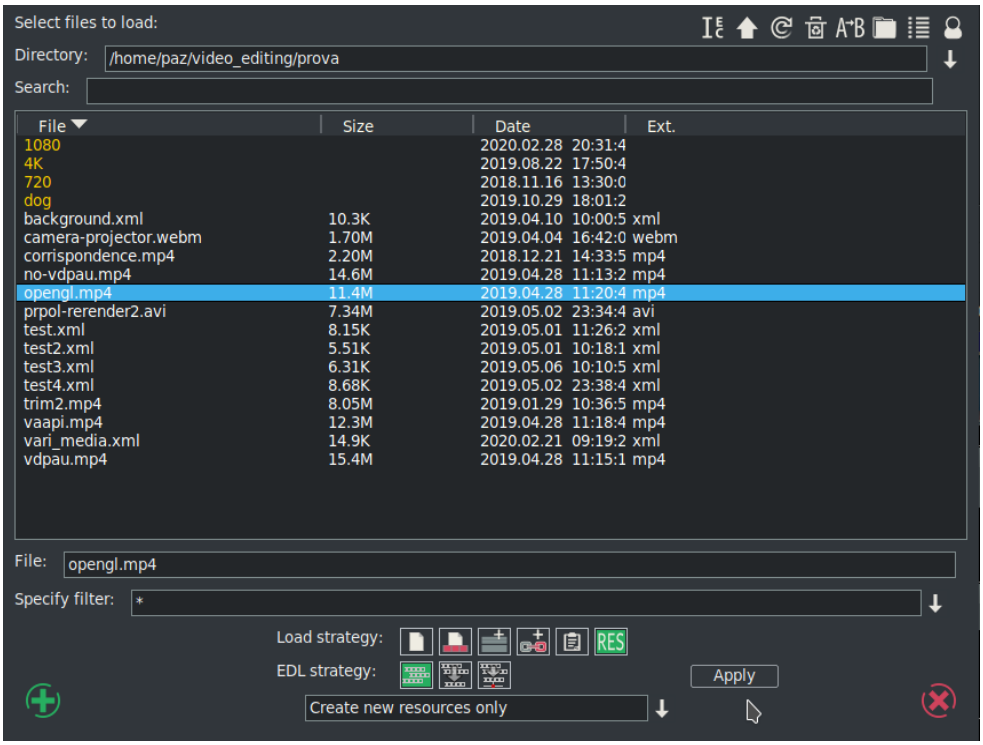


Figure 4.2: *Load file menu. Note the green checkmark for OK and the middle Apply option*

To load a file, you will need to:

1. set a directory path in the Directory input box
2. choose a file or set of files by highlighting your selection
3. select an insertion strategy for loading the file

Once you have completed making your choices and clicking OK or Apply, by default three things occur:

- The existing project is cleared from the screen.
- The project's attributes are changed to match the file's attributes.
- The new file's tracks are created in the timeline.

NOTE: CINELERRA-GG supports paths of up to 255 characters, which is sufficient for most situations. In the case of long folders, subfolders and filenames, be careful not to exceed this limit to avoid errors.

Let's now see in detail the options of loading files.

Insertion Strategy CINELERRA-GG lets you change what happens when you load a file. In the Load dialog window go to the Insertion strategy box and select one of the options in the drop down menu. Each of these options loads the file a different way.

Replace current project: all tracks in the current project are deleted and a set of new tracks are created to match the source file. Project attributes are only changed when loading XML. If multiple files are selected for loading, CINELERRA-GG adds a set of new tracks for each file. New resources are created in the Resources Window, replacing the current ones.

Replace current project and concatenate tracks: same as replace current project, except that if multiple files are selected, CINELERRA-GG will concatenate the tracks of each file, inserting different source files in the same set of tracks, one after another, in alphanumeric order, starting at 0. New resources are created in the Resources Window, replacing the current ones. Files go across the timeline.

For ffmpeg and mpeg files, when the Insertion strategy methodology in the `File → Load files` pulldown is chosen to be either *Replace current project* or *Replace current project and concatenate tracks*, the basic session format parameters are reinitialized to match new media. This selects the default asset and determines its width, height, and video length, frame rate, calculates the colormodel, and assumes square pixels to make an intelligent guess about aspect ratio for video. For audio, the sample rate, audio length, and channel count (mono, stereo, or 5.1) are reinitialized. In addition the *Track Size* will be computed and is reinitialized to match the new loaded media. When using *replace* type insertion strategy, the new asset list is the only media in use so that this update saves the user from immediately needing to change the session format to match the only possibility.

Append in new tracks: the current project is not deleted and new tracks are created for the source, one set of tracks for each file. New resources are created in the Resources Window. Files go down tracks.

Concatenate to existing tracks: the current project is not deleted and new files are concatenated to the existing armed tracks, inserted in the same set of tracks of the current project, one after another, in alphanumeric order, starting at the end of the tracks. If the current project has more tracks than the source, the source file will be inserted in the first set of armed tracks. If no tracks are armed, no files will be inserted. New resources are created in the Resources Window.

Paste at insertion point: the file is pasted into the timeline at the insertion point, on the first set of armed tracks. If multiple files are selected for loading, they will be inserted on the same set of tracks, one after the other. New resources are created in the Resources Window.

Create new resources only: the timeline is unchanged and new resources are created in the Resources Window only. This is default.

Nest sequence: nested assets are added to the timeline by using the Nest sequence insertion strategy. The file will be pasted into the timeline over the current selection or at the insertion point. A nested sequence is media that had already been saved as an EDL earlier. Nesting is described more fully in section 6.6.

The insertion strategy is a recurring option in many of CINELERRA-GG's functions. In each place the options do the same thing. If you load files by passing command line arguments to CINELERRA-GG, the files are loaded with *Replace current project* by default.

Loading Multiple Files In the Load dialog go to the list of files. Selecting files utilizes the motif style selection method.

1. Select a single file by highlighting.

2. Select multiple files by moving to another file and select it while holding down Ctrl. This selects one additional file.
3. Or move to another file and select it while holding down Shift. This selects every intervening file.

This behavior is available in most listboxes. It is an especially useful method when used with *Concatenate to existing tracks* insertion strategy to create an images slideshow or a song playlist.

Loading files from the command prompt Another way to load files is to pass the filenames as arguments on the command line. This starts the program with all the arguments loaded and creates new tracks for every file. For example:

```
{your_cinelerra_program_path} video1.mp4 video2.mp4
```

Finding Files by Extension, Sub-list, or with Search If there are too many files in your media directory, it can be difficult to find the file you want. For this reason, the Load window allows you to filter which files are displayed in the list box by extension name. Click the dropdown box on the right side of the *Specify filter* list box below the file name text box, and select the file extension of your media (for example: mp4, mov, mp3, avi, jpg, etc). The file list now shows only files with the selected extension. Perhaps even easier is to use the Search box on the top underneath the *Select files to load* listbox. Here you can keyin a character or string to look for.
You can also get a sub-list of potential files to choose from. For example, you know that the file you are looking for begins with the capital letter "C". If you keyin "C" into the selection box immediately below the list of files, and then click the left mouse button, a sub-list of files beginning with the "C" shows up under the selection box. Clicking the right mouse button cancels this sub-list.

Loading the backup There is one special XML file on disk at all times.

After every editing operation, CINELERRA-GG saves the current project to a backup in `$HOME/.bcast/backup.xml`. In the event of a crash, the first thing you should do after restarting CINELERRA-GG is select `File` → `Load backup` in order to load the backup. This will start CINELERRA-GG at the point in your editing operations directly before the program crashed. It is important after a crash to restart CINELERRA-GG without performing any editing operations as you will overwrite the backup. Note that the backup.xml file is always a single file which means that when you are working with two instances of CINELERRA-GG open at the same time, they use the same backup file. In this case, the last operation made in whatever instance will overwrite the backup.

4.3.1 Sort within Sort in File Load Dialog

When you use the *File* pulldown to load files, you can do a sort within a sort when you click on the labeled header box (figure 4.3). This is useful, for example, when you want to find the smallest file for a specific extension. In the screenshots below, the first illustrates the default *File* sorted alphabetically; the second shows the *Size* is now sorted; the third shows how after sorting on Size, you sort on Ext. The size sort is maintained within the extension sort so that *c.d* comes before *a.d* in the File header box because the size is smaller.

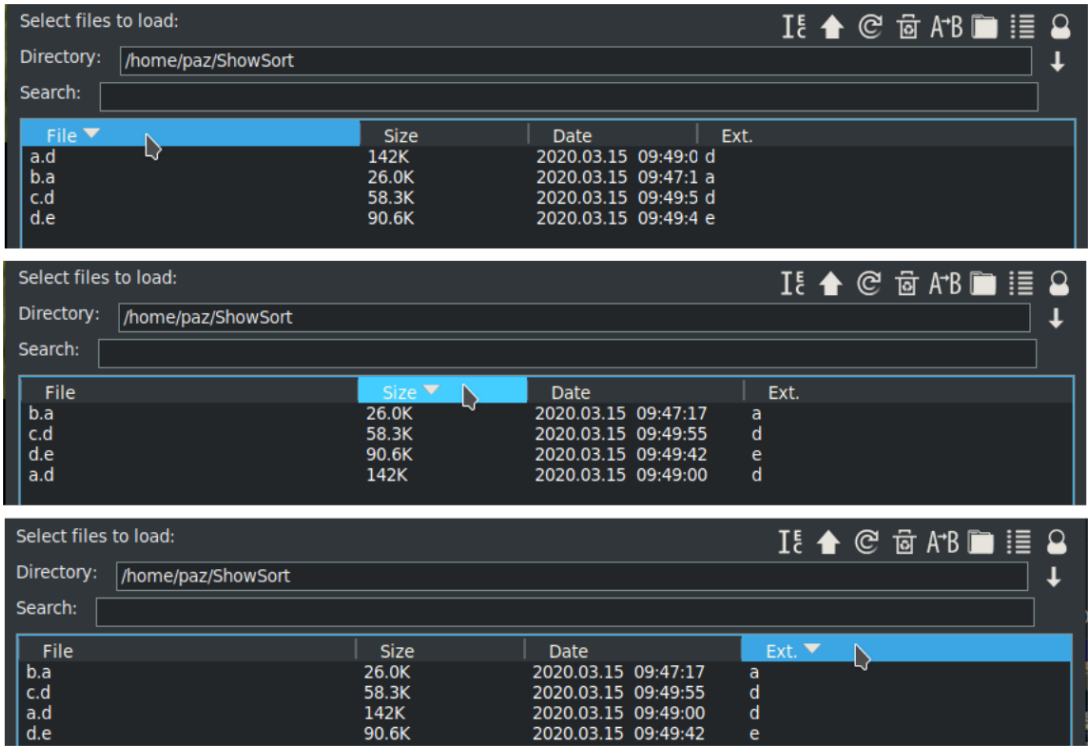


Figure 4.3: *Load - Sort by File name, sort by file Size, and within Extension after a previous Size sort*

4.3.2 Size Numeric Format Displayed in File Load

There are several icon buttons at the top on the right hand side of the Load window. Each has a tooltip to explain what it is for. You can see these in the previous figure. One is for File size format. There are 4 numerical representation variations for reporting the file size in the `File → Load` pulldown. You can see the options in the Load window to the right of the top line that read *Select files to loads* (figure 4.4):

0 this is the default and current behavior and shows bytes the same as the *ls -l* command.

SI 3 significant digits suffixed by lower case k,m,g,t,b for representing magnitudes in 10^3 (1000)

Ic^E 3 significant digits followed by upper case characters for representing magnitudes in 2^{10} (1024)

0, like the exact default byte representation but with comma separators for easy reading. Periods can not be used as separators due to locale conflict with ffmpeg coding.

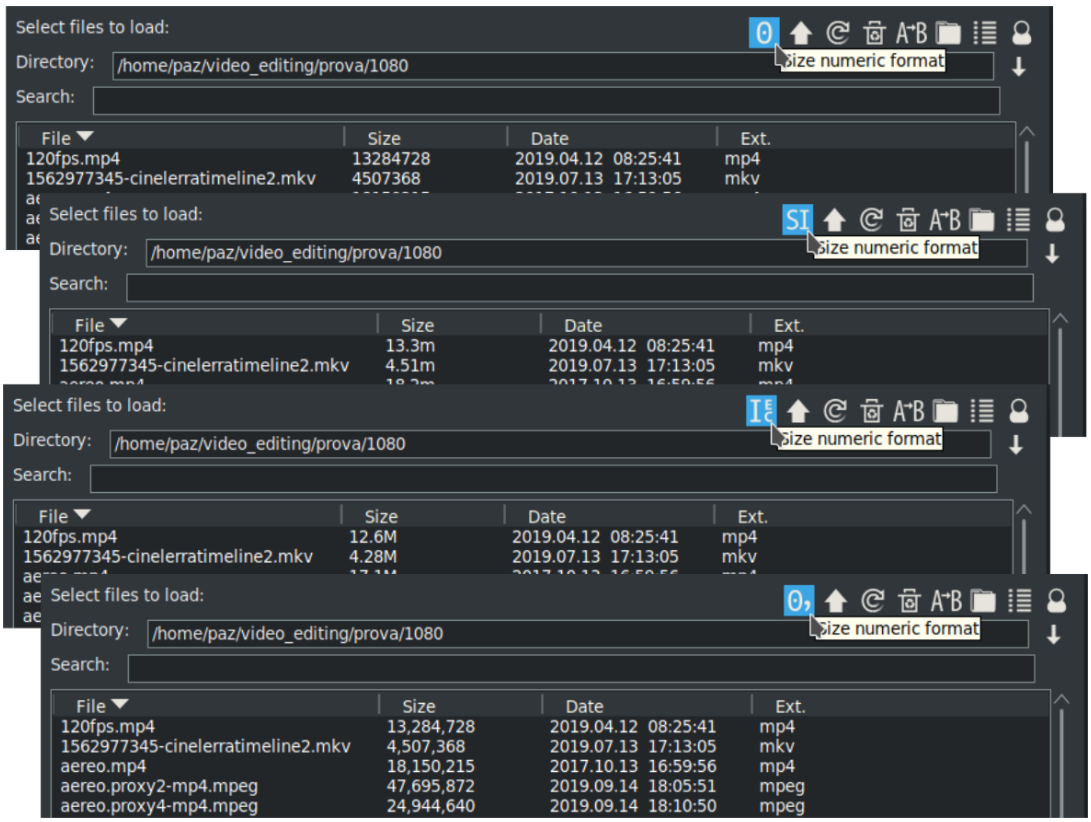


Figure 4.4: *Load windows with various Numeric Sizes*

4.3.3 Probe Order when Loading Media

Why is this mentioned here? So many programs have been written whose functionalities overlap and you may want to ensure that the one you wish to use is actually used. Over time which one matches first may vary. Ffmpeg is so generic that if your setting is *Try ffmpeg first* it will almost certainly get used and it leaves little chance that other methods will even get a chance. Some of the codec file drivers can open a variety of media, and some of the more common methods may have more than one file driver which could be useful to decode your media file, for

example Tiff. For expert specialized usage, when you want to guarantee that a certain method is used, you can change the *probe order*. Use the pulldown `Settings` → `Preferences` to get to the *Interface tab* where you will see a box in the *Operation* section on the left side called *Probe Order*. Click on the box and use the up/down/enabled boxes to change the order of the item you have highlighted (figure 4.5).

Up move the item up 1 (if the item is currently on the top, it will be moved to the bottom)

Down move the item down 1 (similarly, if on the bottom, it will be moved to the top)

Enable there will be a check mark if the item is currently enabled. If you disable it, it will not be used to probe the media to determine what it is. Double left mouse click will toggle the enabled/disabled state of the highlighted item. If both FFMPEG Early/Late are enabled, the FFMPEG code could be run twice if the 1st FFMPEG run failed (but so will the 2nd!).

The default setup is set to duplicate the past expected behavior with the exception that CR2 for Raw Camera mode is disabled. Changes made in the settings will be retained in the `.bcast5` file.

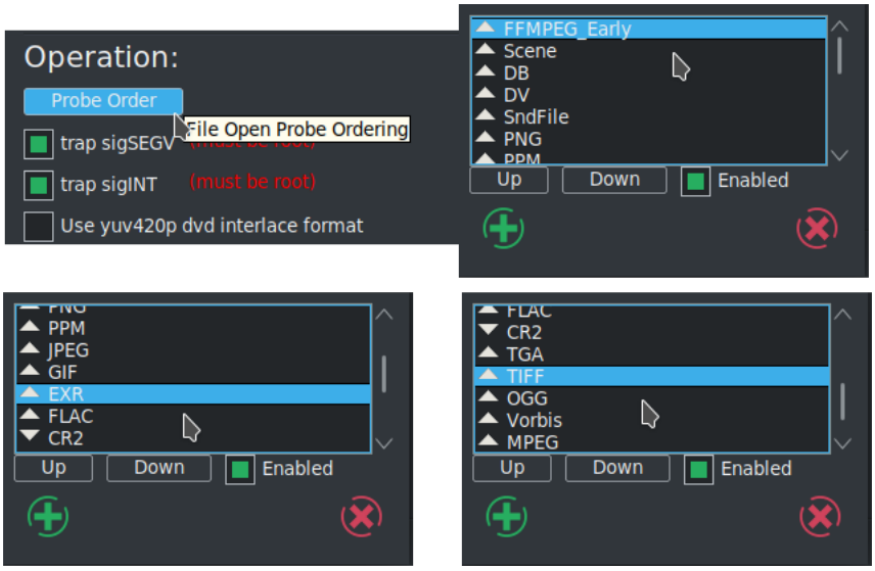


Figure 4.5: *Three example of Probes window*

Figure 4.5 show the first few probe items. Note that the up arrow on the left, signifies *enabled*. Scrolling down shows the next 2 pages of possible drivers for a total of 17.

The order change will not take effect until you click on the checkmark in both the Probes window and the Preferences window. When you click on the FF button, which is in the corner of the main window (figure 9.1) to change *Try FFMpeg first/last*, enabling of FFMPEG_Late or FFMPEG_Early will be toggled automatically in Probes to match that choice but does NOT change its position in the table. Be sure to only click on the FF button without the Preference/Probes window up to avoid unexpected results. It is also recommended to leave FFMPEG_Early/FFMPEG_Late close to the top/bottom positions. There is one case where you may want to disable all of the probes if you want to force PCM – Pulse Code Modulator. This code is always run when all other probes fail.

4.3.4 Program Selection Support after Load

Some kinds of media have *program* streams, like captured mpeg broadcast stream data. For example, you may be able to *tune* to channel 9, but be able to see 9-1, 9-2, and 9-3 on your TV. If you open a capture of this kind of media, all of the channels are present in the timeline. To select and view just one program, you can use Alt-1 to select program 1, or Alt-2 to select program 2, etc. up to Alt-8. This will remove all of the other unrelated tracks and reset the format. This feature can be used even if there is only one program, by pressing Alt-1, and the effect will be to reset the session format to the parameters from the media probe. Note that there may be several audio *programs* associated to a video stream; for example, there may be dialog in another language or some kind of descriptive dialog. Since the first associated audio is always selected, this may not produce the intended results.

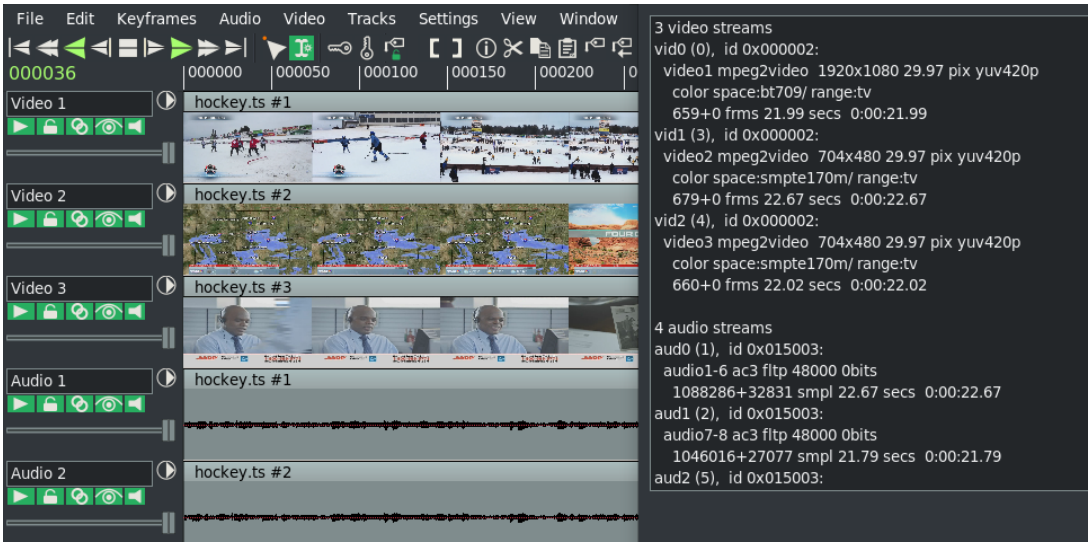


Figure 4.6: *Multiple program streams and Asset Detail*

Below are screenshots illustrating multiple program streams (figure 4.6). The left

screenshot is a partial main CINELERRA-GG window showing a pre-recorded broadcast TV media/audio stream with 2 programs plus several associated audio tracks. The second screenshot of *Asset Detail* provides detailed information on each of the streams obtained through executing the `Info` → `Details` as explained in the section Info Asset Details.

4.4 Saving Your Work

You can save your work as a project, which is what is loaded in CINELERRA-GG now, or as an export, which is all the media it takes to reproduce your project space.

4.4.1 Saving Project Files

Saving XML files is useful to save the current state of CINELERRA-GG before quitting an editing session. CINELERRA-GG saves projects as XML files. There are a few options you can use to save your work via the *File* pulldown menu: *Save*, *Save as...*, *Export project*, *Save backup*. You can either overwrite an existing file or enter a new filename. CINELERRA-GG automatically concatenates `.xml` to the filename if no `.xml` extension is given.

When CINELERRA-GG saves a file, it saves the EDL of the current project but does not save any media, instead just pointers to the original media files. For each media file, the XML file stores either an absolute path or just the relative path . If the media is in the same directory as the XML file, a relative path is saved. If it is in a different directory, an absolute path is saved.

You have to be careful when moving files around to avoid breaking the media linkages. You can keep the media and the XML file in the same directory forever and freely move the whole directory, since relative paths are saved. Alternatively you can save the XML file in a different directory than the media but then you can't move the media. In this case you can freely move your XML file around, since absolute paths are saved. If you saved your XML file in the same directory as your media but you would like to move location, you can change the paths from relative to absolute by going to `File` → `Save as...` and entering the new location. Similarly if you saved your project outside your media directory but you would like to move your media to another location, you can change the paths from absolute to relative by going to `File` → `Save as...` and saving your XML file in the same directory as the media.

You can also repair broken media linkage by editing the XML file in a text editor. For every media you moved, search for the old path and replace it with the new one. You should make a backup copy of your XML file before editing. You can also replace the path of every asset whose source file you moved also within the

program, by entering the new location in the Asset info window. To open this window, right click on the asset in the Resources window and choose Info... in the popup menu. Directly type the path in the first field of the dialog or click on the magnifier on the right to browse your files. Operating from the GUI is convenient only when a very small number of changes is needed.

Real-time effects in an XML file have to be re-created every time you play it back. The XML file also contains copies of all the source assets on disk, which takes up space. Render your projects to a final format for more persistent storage of the output.

4.4.2 Export Project – Save or Moving Project to another Computer

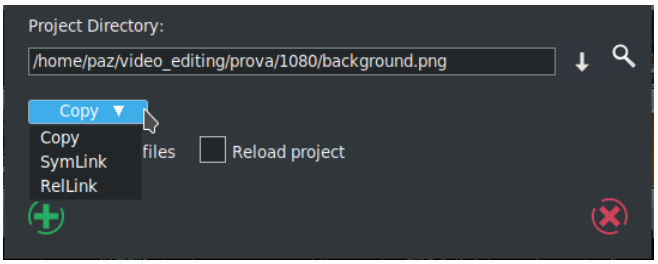


Figure 4.7: *Export Project option popup and the 3 available options.*

A *File* pulldown called *Export Project...* is also available (figure 4.7). Although, it can be used in the same manner as the other *save* options, it is very useful when it is necessary to move a project to another computer that may have a different top level directory structure or if you want to include subdirectories to better organize your files.

Originally, the easiest way to maintain a project for moving to another computer, was to put all of the files in a single directory with no subdirectories along with the EDL saved `.xml` file. This is commonly called a *flat* file structure. So if the media was in the same directory as the XML file, a relative path was saved. If it was in a different directory, an absolute path was saved.

Definition of Fields:

Project Directory name of the directory where you want the xml file to be saved. It will only create a subdirectory in 1 level of the defined directory. Available option types for saving a project:

Copy all files are copied to the project directory and the xml is saved; same as original *flat*. Option is very useful to ensure all files needed to illustrate a problem are available for analysis.

SymLink symbolic links are created for absolute paths of media in their current location.

RelLink symbolic links are created for relative paths of media in their current location. This option allows for using relative paths without the requirement to maintain a *flat* file structure and makes it easy to move to another computer.

Overwrite files when checked, if any files with the same name currently exist in the directory, they will be overwritten. In any case, the named XML file will always be overwritten.

Reload project when checked, after the save option the new saved project will be loaded. Default is not to do so.

Keep in mind that to maintain the integrity of your project xml file for easy moving to another computer, do not delete the symbolic links. You will want to use `cp -a` to maintain the links for moving to a USB key or another computer.

4.4.3 Export to EDL

CMX3600 is an outdated and limited format for EDL. For example it only goes up to 999 edits per project, supports only 4 audio tracks per video track and supports only Cuts, Fades and Keys. But it has the advantage of being a standard format supported by many NLEs. CINELERRA-GG uses a rich XML format that is not compatible with other NLEs. You can use the *Export to EDL* option to try to produce EDL files of type CMX3600 in *strict mode* and thus be able to bring them into other NLEs. Given the limitation of this format, however, it is difficult to get a working EDL file that exactly reproduces the original xml. This feature also produces files that each contain a video track or an audio track and does not embed video with audio or audio with audio. Manual intervention with a plain text editor on the produced EDL files may be necessary. For example, if the program in which you want to open the EDL reports the message of not being able to find the source files, then we can manually add the path to each line that reports the source file.

4.4.4 Backup and Perpetual Session

Default behavior:
In an effort to minimize loss of work due to user, hardware, or software issues, CINELERRA-GG has some automatic backup capabilities. CINELERRA-GG automatically saves every editing operation to the current project on disk continuously to

a file named `$HOME/.bcast5/backup.xml`. We can also make a manual backup via `File` → `Save backup` ("b") or we can load the chosen backup via `File` → `Load backup`. Before a new backup is rewritten to `backup.xml` by an editing operation, the content is automatically copied to `backup.prev` which becomes a second backup file. In total we have 2 levels of backups: the previous one, called `backup.prev`, and the current one, called `backup.xml`.

In the un-likely event of a crash, when you restart CINELERRA-GG, you should select `File` → `Load backup` in order to continue with the operations that were recorded before the crash. If you have more than 1 instance of CINELERRA-GG running, only the last editing operation made in whichever instance it was last made, will overwrite the `backup.xml`. If for some reason you forgot to use *Load backup* immediately when restarting or you did a Load with *Replace current project* in your current session (not default behavior!), you have a second chance to use `File` → `Load backup` and select `$HOME/.bcast5/backup.prev` as long as you only loaded a different file and have performed no editing operations. This same file is also used by multiple instances of CINELERRA-GG.

Use of *Autosaves continuous backups:*[1]
Autosaves continuous backups can be enabled in (see Dangerous section):
`Settings` → `Preferences` → `Appearance Tab` → `Dangerous`
In this modality the formation of `backup.xml` and `backup.prev` remains the same. In addition there is a simultaneous formation of a third copy with a timestamp, called `backup_prev.xxxxxxxx.xxxx` (with *Date* and *Clock time*). So at each editing operation a backup will be created with its own timestamp. The user can thus go back to the moment when he wants to restart an editing, because of an error, or to experiment with different choices from a certain moment on.

We can load the backup file of our choice via `File` → `Load backup`. For long editing sessions it is easy to have lots of backups but when you exit from CINELERRA-GG, all files with timestamps will be deleted except the last 50, which will still be available at the next restart.

Perpetual session:
is very useful for working on a project over many days so you can just quit before shutting down and the next time you start up CINELERRA-GG you will be right back where you left off. You will retain all of your undo's and redo's. The binary file name is `$HOME/.bcast5/perpetual.dat` and as long as `Settings` → `Preferences`, the *Appearance* tab has the Flag *Perpetual session* set this capability takes effect. It is very important to understand that this is not the same as the continuously editing- operation-updated `backup.xml` file which is still operating as usual. The `perpetual.dat` file is updated so that when you Quit CINELERRA-GG

[1] credits to Andrew Randrianasulu and Reuss Andràs

in the normal manner, interrupt the program, kill the program, or there is a segv or system crash, the `perpetual.dat` file still reflects the state of your project. This means the editing/undo's/redo's you executed during the current session should still be available on a restart with the exception of any edit that may not have had a chance to be written on disk before an abnormal stoppage.

Some notes to keep in mind about Perpetual session are:

- when you Quit in the normal manner, it does not have to ask whether or not to save a backup
- takes disk space in `.bcast5` area and this could get really big
- after you complete a project, it is advisable to turn off the Perpetual session flag before quitting so that when you start a new project, you can start with a fresh `perpetual.dat` by turning the flag on or after stopping CINELERRA-GG, delete the current `$HOME/.bcast5/perpetual.dat` file
- only session data is backed up (not program feature setup)
- the files `backup.xml` and `backup.prev` will operate the same as before so that if there is a crash, you can still use `File` → `Load backup` in order to continue where you were interrupted.
- to start CINELERRA-GG without using your Perpetual session data even if enabled, keyin: `/your_cinelerra_path/bin/cin -S`

For developers:
We report the list of editing events that lead to the automatic formation of a backup file:

- cinelerra/cwindowgui.C
- cinelerra/swindow.C
- cinelerra/keyframepopup.C
- cinelerra/pluginpopup.C
- cinelerra/mwindowedit.C
- cinelerra/presetsgui.C.sav1
- cinelerra/assetpopup.C

- cinelerra/cwindowtool.C
- cinelerra/render.C
- cinelerra/setformat.C
- cinelerra/mwindow.C
- cinelerra/keyframegui.C
- cinelerra/mwindowgui.C
- cinelerra/mainmenu.C
- cinelerra/loadfile.C
- cinelerra/savefile.C
- cinelerra/menueffects.C
- cinelerra/main.C
- cinelerra/presetsgui.C.sav
- cinelerra/record.C
- cinelerra/plugindialog.C

Chapter 5

Editing

Editing comprises both time and track space. The timeline consists of the time certain media appear on the track going left to right and a set of tracks from the top to the bottom. There are 2 methods of timeline editing – drag and drop editing , also called *arrow mode* , and cut and paste editing or *I-beam mode* . Cut and Paste is the default editing mode. An additional, but not often considered editing method is called *two-screen editing* (or 3 point editing) where the Viewer is used to view media and then the desired clip from the media is transferred to the timeline.

In the timeline, Audio tracks are different from Video tracks so a media including both audio and video will be split into 2 or more independent tracks. There is no *Link/Unlink*-like function present in other programs, although there are workarounds.

The timeline is where all editing decisions are made (figure 5.1). This is a stack of tracks in the center of the main window. It can be scrolled up, down, left and right with the scrollbars on the right and bottom. It can also be scrolled up and down with a mouse wheel, or left and right while holding down the Ctrl key and using the mouse wheel.

Figure 5.1: *Timeline editing session*

The active region is the range of time which is affected by editing commands on the timeline. The active region is determined first by the presence of in/out points on the timeline. If those do not exist the highlighted region is used. To reiterate,

highlighting is done in *cut and paste mode* by moving the insertion point with the mouse in the timeline to where you want to start. Then hold down the LMB, drag the mouse to where you want the end point to be and release the LMB. In *drag and drop mode*, the method to create a highlighted selection is to hold down the Ctrl key and double click with the LMB with the mouse over that column.

If no highlighted region exists, the insertion point is used as the start of the active region. Some commands treat all the space to the right of the insertion point as active while others treat the active length as 0 (zero) if no end point for the active region is defined.

Most importantly, editing decisions never affect source material meaning that it is non-destructive editing. So not only does your original media stay completely untouched, it is much faster than if you had to copy all the media affected by an edit. Editing only affects pointers to source material, so if you want to have a new modified media file at the end of your editing session which represents the editing decisions, you need to render it. Saving and loading your edit decisions is explained in the Load, Save and the EDL section and rendering is explained in the section on Rendering.

Nomenclature: media loaded in the timeline, in whole or in part, are called *edits*, unlike other programs that call them *clips*. Clips in CINELERRA-GG are those that are created in the Viewer window or in the timeline via the `to clip` command and brought into the Resources window, inside the `Clips folder`, where they can be renamed and a description added. These, once brought into the timeline, are renamed edits. In CINELERRA-GG the difference edits/clips is not important and you can use them as synonyms; however, the difference of clips as it is intended in other NLEs is important. These are media (or parts of it) complete in themselves and indipendent from other clips and the timeline. Edits in CINELERRA-GG on the other hand, can be a media (or part of it) but can also be any region of the timeline between In/Out Points or from a highlighted, on which we can do editing operations. This leads to some advantages (e.g. putting an effect only in a part of the active region) but one must always keep in mind that the edit remains dependent on the track and the entire timeline.

In the following editing sections, references to common operations are scattered within any of the modes where they seem pertinent. However, many of the editing operations work in different modes.

5.1 The Patchbay

On the left of the timeline is a region known as the patchbay. The patchbay enables features specific to each track as described next.

Textbox for naming the track. The default names will usually be Video #, Audio #, or Mixer # if using the multi-camera/mixer operations. A # will be designated for subsequent tracks as in 1, 2, 3 and so on.

Expander which is a down arrow on the right side, is for viewing more options on the patchbay and for viewing the effects represented on the track. You can just click on the expander to expand or collapse the patchbay and the track. If it is pointing sideways, the track is collapsed. If it is pointing down, the track is expanded. Existing effects appear below the media for the track.

Below the textbox name are several toggles referred to as *attributes* for different features (currently there are 7 as shown in figure 5.2). If the toggle button is shadowed by a color, the feature is enabled. If the toggle is the background color of most of the window, it is disabled. Click on the toggle to enable/disable the feature.

Several mouse operations speed up the configuration of several tracks at a time. Click on an attribute and drag the cursor across adjacent tracks to copy the same attribute to those tracks. Hold down Shift while clicking a track's attribute to enable the attribute in the current track and toggle the attribute in all the other tracks. Or you can:

1. hold down Shift while clicking an attribute,
2. click until all the tracks except the selected one are disabled,
3. then drag the cursor over the adjacent track to enable the attribute in the adjacent track.

Figure 5.2: *Patchbay*

The 7 *attributes* are described here next followed by the other available feature icons and their description.

Play Track determines whether the track is rendered or not. If it is off, the track is not rendered. For example if you turn it off in all the video tracks, the rendered media file will have only audio tracks. If the track is chained to any other tracks by a shared track effect, the other tracks perform all the effects in this shared track, regardless of play status of the shared track that in this particular case affects the media output but not fade and effects.

Arm Track determines whether the track is armed or not. Only the armed tracks are affected by editing operations. Make sure you have enough armed destination tracks when you paste or splice material or some tracks in the material will get left out. In addition to restricting editing operations, the armed tracks in combination with the active region determine where material is inserted when loading files. If the files are loaded with one of the insertion strategies which do not delete the existing project, the armed tracks will be used as destination tracks. Note that disarming a track does not prevent you from dragging or attaching an Effect/Plugin onto a disarmed track - this is not considered an edit in this case.

Draw Media determines if picons or waveforms are drawn on the asset in the track. You may want to disable this if you know that the media/format takes a long time to draw on the timeline. By default it is set to on in order to see picons on the timeline.

Don't send to output – more commonly called *mute* – causes the output to be thrown away once the track is completely rendered. This happens whether or not *Play track* is on. For example if you mute all the video tracks, the rendered media file will have a blank video track. Mute track is represented on the timeline with a line that has the default color of a pinkish-orange. Use the pulldown `View` → `Mute` to have the line displayed. It is a keyframable attribute, but Mute track keyframing is a toggle and it has only the two values of on or off. If a track is part of a shared track effect, the output of the track with the shared track effect is overlaid on the final output even though it is routed back to another track (the shared track). Mute track is used to keep the track with the shared track effect from overlapping the output of the source track (the shared track) where the shared track effect is not present.

Gang Fader cause the fader to track the movement of whatever other fader you are adjusting by dragging either the fader or the curve on the track. It doesn't affect the editing made with menu controls. A fader is only ganged if the arm track is also on. This is often used to adjust audio levels on all the tracks simultaneously. Gang also causes Nudge parameters to synchronize across all the ganged tracks.

Master Track Mark a track as *master* serves when using *Gang Channels* or *Gang Media* mode. See Displaying tracks: Ganged mode

Track Data Height this up/down toggle symbol to the immediate right of the 5 attributes, is used to individually resize each track. This makes it very easy to temporarily expand or contract the size of that track either by clickin with the left mouse button or using the middle wheel up/down.

Fader slider fade values are represented on the timeline with a pink (default color) curve that is keyframable. All tracks have a fader, but the units of each fader depend on whether it is audio or video. Audio fade values are in dB. They represent relative levels, where 0 is the unaltered original sound level, -40 is silence, -80 the minimum value set by default. You can move fader and keyframes down to -80 but the parameter's curve won't go below -40. For your convenience you can set a different fade range with the curve zoom. Audio fader's main purpose is to *fade out* sound or to lower the sound level smoothly to silence, or *fade in* to make sounds appear gradually instead of suddenly. Video fade values are the percentage of opacity of the image in normal overlay mode, the percentage of the layer that is mixed into the render pipeline in the other overlay modes. Click and drag the fader to fade the track in and out. If it is ganged to other tracks of the same media type, with the arm option enabled, the other faders should follow. Hold down the Shift key and drag a fader to center it on the original source value (0 for audio, 100 for video).

Mixer in the expanded patchbay for that track designates the multi-camera mixer mode.

Overlay mode in the expanded patchbay is used for porter-duff operations and is full explained in Overlays chapter.

Nudge is in the expanded patchbay. The nudge value is the amount the track is shifted left or right during playback. The track is not displayed shifted on the timeline, but it is shifted when it is played back. This is useful for synchronizing audio with video, creating fake stereo, or compensating for an effect which shifts time, all without altering any edits (figure 5.3).

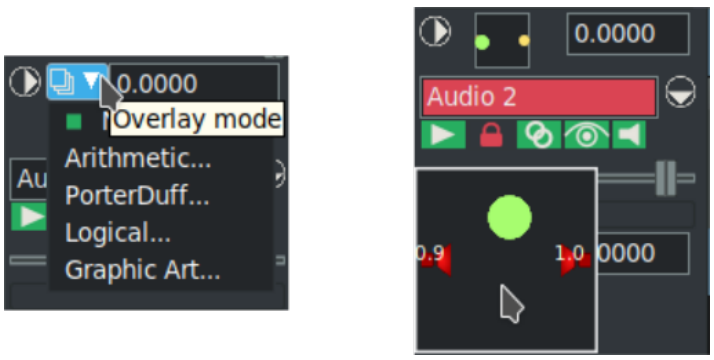


Figure 5.3: *Video Overlay, audio Pan and Nudge.*

Enter the amount of time to shift to instantly shift the track. Negative numbers make the track play later. Positive numbers make the track play sooner. The nudge units are either seconds or the native units for the track (frames

or samples). Select the units by right clicking on the nudge textbox and using the context sensitive menu. Nudge settings are ganged with the Gang faders toggle and the Arm track toggle. Use the mouse wheel over the nudge textbox to increment and decrement the value.

Pan is available in the expanded patchbay for audio tracks via a panning box. Position the pointer in the panning box and click/drag to reposition the audio output among the speaker arrangement. The loudness of each speaker is printed on the relative icon during the dragging operation. The panning box uses a special algorithm to try to allow audio to be focused through one speaker or branched between the nearest speakers when more than 2 speakers are used.

Press the Tab key while the cursor is anywhere over a track to toggle the track arming status. Press Shift-Tab while the cursor is over a track to toggle the arming status of every other track.

Automatic audio mappings Several convenience functions are provided for automatically setting the panning to several common standards. They are listed in the Audio menu. These functions only affect armed audio tracks. They are:

Audio → Map 1:1 This maps every track to its own channel and wraps around when all the channels are allocated. It is most useful for making 2 tracks with 2 channels map to stereo and for making 6 tracks with 6 channels map to a 6 channel sound card.

Audio → Map 5.1:2 This maps 6 tracks to 2 channels. The project should have 2 channels when using this function. Go to `Settings → Format` to set the output channels to 2. This is most useful for down-mixing 5.1 audio to stereo (for more information refer to Configuration, Settings and Preferences section 16.2.1).

Standard audio mappings Although CINELERRA-GG lets you map any audio track to any speaker, there are standard mappings you should use to ensure the media can be played back elsewhere. Also, most audio encoders require the audio tracks to be mapped to standard speaker numbers or they will not work.

In the channel position widget, the channels are numbered to correspond to the output tracks they are rendered to. For stereo, the source of channel 1 needs to be the left track and the source of channel 2 needs to be the right track. For 5.1 surround sound, the sources of the 6 channels need to be in the order of center, front left, front right, back left, back right, low frequency effects. If the right tracks

are not mapped to the right speakers, most audio encoders will not encode the right information if they encode anything at all. The low frequency effects track specifically can not store high frequencies in most cases.

5.2 Manipulating Tracks

Tracks in CINELERRA-GG either contain audio or video. There is no special designation for tracks other than the type of media they contain. When you create a new project, it contains three default tracks: one video track and two audio tracks. You can still add and delete tracks from the menus. The Tracks menu contains a number of options for dealing with multiple tracks simultaneously. Each track itself has a popup menu which affects one track.

Operations in the **Tracks pulldown** affect only tracks which are armed.

Move tracks up | Move tracks down shift all the armed tracks up or down the stack.

Roll tracks up | Roll tracks down wheel motion of the tracks up or down.

Delete tracks deletes the armed tracks.

Delete first track deletes the first track, whether it is armed or not.

Delete last track deletes the last track, whether it is armed or not.

Concatenate tracks operation copies all the assets of every disarmed but playable track and concatenates it by pasting those assets at the end of the first set of armed tracks. They are pasted one after the other, keeping the same order they have on the stack. If there are two armed tracks followed by two disarmed tracks, the concatenate operation copies the assets of the two disarmed tracks and pastes them after the assets of the two armed tracks. If there are three disarmed tracks instead, the assets of two tracks are pasted after the assets of the armed tracks and the assets of the third track are pasted at the end of the first armed track. The destination track wraps around until all the disarmed tracks are concatenated. Disarmed tracks that are not playable are not concatenated.

Align timecodes see Align Timecodes

Append to project allows for creating new tracks after any existing tracks.

Add subttl will add a track for subtitles at the top of the other tracks.

The **Audio** and **Video pulldowns** each contain an option to add a track of their specific type. In the case of audio, the new track is put on the bottom of the timeline and the output channel of the audio track is incremented by one. In the case of video, the new track is put on the top of the timeline. This way, video has a natural compositing order. New video tracks are overlaid on top of old tracks.

5.2.1 Displaying tracks: Ganged mode

Often users working on media where Audio is the main focus, want all of a media's audio channels, whether stereo or 5:1 channels, to be treated as a single unit. They are more familiar with working with a DAW (Digital Audio Workstation software) and find that it takes extra work and a lot more care to have to individually manage audio tracks rather than have them automatically edited as a ganged group.

To get this capability, there is a *Gang Tracks* toggle button on the main timeline controls to switch between 3 modes of working with multiple channels so that the tracks are automatically edited as a single unit. Operations affected include edits such as cuts, moving sections, and adding plugins. In addition to letting you work on the tracks of a group as one, you also get a saving of space on the timeline that makes it easier to view and edit the tracks. Group masters are marked by the *Master Track* toggle in the patchbays . A track gang group begins on a master track, and extends to, but not including, the next master track. The 3 *Gang Tracks* modes are:

1. **Gang None:** this is the default mode and is the traditional way Cinelerra operates. If you never toggle the "Gang Tracks" button, everything operates normally as it always has. In this mode, the button icon looks like 3 tracks with ... on the end and all tracks are visible on the timeline. See figure 5.4

Figure 5.4: *Gang None: only the video track is master. We see all tracks both master and non-master (default)*

2. **Gang Channels:** in this mode, all *Master Tracks* are shown. The exception is when there is no Master video, no Master audio or no Master subtitle track, in which case for each the first track of video, audio, and subtitle will be shown. For users who routinely switch the track order frequently, the patchbay has the *Master Track* icon to set your own preference. Stereo tracks, or 5:1 channels/any number of audio tracks, are drawn as 1 audio track for the purpose of making changes on that single track which are propagated to all of its other channel tracks automatically. This is the DAW-like mode. The *Gang Tracks* button icon looks like 2 tracks and only the first video and first audio tracks will be shown. See figure 5.5

Figure 5.5: *Gang Channels: only the video track is master. Let's see the master tracks and the first of the non master tracks*

3. **Gang Media:** this mode only shows the *Master Tracks* of the media but changes are propagated to the appropriate other related tracks (even if they are not visible). The *Gang Tracks* button icon looks like a single track. See figure 5.6

Figure 5.6: *Gang Media: only the video track is master. We only see the master tracks*

The current *Gang Tracks* mode is saved across sessions and will be saved in your project. Although most likely users will work in the *Gang None* mode, users more familiar with DAW software will tend to use the *Gang Channel*s mode.

More details are described next about master tracks. Each patchbay has a *Master Track* button on the right of the patchbay. These mark which tracks of the media are master tracks. The span of tracks between the gang group masters are the gang track group. The master track toggles can be disabled/enabled as the user wishes to create the desired track groups. Normal file loads will mark the first stream of each file loaded as a master, if the media is loaded with insertion strategy of *Replace current project*, *Replace current project and concatenate tracks* or *Append in new tracks*. Frequently, editing is done on video, audio, or audio/video groups to maintain timeline synchronization. By grouping related tracks, this procedure is much more automatic.

Previously existing projects created before the addition of ganged tracks, will have ALL tracks marked as master tracks and changing the *Gang Tracks* button mode will have no effect. The user will have to properly designate which track is the master track and disabling the others in order to make use of the toggle modes.

There are no restrictions on how you use this feature and there may be variations for users not concerned about DAW-like workings that are very useful. Several methodologies are possible, but you have to know what you are doing.

NOTES:

- When in *Gang Channels* or *Gang Media* mode, if the first audio track is not disarmed, but any of its connected channel tracks are, the disarm of those channels are ignored and all channels are treated as being armed. This is to be consistent with the purpose of using this mode; the purpose being that any edits to the first master track are automatically propagated to all other channels. *Warning*: this behavior is contrary to the default of CINELERRA-GG whereby an unarmed track never undergoes any alteration. This exception was made to mimic the behavior of DAWs, where the various channels of an audio track are combined into a single "line", and it makes no sense to talk about armed or unarmed channels.
- It may be better when using the *Gang Channels* or *Gang Media* mode to make any changes to the first audio channel to be automatically duplicated before reverting to the *Gang None* mode to make other changes. Once you start changing individual channels (that is, audio tracks) so that they are no longer coordinated in their edits/plugins, it may be more difficult to switch

back to *Gang Channels* mode and get the desired results since the channels may no longer be lined up.

- Gang modes should not be used with Multi-camera/Mixers because they have very specific operational capabilities which are not compatible with the way that Ganging works.
- Gang flags in the patchbay specifically when in *Gang Channels* or *Gang Media* modes will operate as set in the Master Track. For example, if when in *Gang Channels* mode, you turn off “Play track” on the visible audio track, none of the associated channels (i.e. the other stereo or other 5 channels) will play. And if in *Gang Media* mode, if you turn on “Play track” for the Master Track, all associated video and audio tracks will play unless you turned off “Play track” for some of the associated audio/video slave tracks which will not play.
- When adding a Plugin/Effect in *Gang Channels* or *Gang Media* mode to the track, highlighted selected region, or In/Out pointers area via dragging the plugin from the Resources Window the plugin will be added on the master and all slave tracks. If using the Audio or Video pulldown, *Attach Effect* option and you want to propagate the effect to all slave tracks, make sure that the checkbox for "Attach single standalone and share others" in the Dialog window is checked. However, currently when using the right mouse button (RMB) on the master track to *Attach effect*, the effect will only be inserted on that track.

5.3 Two Screen Editing

This is a fast way to construct a program out of movie files (in other programs is called *three points editing*). The idea consists of viewing a movie file in one window and viewing the program in another window. Subsections of the movie file are defined in the viewer window and transferred to the end of the program in the program window. Two screen editing can be done simply by using keyboard shortcuts. To get familiar with which keys to use, move the mouse pointer over the transport panel and a tooltip appears, showing what key is bound to that button.

To begin a two screen editing session, load your media resources by using the main menu **File pulldown** and choose *Load files* ; make sure the insertion mode is set to *Create new resources only*. This insertion strategy is to ensure that the timeline stays unchanged while new resources are brought in. Go to the Resources window and select the Media folder. The newly loaded resources will appear. Double click on a resource or drag it from the media side of the window over to the Viewer window .

Check to make sure there are enough armed tracks on the timeline to put the subsections of source material that you want. Usually this would be one video track and two audio tracks, but if there are not enough, just create new tracks or arm more tracks.

Now to start your 2 screen editing, in the viewer window, define a clip from the media file:

1. Set the starting point with the In pointer button. You will see a left hand bracket on the timebar.
2. Move your cursor to the ending point of the clip you want to use.
3. Set the ending point with the Out pointer right hand bracket.
4. You will see a colored bar inside the brackets for easier viewing.
5. Drag the In/Out point with the mouse to conveniently change their position.

These In/Out points define a clip . You can now use this in a couple of different ways.

Splice The splice icon, or shortcut letter “v”, inserts the selected area in the timeline after the insertion point. After the splice has taken effect, the insertion point moves to the end of the edit ready to be used as the next splice location. This way you can continuously build up the program by splicing. If an In point or an Out point exists on the timeline the clip is inserted after the In point or after the Out point. If both In and Out points are set on the timeline, the clip is inserted after the In point. If there are edits after your chosen splice location on the timeline, they will be moved to the right.

Overwrite The overwrite icon, or shortcut letter “b”, overwrites the region of the timeline after the insertion point with the clip. If an In point or an Out point exists on the timeline the clip is overwritten after the In point or after the Out point. If both In and Out points are set on the timeline, the clip is inserted after the In point. If a region is highlighted or both In and Out points exist they limit the region of the overwriting and the clip may therefore be shortened. Here is a detailed explanation to take advantage of this method.

To overwrite exactly on a precise region of the timeline:

1. Arm only tracks to change.

2. Define the destination region on the timeline with [and], the In and Out points.
3. You can achieve maximum precision by setting the active region in the zoom panel.
4. Define the clip you want to use in the viewer with [and], the In and Out points.
5. Overwrite from Viewer to the timeline.

If the destination region is shorter than the clip defined in the viewer, the portion of the clip longer than the destination region won't be inserted and on the timeline the following edits won't move. If the destination region is longer than the clip defined in the viewer, the destination region will shrink and on the timeline the following edits will move to the left.

Clip The clip icon, or shortcut letter "`i`", generates a new clip for the resource window containing the affected region but does not change the timeline. Every clip has an optional/default title and description.

Copy The copy icon, or shortcut letter "`c`", copies the selection into the copy buffer.

5.3.1 Use Case – Working with Sequences

From the Viewer to the Timeline with the sequences imported in a Master Project.

A convenient methodology for working on a Master project along with 1 or more previously saved Sub projects or *sequences* use case is described here. A sequence is an edited assembly of audio and video clips generally consisting of a series of videos that relate to the same activity. This use case explains how to work this way and some things you need to be aware of.

1. First load your Master project, which you worked on and saved earlier as an `.xml` file, using an Insertion strategy of *Replace current project*. Generally this Master project consists of media with any of the attributes of clips, autos, possibly keyframes, and effects. You will see your project on the main timeline and the Media files that are part of this Master project will be displayed in the Resources window in the Media folder.

2. Previously you may have also saved a Sub project, which will now be referred to as a Sequence, as an `.xml` file that may contain any of the same such things: media, clips, autos, keyframes, effects. Second you will want to load the Sequence using an Insertion strategy of *Create new resources only*. When

you do the load, this Sequence will show as a file in the Resources window in the Clips folder. The actual media will show in the Media folder.

3. Now Drag and Drop the Sub project from the Clips folder to the Viewer.
4. Set In and Out Pointers in the Viewer to the region of interest in the Sub project and in the Timeline of the Main window of your Master project, move the cursor position to where you would like to insert this In/Out section.
5. Click on the *Splice (v)* button in the Viewer to insert this section into the Master project timeline. All of the attributes of the selected Sub project section will now be inserted in the main timeline to include the autos, keyframes, effects, and labels.
6. Alternatively, if you click on the *Overwrite (b)* button in the Viewer, you can see the Sub project In/Out section in the timeline, but without its autos, effects, keyframes, etc. If in the timeline there were some autos, effects, and keyframes in that Master project, they will be in effect for the new section.

You can see the advantages of using Splice versus Overwrite to either insert (splice) with all of the attributes of a specific section of your Sequence or to overwrite without the attributes to allow for the smooth operation on the timeline by retaining the timeline's attributes at that point.

NOTE: for correct operation of this use case, you should have the same (or more) number of tracks in the Master project as you do in the Sequence. To avoid having to know how many tracks you need, you can use the Nest feature as described in the Nesting section (6.6).

5.4 Cut and Paste Editing

This is the more traditional method of editing in CINELERRA-GG and therefore is the default. To enable the cut and paste editing mode on the timeline, select the I-beam toggle on the control bar at the top of the main program window. You can copy edits in the same track, copy from different tracks in the same instance, start a second instance of CINELERRA-GG and copy from one instance to the other or load a media file into the Viewer and copy from there.

To start editing, load some files onto the timeline. Select a region of the timeline by click dragging on it and select the cut button to cut it. Move the insertion point to another point in the timeline and select the paste button. Assuming no In/Out points are defined on the timeline this performs a cut and paste operation.

Most editing operations are listed in the *Edit* pulldown . Some of them have a button on the program control toolbar as well as a keyboard shortcut. The keyboard shortcut is in parenthesis here.

Split | Cut (x) Delete the selected area and put it in the cut buffer for future pasting. If a cut is made on the *Insertion Point* only, without selecting a region, *hard edges* are created.

Copy (c) Copy the selected area and put it in the cut buffer for future pasting.

Paste (v) Paste the material that is in the cut buffer.

Clear (Del) Clear the selected area. If the insertion point is over an edit boundary and the edits on each side of the edit boundary are the same resource, the edits are combined into one edit comprised by the resource. The start of this one edit is the start of the first edit and the end of this one edit is the end of the second edit. This either results in the edit expanding or shrinking.

Paste silence (Shift+Space) Paste blank audio/video for the length of the selected area. Following edits will be pushed to the right.

Mute Region (m) Overwrite blank audio/video on the selected area. Following edits don't move.

Trim Selection Delete everything but the selected region.

Select All (a) Select the whole timeline.

In Cut and Paste editing mode you can *edit labels* as well. By enabling Edit labels in the **Settings pulldown**, or by disabling the Lock labels from moving button on the Program Control Tool Bar, labels will be cut, copied or pasted along with the selected regions of the armed tracks.

Using labels and In/Out points are useful in editing audio. You can set In/Out points for the source region of the source waveform and set labels for the destination region of the destination waveform. Perform a cut, clear the In/Out points, select the region between the labels, and perform a paste.

In / Out Points The In and Out bracket placement is explained here to illustrate their usage. Because of the shape of the markers [and] you may assume that they are inclusive – that everything placed in between would be included in the clip , such as in the case of being transferred to the timeline from the Viewer. In reality, one of the two markers will not include the frame that was visible at the time the

marker was affixed. Depending on whether the *Always show next frame* option is used or not, it is the In or Out marker that will not be inclusive.

To obtain a clip on the timeline exactly as you saw in the Viewer, you must necessarily move the In mark back from the beginning before the first desired frame or move the Out mark forward after the last desired frame, depending on the *Always show next frame* setting.

Some of the confusion can be attributed to the fact that the Viewer shows frames, while the markers determine spaces, i.e. times, that are not visible between frames. You have to think of each frame as being delimited by two spaces – one preceding and one following. The In mark is always placed before the displayed frame and the Out mark is always placed after the displayed frame, while taking into account in its calculations whether the *Always show next frame* option is used or not. If you just remember that the reference of the markers is in the middle of the icon, you will avoid confusion.

Overwrite To perform overwriting within the timeline paste on a selected region (highlighted or between In/Out points). The selected region will be overwritten. If the clip pasted from the clipboard is shorter than the selected region, the selected region will be shrunk. Following edits will move. If the clip pasted from the clipboard is longer than the selected region, the selected region will be overwritten with the first part of the clip and the remaining part of the clip will be written after the overwriting. Following edits will move.

Tracks → Concatenate tracks This operation copies all the assets of every disarmed but playable track and concatenates it by pasting those assets at the end of the first set of armed tracks. They are pasted one after the other, keeping the same order they have on the stack.

Split – blade cut and hard edges: You can cut the tracks into 2 pieces on the timeline by putting the hairline cursor on the place you want to do a cut and then using the character “x” or the scissors tool (figure 5.7).

A *cut* uses a non-empty selection region, where the *blade cut* or *split* has no duration in the selection, just a hairline. As usual the use of cut when a selection is set, deletes/cuts the highlighted area. In the case where an In point or an Out point exists on the timeline, the clip is split at the location of the In/Out point since it has priority over the cursor location. A blade cut simply splits the edit into two edits. In order to have the video and audio aligned, it works best to have `Settings → Align cursor on frames`. When a blade cut occurs, the edges are created as *hard edges*. These are edges that cannot be deleted by track optimizations . CINELERRA-GG has built-in optimization on the timeline. So that whenever two

parts on the timeline are sequential frames, it automatically optimizes by making them into 1 item. So if you are cutting, dragging, editing, or whatever and somehow frame # 40 ends up right next to frame # 41, it optimizes them together. This optimization affects many areas throughout the program code. When you do a blade cut/split, all armed tracks will be included in the cut and green-colored triangles will show on the bottom of the track on both the left and the right side of the cut. This is a *hard edge* marker toggle, as opposed to the soft edge designation for an ordinary edit. The *hard edge* marker can be toggled off/on if so desired. In order to not interfere with the usual drag handles, only a few pixels are used for the toggle so you have to be sure you have the cursor right over the hard edge triangle – when in position, it will be obvious because you can see an arrow pointing to the corner. Use Shift-left mouse button 1 to toggle off/on the hard edge marker on all tracks simultaneously.

NOTE: Hard Edges do not allow trim operations. For more details see: Trimming.

5.5 Drag and Drop Editing

To enable the drag and drop editing mode on the timeline, select the arrow toggle on the control bar at the top of the main program window. Drag and drop editing is a quick and simple way of working in CINELERRA-GG, using mostly only the mouse. The basic idea is to create a bunch of clips , then drag them in order into the timeline, thus building prototype media that you can watch in the compositor. If after watching it, you wish to re-arrange your clips, set effects, add transitions or insert/delete material, just drag and drop them on the timeline.

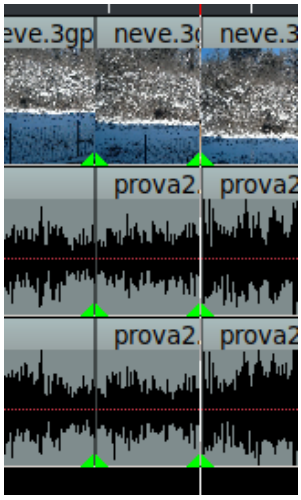


Figure 5.7: *Blade cut*

To simply get started, perform the following operations which are useful for working in a drag and drop editing session. First load your media by using the main menu *File* pulldown and choose *Load files*; make sure the insertion mode is set to *Create new resources only* . This loads the files into the Resources window .

1. Create some video and audio tracks on the timeline using the *Video* and *Audio* pulldowns.
2. Open the Media folder in the Resources window. Make sure the necessary tracks are armed and drag a media file from the Resources window to the

timeline . If the media has video, drag it onto a video track or if just audio, drag it onto an audio track. For a still image, drag it onto a video track.

You can also drag multiple files from the Resources window. When dropped in the timeline they are concatenated. If you have *Display Icons* selected in the Resources window, drawing a box around the files selects contiguous files. If you have *Display Text* selected, Ctrl-clicking on media files selects additional files one at a time; Shift-clicking on media files extends the number of highlighted selections. In addition to dragging media files, if you create clips and open the clip folder you can drag clips onto the timeline.

CINELERRA-GG fills out the audio and video tracks below the dragging cursor with data from the file. This affects what tracks you should create initially and which track to drag the media onto. To drag and drop a file on the Program window, you need to create on the timeline the same set of tracks as your media file.

When you drag your chosen media from the media folder to the timeline, your mouse pointer will drag a thumbnail and, once over the timeline, the outline of a white rectangle, as big as the edit you are going to have appears. Drag the media to the desired position of an empty track of the timeline and drop it. If there are other edits on that track, when you move the white outline over an edit, you will see a bow tie symbol ⋈ appearing at edit boundaries. If you drop the media there, the new edit will start from the edit boundary indicated by the center of the bow tie ⋈.

Since the mouse pointer is in the middle of the white outline, when this rectangle is bigger than the visible part of the timeline, it is quite cumbersome to precisely insert it for long media. Lengthening the duration visible in the timeline by changing the sample zoom in the zoom panel will reduce the size of the white rectangle, making a precise insertion possible.

When you drag and drop edits within the timeline:

- If you drop an edit when bow ties ⋈ are shown, that edit will be cut and pasted starting at the edit boundary indicated by the center of the bow tie ⋈. Following edits on the same track will move.
- If you drop an edit when there are no bow ties ⋈ shown, the original edit will be muted and pasted where you dropped it. No edits will move. A silence will appear in place of the original edit.
- If you have more armed tracks on the timeline than in the asset you are dragging, only the following edits of the tracks affected by the drag and drop

operation will move to the right. This will cause loss of synchronization . To restore it, disarm the tracks affected by the drag and drop operation, highlight the just dropped edit and paste silence over it using the *Edit* pulldown, *Paste Silence*.

Labels sometimes work differently in Drag and Drop editing mode in that you can't drag and drop them. They might be locked to the timebar , even with the Edit labels option enabled. Although with the Edit labels option enabled, if a selected area of a resource is spliced from the Viewer to the timeline in a position before labels, these labels will be pushed to the right for the length of the selected area.

In/Out points can be used to perform Cut and Paste operations in Drag and Drop mode as well as in Cut and Paste mode. Use the *Edit* pulldown to view the list and their keyboard shortcuts.

5.5.1 Copy/Paste Behavior

There are many options for moving, copying, pasting, inserting, and deleting selected *edits*, more commonly referred to by the user as *clips*, when in the Drag and Drop (arrow) editing mode. This makes it easier to avoid constantly having to disarm/arm tracks. To create a selection move the cursor over the clip and just click the left mouse button; remove a selection by left mouse button click again. This will mark your selection with a colored border which contains some red. The easiest way to initially use the various modes is to click on the middle mouse button when your cursor is over a track and a popup displays the modes and shortcuts. However, for those users who prefer the addition of the Ctrl key to add multiple selections as is commonly done for listbox operations, there is a preference in `Settings` → `Preferences` → `Appearance` tab, called *Clears before toggle* that changes the behavior.

When an edit is marked as selected, it can be cut/copied into the paste clip buffer. The constructed clip buffer will begin with the leftmost edit and end with the rightmost edit. The edits may contain media, or be silence, or skipped if they are not selected. The clip tracks are copied from the first track with an active edit selection to the last track with an active edit selection. A clip track can be completely empty if no selection was made on the track. The word *packed* means that the silent edits and empty tracks are not included in the copy to the clip buffer, and all of the elements are packed together, no gaps. Packing a clip buffer makes it easier to move *blobs* of data around. Once the edits have moved and have a relative relationship applied, an unpacked clip buffer allows the media to be copied with the relative positions of the edits preserved.

The *edits* popup is activated on a track and a red and yellow colored reticle appears

to temporarily mark the location when you click on the middle mouse button. An expanded explanation is provided below.

Key	Operations
Drag	Hold down the Ctrl key while dragging to move a single edit elsewhere.
Left mouse button	Selects and highlights the edit under the cursor with a red selection box. Left mouse button also will toggle that clip selection off if it is on.

Popup Label	Key	Operation
Clear Select	Ctrl-Shift-A	Deselect all selected edits – ones that have the red lines around them.
Select Edits	Ctrl-Alt-'	Select all edits within a highlighted area of the timeline * (see note below)
Deselect Edits		Deselect all edits within a highlighted area of the timeline * (see note below)
Copy	Ctrl-c	Copy the selected edits into the copy buffer.
Cut	Ctrl-x	Delete the selected edits after copying them into the buffer. The edits
Mute	Ctrl-m	Delete the selected edits after copying them into the buffer. The space previously occupied by the edits will be replaced with silence.
Copy Pack	Ctrl-Shift-C	Copy the selected edits into the buffer and remove any silent sections.
Cut Pack	Ctrl-z	Delete the selected edits after copying them into the buffer. The edits after the election will move left to occupy the vacated space. The edits in the copy buffer will be packed together within each track.
Mute Pack	Ctrl-Shift-M	Delete the selected edits after copying them into the buffer. The deleted edits will be replaced with silence. The edits in the copy buffer will be packed together within each track.
Paste	Ctrl-v	Paste contents of the copy buffer at the insertion point marked by the red & yellow reticle of the popup menu or the position of the hairline cursor. This is a splice operation which creates space for the edits.

Popup Label	Key	Operation
Overwrite	Ctrl-b	Paste contents of the copy buffer at the insertion point marked by the red & yellow reticle of the popup menu or the position of the hairline cursor. This destroys the current edits in that space.
Overwrite plugins	Ctrl-Shift-P	Pastes plugins that are in the Copy buffer to current location but no clip. Plugins.

** If the selection area created is larger than the edits we want to act on and includes additional edits, those edits will be involved if "Select Edits" or "Deselect Edits" is used. Indeed, these options act on both the selected edits and those that are not selected but are included in the selection area created. In practice, edits that were selected are deselected, while unselected edits are selected.*

The copy/paste behavior respects the armed/disarmed tracks state. A paste of audio on a video track will fail and vice versa. In addition if you attempt to paste edits consisting of more tracks than what is available at that location it will not allowed.

Attaching transitions to multiple selected edits via the *Video* or *Audio* pulldowns is also available. The new transitions are attached at the start of the edits, and will replace any existing attached transitions. As a side note, when in drag and drop mode the end transition does not drag.

5.5.2 Snapping while Cutting and Dragging

Cutting/Snapping edits cuts from an edit handle to the insert point. There are Edit Panel buttons which normally are used to move to the previous or next edit handle/label.

They look like tags and the letter E on the menu bar and are oriented forward/backward. These same buttons can be used to *cut* from the insert pointer to the previous or next edit/label when the ctrl+alt keys are both pressed when the buttons are used. They *snap* off the media instead of doing the standard re-positioning. This is useful to minimize the number of operations necessary to cut between edits/labels.

Instead of using the edit panel buttons, you can more easily use the following keyboard shortcuts to perform the same functions:

snap_right_edit	ctrl+alt+ '.'	
snap_left_edit	ctrl+alt+ ','	
snap_right_label	ctrl+alt +shift '.'	shift+period is the > sign on US keyboards
snap_left_label	ctrl+alt +shift','	shift+comma is the < sign on US keyboards

Drag Snapping if you hold down the Ctrl + Alt keys while dragging using the mouse, once the clip gets near to an edit, a label, an in/out pointer or the start/end of the timeline, the dragged clip will snap next to that marker. The 2 will now be exactly aligned with no gap and no overlap. As you drag the clip close to one of the markers, when they are within a short distance they start to stick and stay that way until you move further away from that distance. Also, the line will turn color from green to yellow while in the sticky phase. In addition, this works for a plugin while dragging so that it will be the same length as the edit.

5.5.3 Copy/Paste clips/medias across Multiple Instances

It is easy to copy/paste clips/media within a single instance of CINELERRA-GG or across multiple instances. The reason this works is because there are hidden X cut buffers and these are used to transmit EDL from 1 instance to another.

Steps to copy from a source timeline and paste to a target timeline:

1. highlight a selection on the timeline in 1 instance of CINELERRA-GG
2. use the Copy icon (shortcut c) on the main menu bar to copy into a buffer
3. move the pointer to another instance of CINELERRA-GG and set an insertion point in its timeline
4. use the Paste icon (shortcut v) to paste the clip to that other instance selection target

5.5.4 Grouping edits

CINELERRA-GG recognizes as a group, the edits of different armed tracks that have aligned beginnings, regardless of whether they have the same source or aligned ends. You can drag these edits around on the timeline to construct your movie by rearranging scenes. If more than one track is armed, CINELERRA-GG will drag any edits which start on the same positions the edit the mouse pointer is currently over. Another method of Grouping of edits is performed as follows:

1. Select each of the clips you would like to be part of a group.

2. Use the desired Copy mode as described above to get into the buffer.
3. Go to the Resources window Clip folder and right click on an empty spot.
4. Choose the Paste Clip option. Now you have a named clip of the current selection.

A more traditional Grouping of edits which make a Permanent Group works as described next. A temporary group is just a set of selected clips and works the same as a single selection.

1. Create a group:
    - Select multiple clips/edits with left mouse button over the clip;
    - Hold down shift and left mouse click over any of the selected clips to create a group;
    - A Group Id and color are assigned to this group making it easy to distinguish.
2. Move a group:
    - Move your cursor over any of the clips within the group; click to select the clip group.
    - Click and Hold down the left mouse button to drag.
3. Dissolve a permanent group:
   To ungroup select any of the group edits/clips and shift left mouse click to ungroup (same as creating). The edits will be ungrouped, and the current selection will be replaced with the ungrouped clips.

The color of the created groups are not muted and are assigned by Group Id going sequentially through the number of 1 to 64. Therefore if you dissolve a group and make it again, it will have a different Group Id and a different color.

5.5.5 Dragging Groups

Dragging while in *Drop and Drag editing mode* (arrow mode) is really easy. Just select the clip or clips you want to drag using the left mouse button, then put your cursor over one of them and drag while holding down the left mouse button. Keyframes, autos, labels, and plugins will also be dragged. Dragging honors armed/disarmed tracks. When you drag there will be some possible colors as defined; depends on how the edges of edits and groups interact:

- Green color means OK to drop in that position as it will fit.
- Yellow color means you can drop here and when you do it will be exactly next to that existing edit.
- Blue color means it overlaps something and this includes overlapping itself.
- Red color means can not drop here because it will not fit in the space.
- Orange color means the track types do not match so it can not be dropped here.

Remember: With the `Shift` key on, it will always *overwrite*. Without the `Shift` key enabled, it always *inserts* only.

The original (older) method of dragging while in Arrow mode, lets you just left mouse click on a single clip or aligned clips and just drag. This older method of dragging does not move any of its effects with it at this time. There will only be a white outline while dragging and it will let you drop only if it fits. You can also perform some dragging and grouping while in the *Cut and Paste editing mode* (ibeam mode) by taking advantage of the Ctrl button in conjunction with the left mouse button.

- Double click selects a column so you can move, for example, the audio and video together by holding down the Ctrl key and dragging.
- A single clip can be dragged without any of its aligned clips, by holding down and Ctrl key and drag.

This last section on Dragging, outlines the difference between *column selection* and *marking selection* . Column selection is available to make it easy to still be able to do some dragging in I-beam mode whereas Marking selection makes it easy to drag clips together that are not columnated.

5.5.6 Selection Methods

Concerning *Selection* methods, the following information is partially pertinent to all editing, but is most important to keep in mind when using Drag and Drop Editing.

Originally, there was the column oriented timeline drag selection which can be seen in 1 of 3 ways:

1. a highlighted vertical column
2. the colored line region between the [in and out] marker labels
3. a single flashing line

The selection priority works like this. When the highlighted vertical drag column is in use (1), it has the highest precedence and is used as the selection. When the column is a single line (2), then the fall back selection is the [in/out] marker region, if they are set. When they are not set, and the cursor is flashing, then the selection start=end and the selection is empty, but it does have a position on the timeline which can be used for editing. This is input for the vertical style cut/paste drag/drop editing.

More recently, in addition to the column oriented timeline drag selection, there is now *group* capabilities which have various *edit* selections. These are created in the Drag and Drop editing mode by clicking edits to toggle select/deselection. These groups are input to a different (more modern) set of cut, paste and overwrite drag/drop editing. You can see this set of operations when you click on an edit with the middle mouse button, and are also available using the shortcuts shown. They are more like text editor commands to include `ctrl+x=cut`, `ctrl+c=copy`, `ctrl+v=paste`, and so on ...The keyboard Delete key is not hooked to these operations, and is hooked to the original editing methods.

In this *group* mode, if there are In/Out markers set, they enter the selection priority queue between the column selection and the cursor only. You can see the In/Out markers selected region colored line across the timebar (slightly underneath where the time, samples or frames show) on the main timeline extending between the [and]. This means that when the highlighted cursor selection is empty, the In/Out selection will be used.

Summarizing: to select multiple edits/clips on the timeline you can simply use LMB on each edit. Or, if in `Settings` → `Preferences` → `appearence tab` → `Flags` we have enabled *Clears before toggle*, use CTRL + LMB on each edit.

There is also a faster way:

1. Arm the tracks you want to be used.
2. Highlight the area as usually you do with *I-beam* mode (or use *In/Out Points*; or position *Insert Point* over the stack of edits). If you miss an edit that you want included, the same method can be used to add it. It is an additive operation. And, it will include the partial edits in the selected area, thus

making it easy to disinclude ones you do not want. It will include silence also but you can remove that or any other edit with CTRL + LMB.

3. Press MMB on a clip inside the highlight area and the PopUp menu is shown with the *Select edits* option on the top.

4. Choose the *Select or Deselect edits* option and the clips inside the highlight area will be selected or deselected.

NOTE: *Select Edits/Deselect Edits* option toggle between select and deselect edits. If you have selected an edit and you want to deselect it but select the edits near to it, you can highlight the area and using *Deselect Edits* (or *Select Edits*) the edits near to it will be selected while the edit selected before will be deselected. Of course it works for more edits selected before too. In practice, edits that were selected are deselected, while unselected edits (included in the selection area we created) are selected. If you want to deselect all the selected edits you must use *Clear Select* option by MMB.

There is a Drag mouse way (this requires that the cursor be on the first clip and from there you start the drag):

1. To Drag Select edits, hold down Alt+Left Mouse Button (LMB) and drag over the selection area. Releasing the mouse results in selecting.

2. To Drag Deselect edits. hold down Alt+Ctrl+LMB and drag over the area to be desected. Releasing the mouse results in deselecting.

If using certain operating systems (for example Ubuntu and Arch) that might field the Alt key before it gets to CINELERRA-GG, you can use the following for both Select and Deselect instead:

1. Press and hold down Ctrl+Alt+LMB on an edit

2. Drag on the edits you want to select

3. Release the Ctrl key

4. Release the LMB

5. Release the Alt key

5.6 Inter-View Mode – Identifying Source Targets

Inter-View mode provides a mapping of a particular media file to its timeline usages. It is somewhat similar to Two Screen Editing in that you make use of the Viewer. It makes it possible to precisely trace and indicate in the media the origin of a particular segment of the timeline and visually indicate the use and distribution that the timeline makes of a particular media. A good example usage would be in the case of a 30 minute interview where you use a few short pieces to make a shorter 10 minute section, find out that you have made the timeline 12 minutes instead and need to cut out another 2 minutes. This feature provides the following capabilities:

- You can see on the timeline all of the places where a particular piece of media was used.
- You can see which parts of that particular media are already used so you do not reuse that same piece again.

Figure 5.8 shows an example of the Inter-View mode mapping preview mini-window.

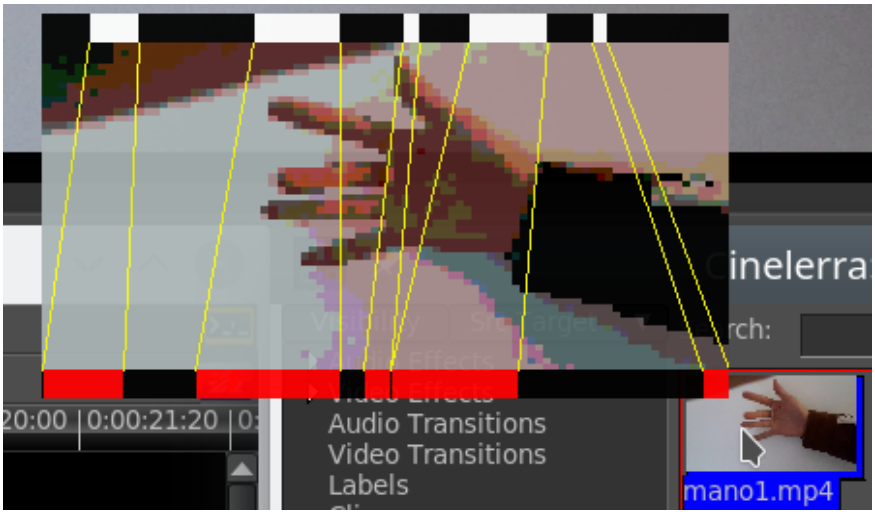


Figure 5.8: *Inter-View mode: white bar→source; red bar→timeline*

There is a second usage of Inter-View mode that can be helpful if some tracks have become misaligned, such as in the case of 2 audio channels no longer being in sync. You can use this mode to get them back in alignment.

Explanation of how to use Inter-View mode will be described here next.

- Do your editing as usual on the timeline until you are ready to see what is used or unused.
- Make sure you are in any of the Preview modes in the Resources window ; you enable the mode using the pulldown to the left of the word *Search*. The option looks like this

.
- Middle mouse click on a thumbnail in the Resources window and a popup occurs of that media with a white colored bar at the top and a red colored bar at the bottom with black sections.

The red/white bars represent the presence and the black sections represent the absence of where that media is used on the timeline. To get to a bigger representation, use the "`f`" key for a full screen. Now you can operate the following buttons to display what you need to see and to move around. It is important to note that *locked tracks* will not be represented. This makes it easy to ignore the audio track segments if you want so there is less confusion in the display.

- Clicking on the top white or black spaces in the top time bar loads the Viewer with the source media, and sets the In/Out [and] pointers to be the selection of that edit.
- Click on a location in the bottom red or black bar, and the main cursor and composer will re-position to the corresponding location on the session timeline.
- Dragging on the red/black bar will correspondingly update the position in the timeline and composer.
- Ctrl-click on the bottom bar and the timeline and composer are re-positioned to the beginning of that edit.
- Shift-click on the bottom bar and a *selection* is made of that section in the timeline and the compqoser is updated with that start position.

Figure 5.9 displays Inter-View window and its relation to the timeline, viewer, and compositor.

The Inter-View mode works for Media, Proxy, and User Bins. When the preview window has only black bars on the top and bottom, it means that this particular media is not loaded in the timeline. So when you are in Proxy, meaning that the Proxy files are loaded on the timeline, there will be only black bars for the corresponding Media file UNLESS there is an audio track associated with the video. Because audio tracks are not proxied, they will show for Media but not for Proxy.

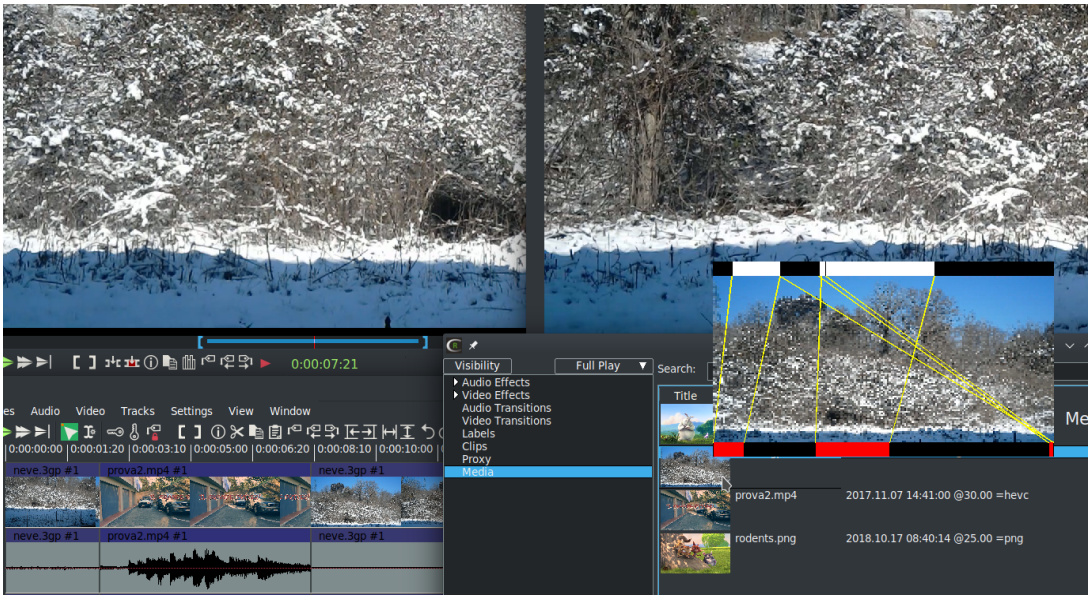


Figure 5.9: *Inter-View mode and the timeline*

5.7 Edit Tools

5.7.1 Edit Length

To set the length of an edit in the timeline, select the region which contains the edit to be modified. Now select the menu bar `Edit` → `Edit Length...` menu item to activate the *edit length* popup (figure 5.10). The duration of the edit can be reset by entering the desired edit length in seconds. Pressing OK will change all of the selected edits (in armed tracks) to the specified length.

5.7.2 Align Edits

When loading media, a common problem is that the various audio/video tracks do not always have exactly the same lengths. For example, you might load audio/video recordings from your camera and be dismayed to see that the audio for each segment is a half second longer than the video. If you load a large set of media clips by concatenation, the audio and video will be more skewed as more media is loaded. Align Edits makes it possible to adjust the edits so the audio and/or video align by adjusting the edits so that the track lengths are consistent. To use this feature, load all of the desired media and select a region which contains all of the edits to be aligned in the timeline. Now select the menu bar `Edit` → `Align Edits` menu item to operate the change. The topmost armed track is used as a template reference, and the rest of the tracks are either cut or padded to align the edit boundaries. Besides aligning audio with the video, you can also align video with the audio if the first armed track is audio. The code performs the

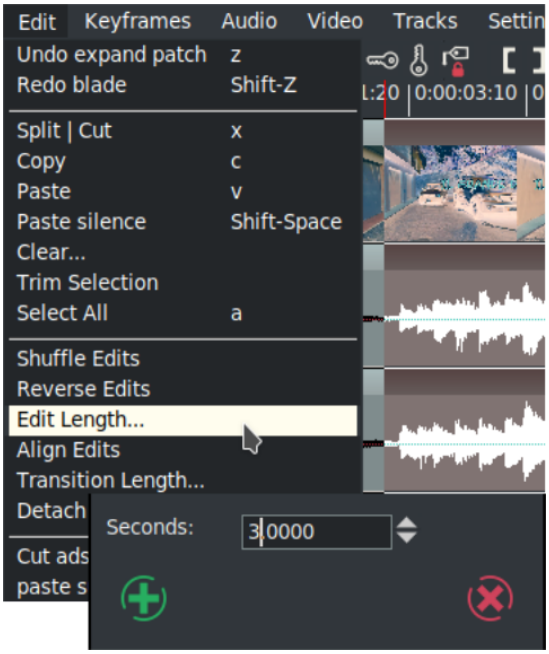


Figure 5.10: *Edit Length window*

following algorithm:

- Use the first armed track as the master track (it must contain data).
- Collect the *edit project start times* on the selected master track. Only edits that are 100% inside the selected area will be used.
- Set all other tracks to match the *edit times* of the template track, either by putting in silence or cutting the region to align the edits on the *edit times* of the master track.

The start time sequence of media and silence edits along the master track are collected as the target alignment boundaries. All armed tracks after the master track are modified so that if the next edit edge is too soon, it adds silence; if it is too late, edits are shortened or deleted past the point of the next target alignment boundary time. Align Edits works best if there are an equal number of Video and Audio sections. Also, it is better to use cuts instead of adding silence – if there are silence edits together, the algorithm will combine the silence edits into a single edit and results may not be as desired.

The first two screenshots in figure 5.11 show the Before, the Highlighted Edits to be manipulated, and the After results for the Align Edits. The third screenshot *adds silence* in the second section as noted in red letters.

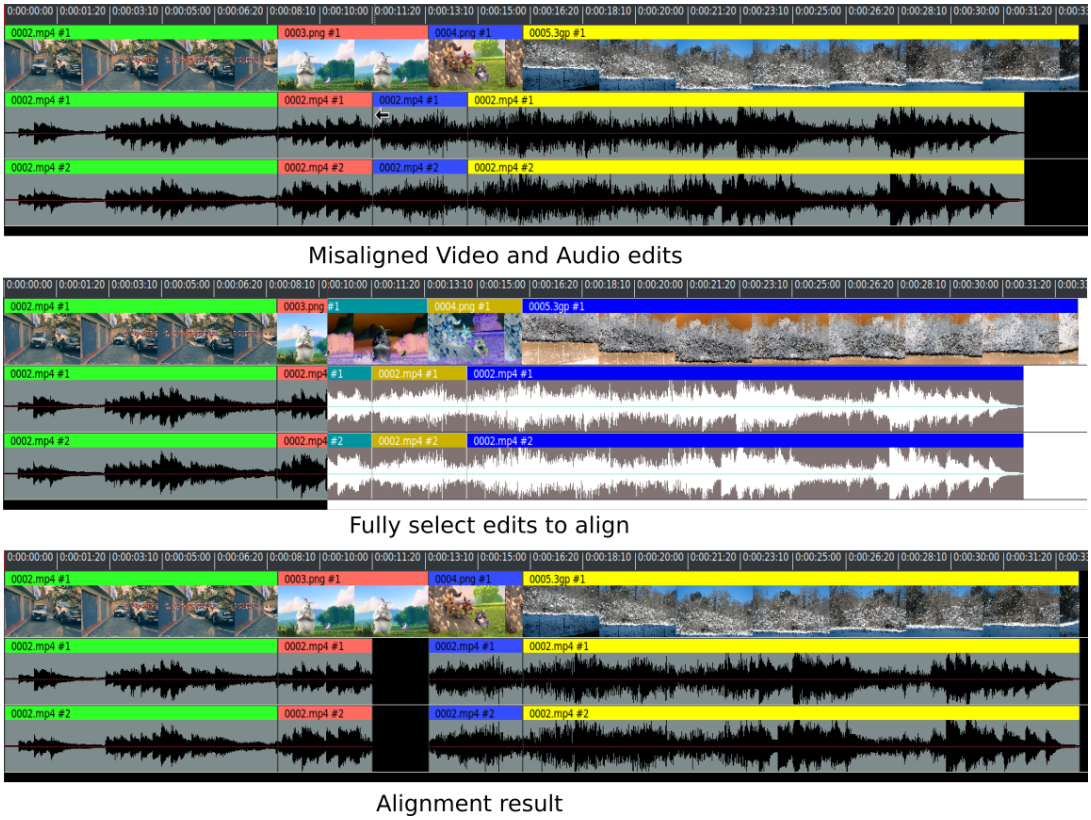


Figure 5.11: *Align edits*

5.7.3 Reverse Edits

The Reverse Edits can be useful to change the order of 2 edits in the case where you would like to put a *teaser* section that occurred in the middle of a movie at the beginning instead, that is, reversed positions. To operate, highlight completely the edit areas you would like reversed and then use the pulldown `Edit → Reverse Edits`.

Figure 5.12 shows the selected / highlighted area to which Edits will be applied. Note the first edit is 0002, followed by 0003, 0004, and 0005 in that order.

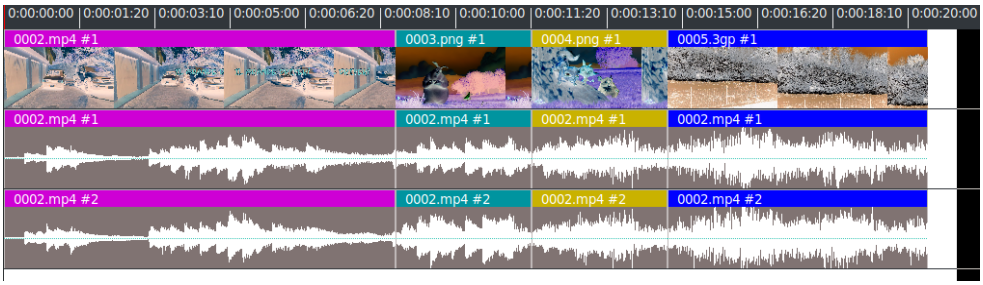


Figure 5.12: *Selected area for Reverse Edits*

Figure 5.13 shows the results of executing *Reverse Edits*. Now you will see the reversed order of 0005, 0004, 0003, and last 0002.

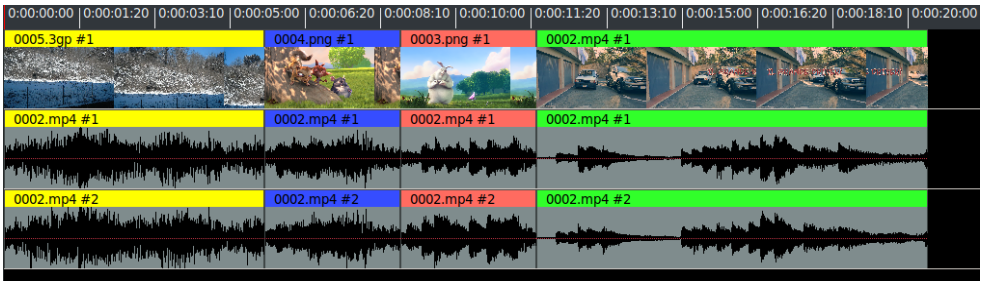


Figure 5.13: *Results of the Reverse Edits*

5.7.4 Shuffle Edits

The file pulldown `Edit` → `Shuffle Edits` will randomly exchange the location of the edits. This feature can be used to change the order of the music like you would do from your MP4 player where you have a playlist of your favorite music. Or perhaps you are creating an advertisement background, you can randomly change it, thus the viewer sees a different order of scenes each time shown.

Figure 5.14 illustrating Shuffle Edits of the highlighted area of the first screenshot on the page. Note the permutation of the fragments resulting in 0002 now being first, then 0004, 0003, and 0005 last.

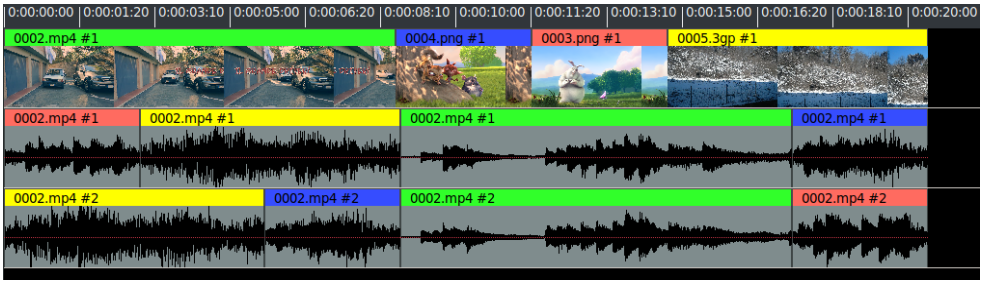


Figure 5.14: *Shuffle edits: the edits are permutated*

5.8 Multi-Session

You can run as many sessions of CINELERRA-GG as your computer resources allow. However, if you are using the same `$HOME/.bcast5`, changes you make for one may impact the others. You can always create and rename a new `.bcast5` from: `Settings` → `Preferences` → `Interface` → `Index files:` and press *Index files go here*

5.9 Multi-Viewer Window Support

You can create as many Viewer windows as you want in CINELERRA-GG. These are handy for users who are adept at working with a lot of different clips simultaneously. By bringing up multiple Viewer windows, each clip can be edited in its own area, making it easy to see all of the separate pieces. After you have loaded some media files, to start another Viewer window, right click on one of the pieces of media in the Resources window. This brings up a menu of several options, one of which is *view in new window*. Choose this option and that media will come up in a new Viewer window for you to work (figure 5.15).

Figure 5.15: *Shown here are 3 Viewer windows and the View in new window popup*

5.10 ShuttlePROv2 and ShuttleXpress Jog Wheels for Editing

The ShuttlePROv2 and ShuttleXpress are affordable jog wheels[1] which can be useful for working with Cin, especially if you do a lot of playing forward/backward, fast/slow/normal, and single frames (figure 5.16). Directions for using the ShuttlePROv2 and the ShuttleXpress with CINELERRA-GG are described next. These devices work by sending keystrokes used in Cin, corresponding to the shuttle action, to the keyboard buffer. The shuttle has been fully integrated into the CINELERRA-GG code so that after the one initial setup, no further intervention is required.

[1] programmatic specifications from Eric Messick

The multi-speed outer wheel works the same and has the same number of S positions on both shuttles but the shuttle Xpress has only 5 keys. Since the majority of user operations will most likely be with the use of the 2 wheels, the slightly smaller Xpress could be a better choice with its 5 easy to reach keys. The Pro is approximately 4 × 7 inches whereas the Xpress is about 4 × 4 inches.

Figure 5.16: *ShuttlePROv2 and ShuttleXpress*

The vendor supplied *string* device names for the shuttles are currently (with the mouse duplication for in at least 1 case:

```
/dev/input/by-id/usb-Contour_Design_ShuttlePRO_v2-event-if00
/dev/input/by-id/usb-Contour_Design_ShuttleXpress-event-if00
/dev/input/by-id/usb-Contour_Design_ShuttlePro-event-if00
/dev/input/by-id/usb-Contour_Design_ShuttlePRO_v2-event-mouse
/dev/input/by-id/usb-Contour_Design_ShuttleXpress-event-mouse
/dev/input/by-id/usb-Contour_Design_ShuttlePro-event-mouse
```

Only 1 necessary initial setup is required due to permission settings for non-root usage. As root, just copy a file that provides the necessary permissions to use the shuttle, then reboot, Example copy:

```
sudo cp {cindat_path}/doc/99-ShuttlePRO.rules /etc/udev/rules.d/
```

then the next time after you reboot, the permissions should be correct. This file only needs to contain one of the following lines depending on which shuttle version

you have/use, but all will be in the file.

```
# for newer PRO model
ATTRS{name}=="Contour Design ShuttlePro" MODE="0644"
# for older PRO model
ATTRS{name}=="Contour Design ShuttlePRO v2" MODE="0644"
# for the Xpress model
ATTRS{name}=="Contour Design ShuttleXpress" MODE="0644"
SUBSYSTEMS=="usb", ATTRS{idVendor}=="0b33", ATTRS{idProduct}=="0020",
    ↪ MODE="0666"
SUBSYSTEMS=="usb", ATTRS{idVendor}=="0b33", ATTRS{idProduct}=="0030",
    ↪ MODE="0666"
```

If you swap your shuttle, for example upgrade from an Xpress to a PROv2, just stop Cin, unplug the original shuttle, plug in the replacement shuttle, and restart Cin. If you start the CINELERRA-GG program and the shuttle does not function as before, stop CINELERRA-GG and then simply unplug it and plug it in again. There are a couple of reasons why it may stop functioning. One is because CINELERRA-GG was not stopped with the usual Quit command and the shuttle was improperly shut down when there was a crash. The other possibility is that a static discharge occurred in the area.

A default shuttlerc file is automatically used when a shuttle device is plugged in when Cin is started. This file sets up the key bindings for CINELERRA-GG to use. You can override any default settings by having a local file in your `$HOME` directory, named `.shuttlerc` to reflect your personal preferences.

5.10.1 How to Modify the Default Key Settings

Detailed information on how to modify your local `.shuttlerc` file is described next, but if you need help you can request more information in the forum at `https://cinelerra-gg.org`. In the `shuttlerc` file, a # always represents a comment and blank lines are ignored. The first thing you must do is copy the system supplied `shuttlerc` file to your `$HOME` directory and rename it as `.shuttlerc` (with a period).

The `shuttlerc` file has sections that in the case of CINELERRA-GG, represent different windows allowing you to set the keys, K1-K15 for the Pro and K5-K9 for the Xpress, the shuttle wheel positions of S0/S1/S-1 for stop, S2 through S7 for wheeling to the right, and S-7 through S-2 for wheeling to the left for reverse. Then there is JR to jog right (clockwise) and JL to jog left (counter-clockwise) for the inner smaller wheel for single frame movement. See the key arrangement on a later page for location of the keys for each of the two different shuttles.

The sections are surrounded by brackets for windows such as CINELERRA-GG (the main window), Viewer, Composer, Resources, Load, and Default. If you want the keys to be defined the same in every window, you can bracket each window on lines one right after the other and then just define one set of keys. The other lines will have the key name/shuttle position followed by its assigned value. The values you use for the keys are usually shortcuts and have to be operationally defined within CINELERRA-GG. For example, the shortcut "f" to go fullscreen is defined so can be used; however the shortcut "h" is not defined so will not do anything. You can check the file, shortcuts.html, for some options to use.

Next are a few actual examples from the default `cindat_path/shuttlerc` file.

The next brackets represent sections. Default, Resources, Load windows all use the same key values.

```
[Default]
[Resources]
[Load]
K5 XK_Home
K6 XK_Button_1 # same as mouse button 1
K7 XK_Button_2 # same operation as mouse button 2
K8 XK_Button_3
K9 XK_End
# for example, in the Load menu, use scroll up to get to the next file name
JL XK_Scroll_Up
JR XK_Scroll_Down
```

Cinelerra with brackets around it next, is the section with some key definitions for the main window.

```
[Cinelerra]

# Most useful functions have to be on K5-K9
# because Xpress only has 5 keys
K5 XK_Home # Beginning
K6 XK_KP_6 # Reverse, or if playing Stop
K7 XK_KP_0 # Stop
K8 XK_KP_3 # Play, or if playing Stop
K9 XK_End # End
...
S-7 REV_16  # Next 6 are reverse keys
S-6 REV_8  # the number on the end represents speed
S-5 REV_4  # number can be decimal up to 64
```

```
S-4 REV_2  # 2 means 2x or double speed
S-3 REV_1
S-2 REV_0.5  # 0.5 represents 1/2 speed
S-1 XK_KP_0  # Because the Shuttle does not generate S0,
     # have to use S-1
S0 XK_KP_0  # Hardware does not generate S0
S1 XK_KP_0  # Because the Shuttle does not generate S0,
     # have to use S1
S2 FWD_0.5
S3 FWD_1
...
```

An explanation for the above REV and FWD key symbol values is necessary to facilitate user preferences. Obviously REV stands for reverse and FWD for forward. You can set any speed up to and including 64x (that is, 64 times the normal speed) on any of the S keys. First in the line is the key name such as S-3 and then the key direction of FWD or REV followed by the symbol for underscore (_) and then the numerical value to use. For example, if you want the 5^{th} forward position, S5, to play $10\frac{1}{2}$ times faster, you would use the statement `S5 FWD_10.5`. Integer or decimal numbers are legal.

For the Viewer, you may want keys defined to do a Splice or an Overwrite so define differently. Note that assignments that contain single character letters must be enclosed in quotes.

```
[Viewer]
# Splice -Viewer only; may be defined
# differently than Composer or Cinelerra
K2 "v"
K4 "b"  # Overwrite
```

To change any key value to an alternative value, just edit the file and make the changes. Besides just keys and alphabetic letters of numbers, you can also use any CINELERRA-GG value that contains the combination with Shift, Alt, and Ctrl. For keys that are not printable characters, you can look up the symbol name to use for a specific operation in the file called: `/usr/include/X11/keysymdef.h`. Some examples:

```
K10 Alt-XK_Left # Go to previous edit \\
K13 Ctrl-XK_Right # Go to next label
```

For sequences of one or more *printable* characters, you can just enclose them in double quotes. For example in the `[Composer]` section, to go into or out of fullscreen mode, automatically start playing and put a label there, you could define a key like this: K7 “f l” - that is printable character f, a space, and printable character l.

After modifying `.shuttlerc`, the next time you use the shuttle, your changes will automatically take affect without even having to stop and restart Cin. However, the first thing to try if problems is to stop CINELERRA-GG, unplug the shuttle, wait a few seconds, plug it in again, and then restart cin. If for some reason, the shuttle keys still do not work after that, you may have an incorrect setup and you will have to correct that first. For example, if you define S5 twice within the CINELERRA-GG setup, it will fail. It is suggested that if you make changes, you should initially uncomment DEBUG in the `.shuttlerc` file and start up CINELERRA-GG from a terminal window so that you can make sure it is working and has no output errors. An error might look like:

```
dupl key name: [Cinelerra]K1
shuttle config err file: /root/.shuttlerc, line:37
```

Keep in mind when changing the values, that the ShuttleXpress has fewer buttons so if you define K1 it will only work for the ShuttlePro.

Any time you are having trouble with your shuttle, you can copy the default `shuttlerc` file from `cindat_path/shuttlerc` to your local `.shuttlerc` file, and edit that to switch to DEBUG mode by removing the # comment from the DEBUG line. But you will have to have started Cin from a terminal window to see the key values. The first time you use the shuttle or after you change the file, the current assignments will show in the terminal window so will look something like:

```
[Cinelerra] # 1
K5[D]: XK_KP_0/U
K5[U]: XK_KP_0/U
```

When you are in DEBUG mode and are just working away, what you will see is something like this:

```
key: 0058 1
key: 0055 0
```

or:

```
shuttle: 00 00 00 00 00
key: XK_Home 0
```

When you change the focus from one window to another, you will see something like this:

```
new focus: 04c00137
new translation: Viewer
key: 0059 1
```

You can also set an environment variable to temporarily use an alternative shuttle configuration file for testing as in:

```
export SHUTTLE_CONFIG_FILE=/tmp/shuttlerc_test
```

The shuttle wheel occasionally will not *stop* after you have wheeled it to play forward. This is a documented known problem from the original code so you just have to joggle it a little in the other direction and then it will stop. S0 does not always generate a signal to do a stop and that is why S1 and S-1 have to be used to relay the stop instead. Also, if you have a fullscreen Composer or Viewer up and the regular one also, the fullscreen takes precedence.

5.10.2 Troubleshooting auxilliary information

In order to see if you hardware was recognized by the operating system, key in:

```
lsusb -v -d 0b33:0030 # for the Shuttle Pro or PROv2\\
lsusb -v -d 0b33:0020 # for the Shuttle Xpress
```

Note 1 Currently, the keys K14 and K15 do not function on the *Contour Design ShuttlePro* but do on the *Contour Design ShuttlePRO v2* due to a Report Descriptor error. You can workaround this by uncommenting `USB_DIRECT` in your local `.shuttlerc` file. This directly uses libusb rather than the generic Linux hid driver. `USB_DIRECT` works for any of the currently tested shuttles.

Note 2 If you are not sure if your shuttle is fully functional, you can verify that the hardware device has been seen by your operating system with this procedure.

1. From a terminal window as an ordinary user key in: lsusb (the first character is a lower case L for list). You will see something like the following depending on which usb device you have the ShuttlePro plugged into:

```
Bus 003 Device 002: ID 0b33:0030 Contour Design, Inc. ShuttlePro v2
```

2. To make sure you have usbmon installed key in:

```
sudo modprobe usbmon
```

3. Next key in the following:

```
sudo od -tx1 /dev/usbmon3
```

   where the last 3 is the same # as the Bus in above. If it lists `Bus 002`, then use `/dev/usbmon2` instead.

4. Now with focus in that same terminal window, press any shuttle key just to see what happens and should see about 12 lines similar to these below – a new set every time you press a single key or the wheel. The lines are usually not important, just the fact that you get a response is. However if you have multiple devices on the same bus, you will get responses from any and all of them. Attempt to isolate your shuttle by temporarily unplugging unnecessary devices on the same bus or plug the shuttle into a different usb port that has fewer devices.

```
0000000 80 70 99 75 53 8c ff ff 43 01 81 02 03 00 2d 00
0000020 4e 61 5c 5c 00 00 00 00 8d 2c 06 00 00 00 00 00
0000040 05 00 00 00 05 00 00 00 00 00 00 00 00 00 00 00
0000060 01 ff 00 00 00 80 70 99 75 53 8c ff ff 53 01 81
0000100 02 03 00 2d 3c 4e 61 5c 5c 00 00 00 00 b1 2c 06
0000120 00 8d ff ff ff 05 00 00 00 00 00 00 00 00 00 00
0000140 00 00 00 00 00 80 70 99 75 53 8c ff ff 43 01 81
0000160 02 03 00 2d 00 4e 61 5c 5c 00 00 00 00 3d d7 09
0000200 00 00 00 00 00 05 00 00 00 05 00 00 00 00 00 00
0000220 00 00 00 00 00 00 ff 00 00 00 80 70 99 75 53 8c
0000240 ff ff 53 01 81 02 03 00 2d 3c 4e 61 5c 5c 00 00
```

```
0000260 00 00 64 d7 09 00 8d ff ff ff 05 00 00 00 00 00
```

5. Next press the key that you want to verify is functioning – if no new lines show up, then the key is non-functional so there is a hardware problem. If you get output, then perhaps there is a problem with your software setup.
6. Use Ctrl-C on the terminal window when done to get back to the prompt.

Note 3 Another method for testing to make sure your model of the Shuttle does not have different key definitions than the one that CINELERRA-GG was coded for is to do the following.

1. Locate the shudmp.C program in your CINELERRA-GG directory.
2. Compile that with the command: `c++ shdmp.C -o shudmp`
3. Make the file executable with the command: `chmod +x shudmp`
4. Execute:

```
sudo ./shdmp /dev/input/by-id/usb-Contour\_Design\_ShuttlePro-event-if00 #
    ↪ substitute your shuttle
```

Then press your shuttle key that is having problems and check the results. They should look like:

Listing 5.1: *Example for K7*

```
event: (4, 4, 0x90007) #The last number, 7, is the expected Key number.
event: (1, 262, 0x1)
event: (0, 0, 0x0)
event: (4, 4, 0x90007)
event: (1, 262, 0x0)
event: (0, 0, 0x0)
```

Listing 5.2: *Example for K15*

```
Example for K15:
event: (4, 4, 0x9000f) #The last number f is 15 in hexadecimal and is the expected Key.
event: (1, 270, 0x1)
event: (0, 0, 0x0)
event: (4, 4, 0x9000f)
event: (1, 270, 0x0)
event: (0, 0, 0x0)
```

When done, you will have to Ctrl-C to get out of the program.

Note 4 For developers, if you have a pre-UEFI Secure Boot kernel it is also possible to do the following for further in depth testing:

```
ls /sys/kernel/debug/hid \# to locate numerical value of the shuttle, e.g. 0003:0B33
    ↪ .0030.0006
cat "/sys/kernel/debug/hid/0003:0B33.0030.0006/rdesc" # substitute your own
    ↪ numerical value
cat "/sys/kernel/debug/hid/0003:0B33.0030.0006/events" # press keys to see the results
```

5.10.3 Shuttle key default arrangement for CINELERRA-GG / Composer / Viewer:

The following is the default setting for the ShuttlePROv2 and ShuttleXpress (table 5.3 and table 5.4). This page can be quickly requested from CINELERRA-GG by simultaeneously pressing the `Alt` key on the keyboard and any button on the Shuttle.

Table 5.3: *ShuttlePROv2 key default arrangement for* CINELERRA-GG */ Composer / Viewer*

K1 Label	K2 Future use Splice (viewer)		K3 Future use Copy	K4 Clip Overwrite (viewer)		
K5 Home	K6 Reverse	K7 Stop Fullscreen (viewer / compositor)	K8 Play	K9 End		
Home(Defaults)	MouseBtn1(D)	MouseBtn2(D)	MouseBtn3(D)	End(Defaults)		
		Shuttle Outer Wheel Play forward (first row) or Play reverse (second row)				
S1=Stop S-1=Stop	S2=1/2 S-2=1/2	S3=Normal S-3=Normal	S4=2x S-4=2x	S5=4x S-5=4x	S6=8x S-6=8x	S7=16x S-7=16x
K14 Toggle In		Jog Left Frame reverse Scroll up(Defaults)	(Inner Wheel)	Jog Right Frame forward Scroll down(Defaults)		K15 Toggle Out
		K10 Previous Edit Future Use(Viewer)		K11 Next Edit Future Use(Viewer)		
		K12 Previous Edit Previous Label		K13 Next Edit Next label		

Table 5.4: *ShuttleXpress key default arrangement for* CINELERRA-GG */ Composer / Viewer*

K5 Home	K6 Reverse	K7 Stop Fullscreen (viewer / compositor)	K8 Play	K9 End		
Home(Defaults)	MouseBtn1(D)	MouseBtn2(D)	MouseBtn3(D)	End(Defaults)		
		Shuttle Outer Wheel Play forward (first row) or Play reverse (second row)				
S1=Stop	S2=1/2	S3=Normal	S4=2x	S5=4x	S6=8x	S7=16x
S-1=Stop	S-2=1/2	S-3=Normal	S-4=2x	S-5=4x	S-6=8x	S-7=16x
		Jog Left Frame reverse Scroll up(Defaults)	(Inner Wheel)	Jog Right Frame forward Scroll down(Defaults)		

CHAPTER 6

Advanced Editing

6.1 Proxy

Proxies were introduced to allow for a smoother timeline experience. Full HD and everything from 4K size up are usually very large in file size. In addition, some commonly used codecs (for example h264/5) are very compressed and interframe type, so they are more CPU intensive for timeline playback. These files cause performance problems on smaller computers. As a solution, proxies were introduced to reduce the file size. Reduced means to minimize the resolution/dimension, as usually not the full resolution is needed because the compositor generally occupies only a fraction of the computer screen. The proxy provides a scaling factor which indicates how much the original resolution should be reduced. There is also the *proxy 1:1* option that maintains the original resolution but still allows you to modify codec parameters (lowering the bit rate, for example). This is explained in the next section, 6.2. Always remember when getting ready to render the project, disable Proxy to return to the original settings with the highest quality.

To use proxy select `Settings` → `Proxy settings` on the main window. You can choose ffmpeg as the *File Format* and for the *File Type* a choice of various codecs associated with that. A good choice for *File Type* is the default of mpeg or mov which can usually be quite fast. In addition you can modify values for that codec by clicking on the wrench icon. Once you have created a proxy file, it will be reused and does not have to be created again as long as you do not change certain parameters. This saves a lot of CPU time.

There are two main ways to use Proxy, with or without *Rescaled to project size* option checked. The options of the *Scale factor* may be different if *Rescaled to project size* is checked versus unchecked. It is a special case when the *Scale factor* is set to "1" and it does not matter if *Rescaled to project size* is checked or unchecked.

The usual steps to use Proxy are:

1. Check that *File Format* is set to `FFMPEG`.

2. Check or uncheck the *Rescaled to project size* (FFMPEG only). The difference with this option will be explained here 6.1.1. But for old computers checking *Rescaled to project size* is not recommended; again due to requiring more CPU. You need to know that when *Rescaled to project size* is unchecked some Effects/Plugins do not work as expected because they use pixel values instead of percentage values as is used in the Project Format. For example *Title*, *Blur*, *Sketcher* plugins do not work right if unchecked, while *Perspective* and *Crop&Position* plugins work correctly for both checked and unchecked.

3. Choose the *Scale factor*. The options may be different if *Rescaled to project size* is checked or unchecked. The smaller the *Scale factor* is, the smoother playing on the Timeline will be ($\frac{1}{8}$ is smaller than $\frac{1}{4}$), but on the Viewer/-Compositor the image will be degraded/pixelated more.

4. Set the *File Type* in the *File Format* for FFMPEG, such as mp4, mpeg, or mov. Still images like jpg, tiff, and png will be converted to PNG files with alpha channel if it had an alpha channel.

5. Click on the wrench icon to configure video compression; just use the default values if you are unsure and prefer not to experiment.

   a) Select the Compressor type (codec).

   b) Change Bitrate, or Quality, and Pixels. Because not all codecs use the alpha channel for transparency, check that the letter "**a**" is in the Pixels name (such as yuv**a**420p or rgb**a**) to retain the alpha channel.

   c) Click on the OK button to close the Compressor window.

6. Now click on the `OK` button to start the rendering for the proxy.

7. Look in the lower right corner of the main window where there is a progress bar and a percentage value showing the progress of the creation of proxies, and in the lower left corner there is the *Creating proxy files... ETA: hh:mm:ss* text with the "Estimated Time of Arrival (completion)" in hours, minutes, seconds. Depending on how many files your project uses, the file size, which codec is used, the scale factor, and cpu performance, creating proxies may take some time.

You will know that Proxy mode is enabled because:

- A white colored **P** meaning ***P**roxy*, or a white colored **S** meaning *Proxy **S**caled* icon is shown in the upper right corner of the main window to the left of the FF icon. If using the Cakewalk or Neophyte theme it will be a **Ps** icon.
- The Proxy folder in the Resources window has been populated with the names and/or icons of the proxy media.

- The edit title bar in the timeline will have the name of the proxy file instead of the original name. So if the video filename is `VID_20221012.mp4`, the Proxy Scale Factor is $\frac{1}{4}$, and the Proxy container is mov, then the proxy name will be `VID_20221012.proxy4-mp4.mov`.

The Proxy icon is a toggle button so that you just click on the icon to change modes. Or it can be toggled using the `Ctrl+R` shortcut.

Proxy State	**Rescaled to project size (FFMPEG only)**	**Proxy icon for themes: Cakewalk/ Neophyte**	**Proxy icon for themes: the Others**
Off	checked/ unchecked	no icon	no icon
Active (Enable)	checked	Ps: P(green) s(white)	S (white)
Disable	checked	Ps: P(red) s(white)	S (gray)
Active (Enable)	unchecked	Ps: P(green) s(red)	P (white)
Disable	unchecked	Ps: P(red) s(red)	P (gray)

Table 6.1: *list States of the Proxy icon*

When you disable Proxy using the toggle button for Proxy, or its shortcut (`Ctrl+R`), in the Proxy Settings window you will see the *Scale Factor = 1*, but if you look at the Active Scale and the State, you can understand what your setting is. Proxy can have three possible States: Off, Active, Disabled. The difference between Off and Disabled is that Disabled means you have set the Proxy and it is temporarily switched off (and you can see the icon in the upper right corner of the main window); you can enable proxy by just clicking on the icon, so the State will then be changed from Disabled to Active.

6.1.1 How Rescaled to project size (FFMPEG only) works

- When *Rescaled to project size (FFMPEG only)* is checked, the size of the video data to be computed will always be the same as the Project Format. The size of your video is scaled down to the *Scale factor* BUT then it is rescaled up to the Project Format. For example, your Project Format is 1920x1080 and Proxy's Scale factor is $\frac{1}{4}$, then your videos (1920x1080) will be scaled down to 480x270 and then up-scaled to 1920x1080. All the Effects/Plugins work correctly because the original resolution/geometry/size is not changed. For an old computer, this option is not recommended.

- When *Rescaled to project size (FFMPEG only)* is UNchecked, the size of the video data to be computed will be reduced to the *Scale factor*. The size of your video is scaled down to the *Scale factor*. For example, your Project Format is 1920x1080 and Proxy's Scale factor is $\frac{1}{4}$, then your videos (1920x1080)

will be scaled down to $480x270$. It works as if the Project Format were $480x270$ (and really it works). If you take a look at `Settings` → `Format...` when Proxy is enabled (Active), you can see that the Width and Height are changed according to the *Scale factor*; you absolutely must NOT change these values! The good thing is that your computer will be faster. The bad thing is that some Effects/Plugins do not work as expected because they use pixel units. Examples are *Title* and *Blur* plugins. A workaround is needed for that, such as using Inkscape to create a title and export it to PNG that you load instead of using the *Title* plugin.

Note: the dimensions of the frame/image are downscaled to the Scale factor; so if the Scale factor is $\frac{1}{4}$ and the dimension of the image is $4288x2848$ pixels, that image will be downscaled to $1072x712$. It doesn't matter what the Project Format is.

6.1.2 Using proxy

To insert a clip/video from Resources window to the Timeline when Proxy is enabled (Active) you MUST Drag & Drop that clip/video from the Proxy folder instead of the Media folder, as well as for Nested clips. And also if you want to load the clip in the Viewer or Compositor. You can use clips in the Clips folder in any State of the Proxy - it does not matter if Proxy is enabled or disabled.

Rendered proxy media is saved in the same directory as the original media. As usual, you can delete proxy files from the project or disk in the Resources window if you no longer want to retain them. You should **always save your project with the Proxy disabled**.

If you create proxies for Nested clips they will be saved in `$HOME/Videos` unless you modify that in `Settings` → `Preferences`, `Interface tab`, *Nested Proxy Path*. If you change anything in a Nested clip using the *Open EDL* feature, you must first delete its old proxy file from disk because it is not updated automatically, and then enable Proxy again so that a new one is created.

Not all the settings for Proxy are saved in the project: the values of Active Scale and the *Rescaled to project size* are saved but not the *File Format*, *File Type* and all video compression data. A good idea is to record this information in a README file about your project where you make a note of your Proxy settings for that project. This is especially important because some Projects may use different Proxy settings and if you load a project where the Proxy is set and disabled, it could use a different setting because you changed the Proxy settings in a project you were working on before. If in that Project its proxy files were not deleted and you want to reuse

them to save time, unfortunately they will be created again because the proxy settings have changed.

If the *Creation of proxy on media loading* option has been checked, then when you load other media later, proxies will be created for them automatically, or else you can change the Proxy's State from disabled to enabled (Active) using the Proxy icon. Previous proxy renders will be reused unless you change one of the following proxy parameters: *Scale factor*, *File Format*, or *File Type*. If proxies have been created and then you decide to change only the Compression type (Codec, Bitrate, Quality, Pixels), without changing the *Scale factor* or Container, you must delete proxy files from the disk beforehand. That is because CINELERRA-GG does not read the compression type; it does read the filename and if the filename does not change, CINELERRA-GG thinks the proxy was already created.

6.1.3 Scale factor set to 1, a special case

This special case is used to temporarily change the codec parameters of your video while maintaining the original resolution - for example maybe your videos use long-GOP or the video codec is too hard to handle due to a strong compression. It does not matter if *Rescaled to project size* is checked or unchecked. For old computers, or underpowered computers, this option is not recommended.

6.1.4 Setting the proxy

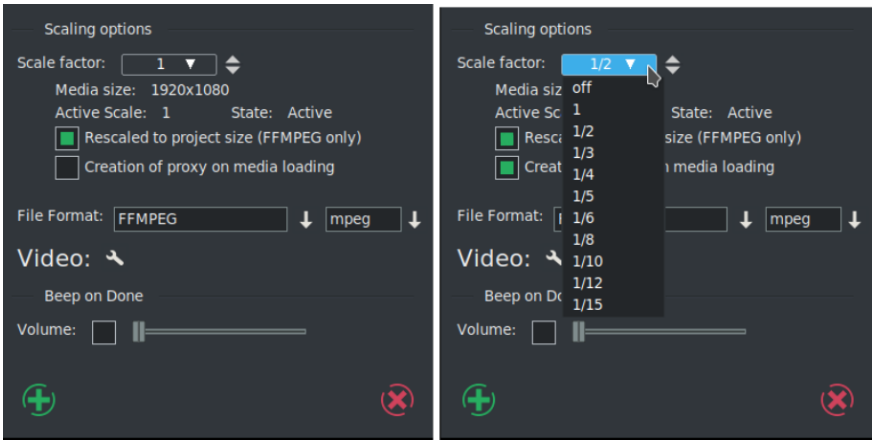


Figure 6.1: *Proxy settings dialog*

Screencast in figure 6.1 shows the *Rescaled to project size* checked so you can still use plugins correctly and the original project size is kept. The *Scale factor* pull-down gives you available size options. Note the new media dimensions shown (partially covered). If the size is an odd number, 1 is added to make the dimensions both even numbers.

To go back to the original media permanently, simply set the *Scale factor* to Off or disable the Proxy by clicking on its toggle button. However, if you decide to reuse the proxies, those which have not been deleted from the Hard Disk (but have been removed from the project) will be used without re-rendering - this saves a lot of time. To completely remove the created proxies you will have to delete them manually from the Hard Disk. There is also a convenient *Beep on done volume* slider included so that you can work on other tasks until there is an audible notify of completion. The default volume is set to 0 for no audible notify.

A good choice for classic proxy settings with a 1080p source video is:

```
Scale Factor:  1/4
File Format: FFMPEG -mpeg
Video Preset:
Compression: mpeg.mpeg
Bitrate: 2000000
Quality: -1
Pixels:  yuv420p
```

If you get errors for some videos, such as those with strange variable bit rate or some types of files made on a smartphone, a usually reliable alternative is to change the following parameters:

```
File Format: FFMPEG -mov
Video Preset:
Compression: mov.mov
```

Or if you want small files with high image quality, a *File Format* of m2ts is optimal. For example a 1 GB file can be reduced to 50 MB with scale $\frac{1}{2}$.

If you get error messages when creating proxies, check the Video wrench settings. These usually default to values that are expected to work correctly for the *File Format* and codec you selected but they can be changed and may result in errors. If you get an error message of *check_frame_rate failed* followed by *Error making proxy* in the popup Errors window, do just that and check the Frame rate value by going to the Resources window, Media folder, and use the right mouse button for the *Info* option for the specific media in question. You can change the frame rate in this window to a more codec acceptable value. Different codecs may have different legal values.

A faster method of creating the proxy file outiside of CINELERRA-GG for users who

have vaapi hardware acceleration capability on their graphics board is via a script. Speed up could potentially be 10 to 30 times faster depending on the parameter values you choose and the size of your video file. This script is included in the doc subdirectory as `vaapi_proxy.sh` for users not using an AppImage version and is shown below. You will have to tune it to your specific needs for the vaapi_device name (currently set to `/dev/dri/renderr128`), file type (currently set to h264 and mp4), and proxy size (now is 6). Once you have created the proxy file, you then start CINELERRA-GG, load the original un-proxied video, use the Settings pull-down and choose the Proxy settings options as you would normally, but because you already have created the existing proxy file with the correct name, it will be automatically loaded and will not have to create it. The filename of the proxy file must be in the same location as the original video and have the exact required naming conventions and scale factor.

```
#!/bin/bash
filename="$1"
fileout="${filename%.*}"
proxy="6"
# Hardware encode AMD
ffmpeg -threads 2 -hwaccel vaapi -vaapi_device /dev/dri/renderD128 \
 -i "$1" -c:v h264_vaapi -vf "format=nv12,hwupload,scale_vaapi=iw/'$proxy':ih/'$proxy
      ↪ '" \
 -vcodec h264_vaapi -preset fast -c:a copy \
 -bf 0 -profile:v 66 "$fileout".proxy"$proxy"-mp4.mp4
```

6.1.5 Proxies with Alpha channel

Next are some examples of tested and working configurations that maintain the alpha channel:

```
Proxy #1
  Scale factor: 1/4
  Rescaled to project size (FFMPEG only) = unchecked
  File Format: FFMPEG | qt
  Video Preset-->
    Compression: png.qt
    Bitrate: 0
    Quality: -1
    Pixels: rgba (or rgba64be)


Proxy #2
  Scale factor: 1/4
```

```
  Rescaled to project size (FFMPEG only) = unchecked
  File Format: FFMPEG | qt
  Video Preset-->
    Compression: magicyuv.qt
    Bitrate: 0
    Quality: -1
    Pixels: yuva444p


Proxy #3
  Scale factor: 1/4
  Rescaled to project size (FFMPEG only) = unchecked
  File Format: FFMPEG | qt
  Video Preset-->
    Compression: openjpeg.qt
    Bitrate: 0
    Quality: -1
    Pixels: yuva420p


Proxy #4
  Scale factor: 1/4
  Rescaled to project size (FFMPEG only) = unchecked
  File Format: FFMPEG | pro
  Video Preset-->
    Compression: prores_4444.pro
    prores_4444xq.pro
    Bitrate: 0
    Quality: -1
    Pixels: yuva444p10le


Proxy #5
  Scale factor: 1/4
  Rescaled to project size (FFMPEG only) = unchecked
  File Format: FFMPEG | mkv
  Video Preset-->
    Compression: user_ffvhuff.mkv
    Bitrate: 0
    Quality: -1
    Pixels: yuva4--p
            yuva4--p--le
```

6.2 Transcode

Transcode, an option under the *Settings* pulldown right next to the Proxy settings option, is a type of full resolution **1:1 Proxy** . The process of transcoding works directly from the resource; it is independent of the timeline. All of the loaded asset media will be converted, that is, rendered in the selected format and loaded onto the timeline. Menu choices besides the usual File Format and File Type include: *Tag suffix* (to add to media filename), *Remove originals from project*, *Into Nested Proxy directory* (an option to have the file saved here instead of the location of the original media), and *Beep on done* volume.

The settings of the project have an effect on the outcome, for example the dimensions are taken into account. The resulting files are also larger than if they were created directly with ffmpeg. Transcode works for videos with or without audio and even single frame files, like png's. If you have a video file that also contains audio, and you convert only the video, the original audio will stay on the timeline if do not check *Remove originals from project*. Or vice versa if audio converted and not video. Multiple stream media will only transcode the first stream (this would be like the TV channel recordings in the United States). You will get an error message if you already have a transcoded file in the selected format with the same suffix name and try to transcode it again with a different selection made – you will have to delete that file first. An example would be an already converted file that has both video and audio and now you request video only.

The BIGGEST gain from using this is if you have media that is not *seekable* , that is, you can play it from the beginning but can not move to another spot and have the audio or video play correctly. A video file with no keyframes makes seeking next to impossible, but then a Transcode generally adds these keyframes. This is particularly useful for the `mkv` container, which often has seek problems. For more details on mkv container see A.2.1.3

Another important function of Transcode is being able to convert the project's media into a high-quality *mezzanine* codec (sometimes also called *intermediate* codec), which makes timeline work lighter and more efficient. In fact such codecs (ffv1, ProRes, DNxHD, OpenEXR, huffyuv, etc) are generally little or not at all compressed; the type of compression is intraframe –more suitable for editing, and the image quality (4:2:2 up; 10-bit color up; floating point; etc) is suitable for *Color Correction*, *Chroma Key* and *Rotoscoping*. The use of mezzanine codecs leads to very large files, so you need to make sure you have enough storage space.

NOTE: CINELERRA-GG cannot do *remuxing* without transcoding. For remuxing only, use *ffmpeg* as shown in the following script. First move to the folder containing the files to be remuxed; the script takes all video files of a certain extension

(in the following example `avi`) from the folder and its subfolders and makes a remux in a new container (in this example `mkv`) inside the new folder `remux`. The internal codec will remain the original one. Here is an example script:

```
for f in $(find . -name '*.avi'); do ffmpeg -i "$f" -c:v copy -c:a copy "remux/{f%.*}.mkv
    ↪ "; done
```

6.3 OpenEDL

To edit EDL that is included with your project as Clips, Nested Clips, Referenced File, or Xml you can use the option *Open EDL* in the Resources window for the highlighted media. Then with a simple button click you can return to your main timeline project. For example, if you have a nested clip that originally had several plugins added to it before it was nested, you can edit those plugin parameter values. Previously to make any changes to these types of EDL you had to remake the whole clip from scratch. The program is actually "opening" the file as it currently exists for that particular media so that it can be edited separately from the project EDL. In other NLEs the term *sub-timeline* or *sub-project* is used. It is important to know that all media used with Open EDL must be loaded on the timeline Main Program window and created from there in that project.

Here is how this works. In the Clip or Media folder or on a timeline EDL edit, the option *Open EDL* for the highlighted clip or nested clip is available so that when you choose this option, that EDL will be brought up on the timeline superseding the current EDL that exists on the timeline . Now, once the clip is open on the timeline, you can edit it however you want. The previous timeline EDL is *pushed onto a stack* so it can be recalled by *popping the stack* with a click of the left mouse button in the upper right hand corner of the timeline to the left of the *shell cmds* icon. Initially this button displays a 0 to indicate your initial timeline/project. Then this button will read 1 if you choose *Open EDL* and then back to 0 and your original timeline with the left mouse click. You can go several levels deep so instead of 1, it could be 2, 3, ... but this requires some thought to avoid potential confusion.

An example of a typical set of steps to follow is:

1. Load your media using insertion strategy of *Replace current project*. There will be number 0 in the upper right hand corner of the main menu with the tooltip of *Close EDL*.

2. Highlight a selection on the timeline and press the *To clip* icon and click the green checkmark OK.

3. In the Resources window, open the Clip folder and you will see that Clip 1 is present.
4. Highlight Clip1 and right mouse the item to bring up available options and select *Open EDL*.
5. Now you will see the timeline change from the original media to just the clip content and the number in the upper right hand corner will change from 0 to 1.
6. Add a visible effect, like AgingTV to the timeline.
7. Click on the number 1 in the main menu bar to see the timeline restored to the original media.
8. Drag the clip from the Resources Clip folder to the timeline and you will see the AgingTV effect.

NOTE: *Open EDL* was not designed to work from the Viewer window. It needs clips to have been created from the timeline Main Program window Project in order to keep track of levels. Clips created in the Viewer have different min and max X/Y dimensions and zoom values.

You can follow the same steps as above by first using the option *Nest to media* in the Clip folder which nests the clip and moves it out of the Clip folder to the Media folder. Then use *Open EDL* on the Nested EDL in the media folder. When you Open EDL and edit the changes, those changes will take affect on any and all occurrences of that nested clip on the current and/or original timeline. The option to unnest that clip and put that back into the Clip folder is the option *EDL to clip*. The nested clip is still in the Media folder. It will now have a name of the next available Clip number but the comment contains the previous name so you can tell where it came from.

Instead of using the number on the main menu to close the current EDL, both the Media and Clip folders have *Close EDL* options with the left mouse button. Clicking on the number button is quick and easy but for infrequent usage it is not obvious, whereas if you use *Open EDL* you see *Close EDL* right below that and so it is very obvious. In addition in the case of where you have opened a EDL, and you no longer see that clip in the folder, the right mouse button where no media is highlighted will also display the Close EDL option.

In addition to the *Open EDL* option in the Resources menu, this option is available on the timeline when the cursor is on an EDL-type edit. To get to this option, click on the middle mouse button on that edit. If it is not EDL, the option will not be shown. In summary:

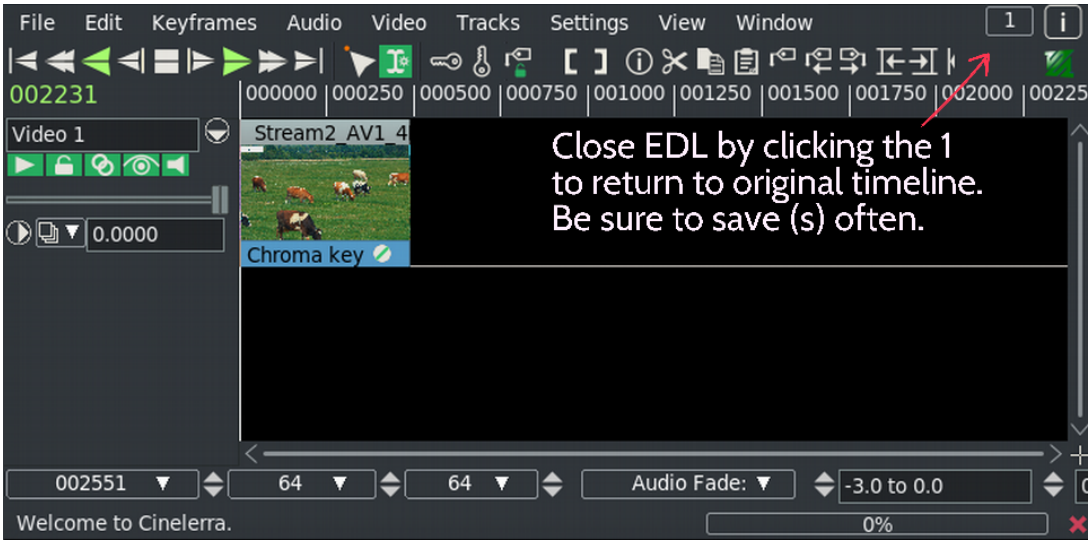


Figure 6.2: *Once you have an Open EDL, the easiest way to close it.*

Media folder of Resources window	Open EDL for Nested or Referenced EDLs
Clip folder of Resources window	Open EDL for clips
Track timeline	Open EDL for Nested or Referenced EDLs

An aside – when nesting and unnesting clips to take advantage of this feature, names of the media can lead to some confusion. For example, if you nest a clip, the new name in the Media folder is the word *Nested* followed by an underscore with the date and timestamp, another underscore, and then the clip name. Then when you unnest this Media folder clip via the *EDL to clip* option, the name will be changed in the Clip folder to the next available Clip number. However the comment field will reflect the nested clip name from which it was derived. To avoid confusion you can easily change the name for these clips in either the Clip or Media folder because they are not real files at this point. To do so, highlight the clip name in Resources, click on Info and type in a new name.

For additional safety, the *Open EDL* feature includes additional backup capabilities. Automatically CINELERRA-GG saves a backup when certain changes are made or you can always use the shortcut 'b' to do one yourself, although keep in mind it will be overwritten whenever CINELERRA-GG wants to do another backup. Now there is a shortcut for the backup shortcut 'b' so you can keep your hand on the mouse instead of the keyboard. Just click on the number button in the upper right hand corner of the main window. If number is at 0, it backs up to backup.xml, if at 1, it backs up to `backup1.xml` and so on ...up to `backup9.xml`.

When *Open EDL* is invoked, the current EDL and current undo stack are both

pushed, and the active session EDL is replaced with the target clip/nested edl. A new undo stack is created, and the active `backup.xml` file name is decorated with the stack level. So, `backup.xml` is `backup1.xml` when your edits are at stack level 1, `backup2.xml` at stack level 2, and so on. This means that if you *load backup* at stack level 1, the session will reload from history at stack level 1, not the main session.

See a real-world workflow at appendix Workflow with OpenEDL and Nested Clips

6.4 File by Reference

It is sometimes handy to have EDL assets not as a copy, but as a reference that is automatically updated into your project. Suppose you have several short videos that at the end have the same credits which include the current year such as 2021. But now it is 2022 and all of the videos would have to be individually updated with the new date. By including a *Referenced File* as the EDL file type when you create each of the videos, you can just change the one credits xml file and the next time you load one of the videos and render it, it will now automatically have the updated information.

The purpose of this feature is to be able to rework a smaller section of a global master project at any time, which can be done by an "assistant" (i.e. external software like GIMP or Inkscape, ...) and then this work is automatically reflected in the global master project. It is for **advanced usage only**.

Up until the addition of this feature, CINELERRA-GG has always used copies and no direct reference in order to ensure original data is never compromised (No Destructive editing). In the usual case, subprojects as xmls are copied into a master project where subprojects had been inserted, so that if you change something in a subproject or delete a subproject, it would have no affect on the master project. But now with *File by Reference*, any project that uses a referenced file will automatically include any changes made to the referenced file when loaded. At the same time, if you use the EDL file NOT as a referenced file in a project since it is then just a copy, it will not be updated. Because of this difference, the user needs to be very aware of what using this feature could do.

Use with extreme caution. However, there are several built-in safety features and a warning that should never be turned off even though it gives you the option to do so. These include:

1. When the `File, Load files` menu is opened, the EDL strategy will always be set to just EDL as default. Although, if you use Apply and leave

the Load Menu open, it will stay changed to what you selected until it is re-opened.

2. When an EDL is opened as *Reference*, the color of that file name in the Resources Media folder is different in order to serve as a reminder that it is special.

3. A warning message is displayed in a popup window when you load a *File by Reference* that reads `Other projects can change this project and this can become a broken link`. Although you can check the warning box to never see this warning again, you would be well advised to not do so. It is a great reminder of consequences and you will not want to be cavalier about the warning. Instead just use the X to dismiss the warning.

Here is a step by step example of how you can use *File by Reference*:

1. Start up CINELERRA-GG and use the Title plugin to create a new credits file. Save as credits.xml.

2. Start a New project and then load an existing master project to the timeline.

3. Load the credits file you created in step 1 with a Load Strategy of Create Resources Only and with EDL Strategy as *Reference*.

4. Note the color change in the credits.xml filename and the reference comment in the Resources Media folder.

5. Drag the credits file to an empty spot on the timeline. Save this new master project and quit.

6. Start CINELERRA-GG up again. Load credits.xml and make a change to the Title and save again.

7. Exit CINELERRA-GG; restart CINELERRA-GG; load your master project and now you will automatically see on the timeline the changes you just made in the previous step.

6.5 Trimming

With some edits on the timeline it is possible to do trimming. By trimming you shrink or grow the edit boundaries (*head* or *tail*) by dragging them. In drag and drop mode or cut and paste mode, move the cursor over an edit boundary until it changes shape. The drag handle shows as a left or right facing fat arrow when you cursor near the clip start or end. If the cursor faces left, the dragging operation

affects the beginning of the edit. If the cursor faces right, the dragging operation affects the end of the edit.

The effect of each drag operation not only depends on the behavior button but whether the beginning or end of the edit is being dragged. When you release the mouse button, the trimming operation is performed.

For all file formats, other than still images, the extent of the trimming operation is limited to the source file length. Attempting to drag the start of the edit beyond the start of the source, limits it to the source start. In all trimming operations, all edits which start on the same position as the cursor when the drag operation begins are affected. You have to disarm tracks in order to prevent edits from being affected.

You have 6 different choices of which mouse button to use for specific types of editing while using the drag handle. You change the drag handle mouse effects by using the `Settings` → `Preferences` → `Interface` tab and modifying the Editing section as shown in the next figure 6.3. The drag handle affects not only the clip you are working on but also frequently the entire duration of all clips on the timeline.

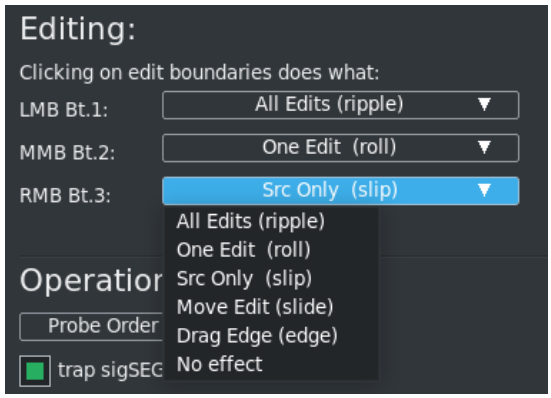


Figure 6.3: *Default choices for mouse: Ripple for LMB; Roll for MMB; Slip for RMB*

A description of the fundamental/common terminology for choices follows.

All Edits (ripple) shorten or lengthen the start or end of a single piece of media while moving all media to the right of that clip up or down on the timeline correspondingly. Timeline duration is modified. In a drag *All Edits* operation, the beginning of the edit either cuts data from the edit if you move it forward or pastes new data from before the edit if you move it backward. The end of the edit pastes data into the edit if you move it forward or cuts data from the end of the edit if you move it backward. All the following edits shift. If you drag the end of the edit past the start of the edit, the edit is deleted.

One Edit (roll) move the in and out point of a single clip without changing the timeline duration. In a drag *One Edit* operation, nothing is cut or pasted. If you move the beginning or end of the edit forward, the source reference in the edit shifts forward. If you move the beginning or end of the edit backward, the source reference shifts backward. The edit remains in the same spot in the timeline but the source shifts.

Src Only (slip) move the in and out point of a single clip without changing the timeline duration. In a drag *Src Only* operation, nothing is cut or pasted. If you move the beginning or end of the edit forward, the source reference in the edit shifts forward. If you move the beginning or end of the edit backward, the source reference shifts backward. The edit remains in the same spot in the timeline but the source shifts.

Slide a single clip is moved but retains its current in and out point; however the out point of the clip to the left changes and the in point of the clip to the right also changes. Timeline duration remains the same.

Edge Left/Right moves the edge of the clips.

No effect no changes are made. You might want to use this choice to prevent accidental movements.

NOTE: you cannot do Trim operations on *Hard Edges*, which are produced by a cut operation ("x"). To create a cut useful for trimming you must select an interval (at the limit of one frame) to be muted ("m"); now the edges of the edits will be trimmable because they are not *Hard Edges*. See 5.4, *Split - blade cut and hard edges*.

The next table displays the options and results with the Key Table here first.

```
s = src media start
p = proj position
l = length
c = cut distance
rest == p+=c: for rest of clips
01 = flags edits_moved, rest_moved
```

	Drag Left	**Drag Right**	
curr s += c, l -= c; + rest	←	→	*rest*
abc12345xyz **Ripple** left edge 11 →	abc012345xyz	abc2345xyz	
curr l += c; + rest	←	→	*rest*

abc12345xyz	**Ripple** right edge 01 →	abc1234xyz	abc123456xyz	
prev l += c; curr ps+= c, l -= c		←	→	
abc12345xyz	**Roll** left edge 00 →	ab012345xyz	abcd2345xyz	
curr l += c; next ps+= c, l -= c		←	→	
abc12345xyz	**Roll** right edge 00 →	abc1234wxyz	abc123456yz	
s -= c		←	→	
abc12345xyz	**Slip** left edge 10 →	abc23456xyz	abc01234xyz	
s -= c		←	→	
abc12345xyz	**Slip** right edge 10 →	abc23456xyz	abc01234xyz	
prev l += c; curr p+= c; next ps += c, l -= c		←	→	
abc12345xyz	**Slide** left edge 10 →	ab012345wxyz	abcd12345yz	
prev l += c; curr p+= c; next ps += c, l -= c		←	→	
abc12345xyz	**Slide** right edge 10 →	ab12345wxyz	abcd12345yz	
curr s -+= c, l += c; + rest		←	→	*rest*
abc12345xyz	**Edge** left edge 11 →	abc2345xyz	abc0123456xyz	
curr l -+= c; + rest		←	→	*rest*
abc12345xyz	**Edge** right edge 01 →	abc1234xyz	abc123456xyz	

Next, a more immediate and colorful view shows these trimming options (figure 6.4).

How to do a J-cut or L-cut A J-cut is a split edit film editing technique in which the audio from a following scene overlaps the picture from the preceding scene, so that the audio portion of the later scene starts playing before its picture as a lead-in to the visual cut. An L-cut is a different split edit film editing technique in which the audio from preceding scene overlaps the picture from the following scene, so that the audio cuts after the picture, and continues playing over the beginning of the next scene (figure 6.5). To do either a J-cut or an L-cut, you first shorten the first or second video a little. Then you block the audio tracks from changing by disarming the appropriate tracks. Finally use *One Edit (roll)* the cutting edge off the videos. Moving to the right creates a J-cut and moving to the left creates an L-cut.

6.5.1 Split View in Compositor Using the Drag Handle with Trim

The Trim Feature using the drag handle provides some good ways to view your video while editing. The playback position in the compositor is updated live and

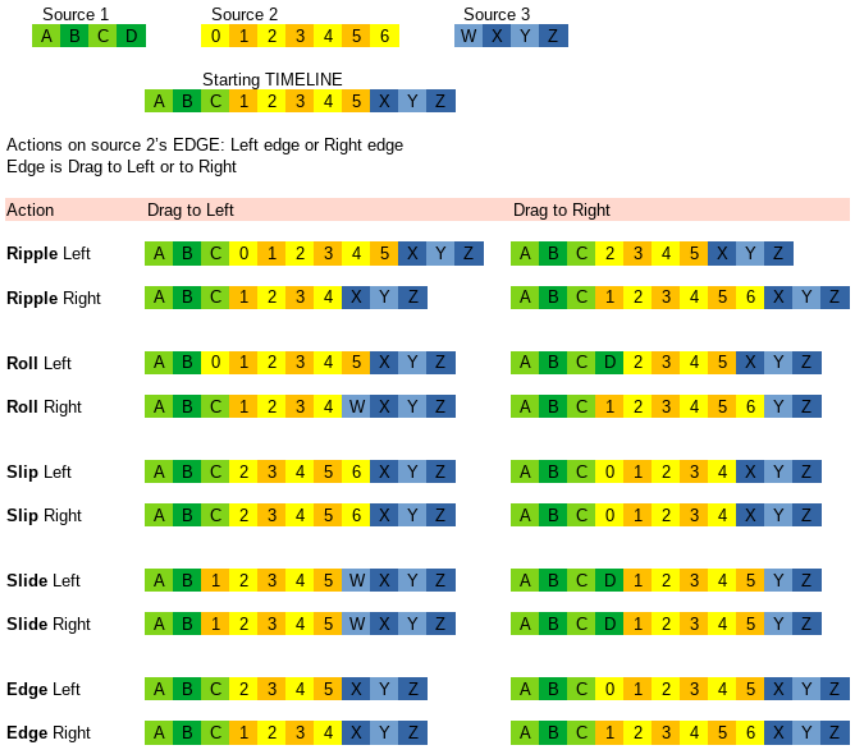


Figure 6.4: *The 5 types of Trim: note the different lengths of the results.*

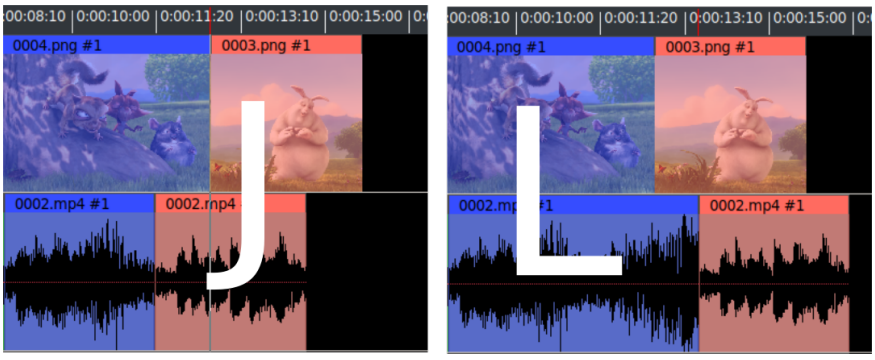


Figure 6.5: *J-cut to left and L-cut to right*

the view in the compositor can be split so that in the left half of the compositor you can see the last frame of the left clip and in the right half the first frame of the right clip. Dragging edits can not be extended past the beginning or the end.

First familiarize yourself with button operation; check your setup by executing the following step. In the `Settings` → `Preferences` → `Interface` tab, Editing section, clicking on the edit boundaries can be set for LMB (Button 1), MMB (Button 2), RMB (Button 3) as one of the following:

Ripple; *Roll*; *Slip*; *Slide*; *Edge* or *No effect*

Now to use this feature, create a track with edits that have trims on the left and/or the right. The edit boundary can be modified using *drag handles* at the boundary between the edits (figure 6.6).

Figure 6.6: *Split compositor screen showing the result of the Trim feature*

Left Mouse Button (LMB) usage: If you grab the edit handle from the right side, you will see a left arrow and dragging the boundary will modify the right edit playback starting time. If you grab the edit handle from the left side, you will see a right arrow and dragging the boundary will modify the left edit playback ending time. In both cases, the composer will show the edit endpoint of the changed edit.

Shift LMB usage: The effect on the edits is the same as described above, but the composer will show a split screen of the left and right edits as they appear at

the drag handle boundary. Dragging will only change one of the two images, since only one edit is being changed.

Middle Mouse Button (MMB) usage: Both the left and the right edit ending/starting times are updated. The image shown in the compositor will be drawn from the side of the drag grab, that is the left if it is grabbed from the left, and the right if it is grabbed from the right.

Shift MMB usage: The effect on the edits is the same as described above, but the composer will show a split screen of the left and right edits as they appear at the drag handle boundary. Dragging will change both of the two images, since both edits are being changed.

Right Mouse Button (RMB) usage: The start/end point of the current edit is moved, but the edit length is unchanged only one image changes since only one edit endpoint is view is updated.

Shift RMB usage: The effect on the edits is the same as described above, but the composer will show a split screen of the left and right edits as they appear at the drag handle boundary. Dragging will only change one of the two images, since only one edit is being changed.

6.6 Nesting clips and assets

6.6.1 Nested Assets

A nested asset is an EDL session that embeds an existing EDL session, all tracks, all plugins, editing, and effects into a media object that appears as one audio/video media object, no plugins, editing, or effects. It is as if the existing EDL was rendered, and loaded in its place. This has several interesting side effects. First, you don't have to render the entire media file to see any portion. Second, it requires no rendering compute time or storage. Third, it changes the precedence of the composer so that you get more control over the projection and automation, so that the results can be sent into another rendering step, not simply part of the current stack. It groups the plugin stack in much the same way that an arithmetic expression is grouped by parenthesis.

The EDL session and the rendered output are visually equivalent. Nested assets allow for complex grouping and stacking of effects, and makes media access much more flexible. This feature can be used recursively, that is, any number of sessions may be stacked and referenced as an asset, as long as all of the rendering resources are available. Nested assets are added to the timeline by using the pulldown `File → Load files...` on the main menu and selecting the *Insertion strategy* of *Nest asset*. The file will be pasted into the timeline over the current selection or at the insertion point.

It is somewhat important to note that nested assets and nested clips will have index files automatically created. These index files can start to clutter up your `$HOME/.bcast5` directory with files named `Nested_###.idx` and you may want to periodically delete any index files which are no longer in use.

6.6.2 Nested Clips

It is also possible to create *clips* and convert them to *nested edl*. This is done by first creating a clip using the standard cut, clipboard, paste, and/or edit panel buttons. Now, using the resources *clip* folder, select a clip to be nested, and use the right mouse button to select a clip. This activates the clip popup menu. Select the *Nest to media* menu item, and the clip will be converted to a *Nested: Clip* and put in Media folder. Conversely, you can select a *Nested: Clip*, use the *EDL to clip* menu item, and the clip will be reverted to a *Clip*. This works similarly to the group / un-group editing features of many graphic design editing programs, but in this case the groups are rendered compositions (figure 6.7).

Nested clips can be proxied and when they are, the resulting files are placed in the user's `$HOME/Videos` directory by default. This can be modified by changing

`Settings` → `Preferences` → `Interface` tab, Nested Proxy Path.

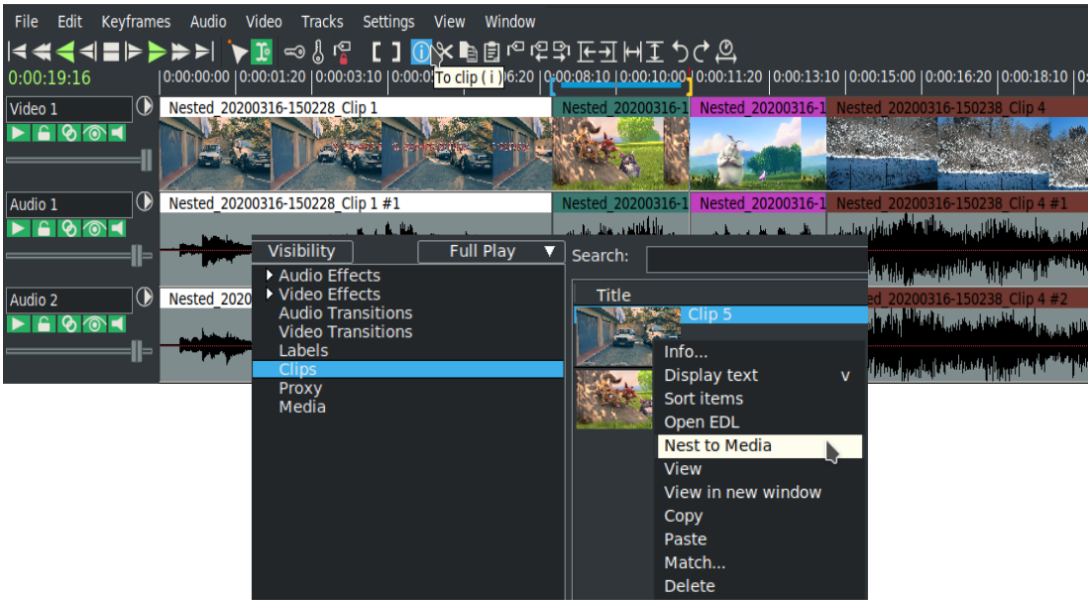


Figure 6.7: *Nested clips in Timeline and Resources window*

Notes: creating nested clips is a complicated internal process for CINELERRA-GG, especially for managing audio tracks. Therefore you need to make sure you have your project set up as you expect before you start nesting with the `Nest to Media` command. Below are several things to remember.

- The creation of the nested clip is based on the settings in `Settings` → `Format`. Be sure that the number of audio tracks and channels is the same as we want to have in the nested clip. In addition, it is better to make each audio track independent from the others (which could be linked in more than one channel) using the option: `Audio` → `Map 1:1`
- You can do any editing on the nested clip once it is loaded to the timeline. For example we can start an OpenEDL session to make the changes. Once out of the OpenEDL session it may be that the changes made are not visible in the original nested clip. You can fix this by doing RMB on the nested clip in the *Resources/Media* window and choosing `Rebuild index`. This process may take some time depending on the size of the media; because of the additional time required to do this, it is not done automatically in order to allow the user to make the decision as to when they are done editing enough and need to see the exact results in order to save time.
- Because of the way the timeline thumbnails are displayed, after editing using OpenEDL of a nested clip, the thumbnails of the video track may be redrawn and may look different and out of sync with the way they did originally with respect to the audio waveforms. This is simply a result of the redrawing of the video track which is independent from the redrawing of the audio tracks. The playback is still synchronized.

Usage Examples of Nested Clips

Example 1: You want to make a flashback/rewind at the end of your video that represents a quick summary of the entire video in black and white. On he timeline, you have 60 seconds of edits with clips, cuts, zoom in, zoom out and any other edits. Now you want to get this 60 seconds *compressed* to 10 seconds, play in reverse, and in black and white at the end of your video. You would copy the 60 seconds in a clip, nest the clip in the Clip folder of the Resources window and drag it to the timeline. You will see only a clean clip without all of the edits that were used to create it because nesting display a clip without having to actually use the Render menu. Now you can add a Reverse effect, Color3way plugin for black and white, and use the Speed auto to get the 60 seconds down to only 10 seconds.

Example 2: You are working on a complex project with a team in a separate location. You create some sub projects, i.e. sequences, that you or the team will use in the Master project to merge the sequences in the right order and to make the final color correction steps.

In each of the examples you can see the benefit of nesting to create clean looking timelines because of the automatic rendering capability of nesting.

6.7 Multi-Camera / Mixer

Use the Mixer Viewer to see multiple media playing simultaneously in re-sizable mini-viewers. This can be used in various ways and is useful to edit videos shot by multiple cameras from different viewpoints that were simultaneously recorded in order to create a single good video. Everything will have to be initially synced so you can decide which one of the camera angles is best suited at any time. Mixer mode does not support audio only tracks but there is a workaround described at FAQ, Known Problems and Workarounds.

The number of cameras/mixers you can have is generally limited to the available resources on your computer. Currently, the number of File Descriptors available in the OS limits cameras to about 50. If you have many *mixer viewers* you will probably want to use proxy mode whenever possible. Also, in the `Settings` → `Playback` `A` tab *Video Out* section, uncheck *play every frame* and choosing a Video Driver of *X11* with *use direct X11 render if possible* checked, will provide better performance.

Figure 6.8 shows 9 media sources in the left corner, the composed video in the right corner, the timeline with the top video track with pieces of the 9 overwrites, and the choice in Resources of Mixed.

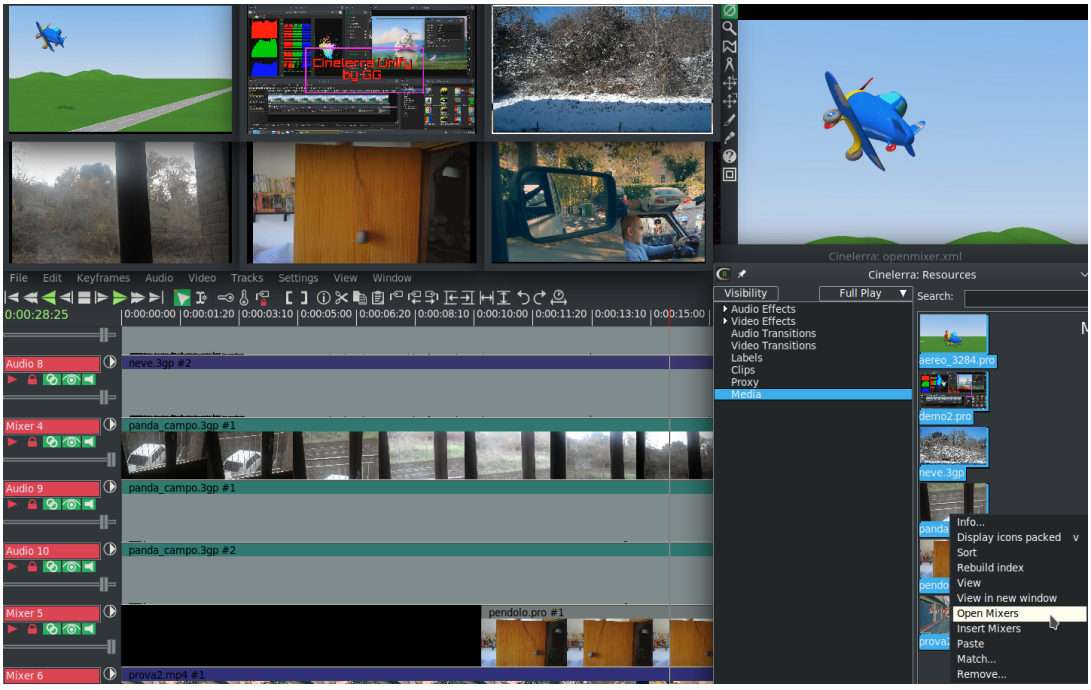


Figure 6.8: *Using Mixer capability in Cin for multiple cameras*

Easiest Method to Getting Started

1. This method assumes all of your media or cameras are aligned the way you want them already.
2. From the *File* pulldown, create a *New project* with the desired format for Audio and Video output (or you can just use the default).
3. `File` → `Load` the media files you want to work with using *Create new resources only*.
4. In the Resources window, with the Media folder, highlight the list of media you want to *Mix*. This is done using a ctrl or shift mouse button press as you would in a standard listbox selection.
5. Right click the mouse on the media selection and choose *open mixers* . This opens multiple mixer viewer windows, one for each media item that was highlighted. You can do them 1 at a time instead. This also adds the source media tracks to the main window.
6. Now use the timeline to play and you will see all viewers/cameras playing. Stop when you get to the end of the *good* camera playback.
7. Simply double click the *good* mixer viewer and from where you first started playing to the playback insertion pointer is the source section, which will be pasted in the destination video/audio tracks at the top of the new project.
8. Repeat steps 6-7. Start playing again, stop when you want, double click the desired mini-viewer!

Some Hints and Caveats

- You can easily overwrite a section of the new track by *selecting* a section on the timeline, then double click on one of the mini-viewers to overwrite/replace that section.
- If you edit the output tracks, it only edits output tracks, and the input tracks may no longer be lined up.
- You can add a silent section by selecting past a section and start overwriting that section from then on.
- If you use the cursor hairline to create the selection endpoint, it must be past the end of the destination.
- The compositor shows composed media. This is the media that will be rendered.
- The program always uses overwrite as the paste operation.

- Use the timeline edit handles to move the start and end points of that section.
- Only middle mouse drag handle operations should be used normally.
- Other drags will displace the media source/destination timeline correspondence.
- To re-tile the mixer windows after you have resized and moved them around, you can use the Window pulldown of *Tile mixers* or the shortcut of Alt-t.

But, I want to use only the first set of audio tracks...

There are many cases where you may want to compose using media from several different tracks while using the the same audio tracks as associated from a specific viewer. Since mixer source tracks can be updated any time by using a mixer toggle, this makes it possible to do this.

Procedure to update the mixer audio source track list:

1. Single click to highlight the mixer window you want to re-associate to the audio track.
2. In that audio track's patchbay click the expand toggle, the arrow on the right side.
3. In the expanded pane that appears, there is another arrow on the left side. This icon has the tooltip *Mixer*. Click this and because in step #1 you highlighted the mixer window, it will now be toggled on. Once you click the mixer icon it will then point up.
4. Now, disassociate any audio that is unwanted by expanding its patchbay and toggling off the mixer.

Expert Usage

When you double click a mixer viewer window, it operates an *overwrite* paste operation. This moves *src* (source) track edits to *dst* (destination) track edits over the same selected timeline region.

- *Src tracks* should be not playable and not armed in the main window patchbay gui.
- *Dst tracks* should be playable and armed in the main window patchbay gui.

Each mixer viewer maintains a list of the tracks which will be used as src. This list is made visible selecting the window with the left mouse button. When the mixer viewer is selected, a highlight is drawn around the media image. All track patchbay *mixer* toggles are updated to reflect the src tracks included in the selected viewer src track list. The *play track* patchbay toggles can be used to manage the list.

- *Turning on* a toggle (pointing up) includes the track in the src track list.
- *Turning off* a toggle (pointing right) removes the track from the src track list.

New Mixer viewers can be created using the main menu `Window → Mixer Viewer`, or with a shortcut of `Shift-M`. When a new viewer is created, the currently enabled patchbay *mixer* toggles are used to create the viewer source track list. The toggles are cleared after the window is created. This is to improve the work flow. Use the following list of steps to create individual mixer viewers.

To create a list of mixer viewers:

1. Setup the session `settings → format`, width, height, frame rate, color model, aspect ratio.
2. Create dst tracks using the a/v track pulldowns (or use shortcuts "t" / "T"), armed and playable.
3. Append src tracks using `file → open → append tracks`, or the resource window using pasting.
4. Using the track patchbay, disarm editing and disable playback of the audio/video src tracks.
5. Using the track patchbay, mark the new tracks as *mixer* source to be added to the viewer.
6. Create a mixer viewer using the main menu pulldown, or the shift "M" shortcut.
7. Repeat steps 3 — 6 for each mixer viewer needed for the session editing.

When you single click a mixer window, it becomes selected and highlighted and all of the patchbay mixer toggles are updated to reflect the state of the viewer's src tracks. Tracks that will be src are shown as enabled. If you change a toggle, the src tracks for the selected window will be modified. This means you can associate or dis-associate any media track to any mixer window.

When you double click a mixer window, an overwrite paste is invoked. The mixer viewer's src tracks are overwritten to the dst tracks. The timeline region for both the source and destination are the same for the overwrite paste function. The selection region is used if it is active. If the selection is empty, that is it is a hairline, the selection region is from the end of the destination playable edits to the selection cursor hairline. The hairline must be past the end of the playable edits on the destination tracks.

The mixer viewer configuration is saved with the session data. When a saved session is loaded in *replace project* or *replace project and concatenate tracks*, the mixer viewer will be reopened.

Using Proxy with *Open Mixers*

The best way to use proxy with your multiple cameras is to follow the steps below:

1. Load media with insertion strategy of *create resources only*.
2. Highlight the media in the Resources window and right click on this to choose *open mixers*.
3. Use the `Settings` → `Proxy settings...` to bring up the proxy menu.
4. Choose the size and other options you want and click the checkmark OK. If you choose the option *Beep when done* you will hear a short beep if all media is already proxied or a longer beep when all proxies have been created.
5. When your editing is complete, use *Settings* pulldown and proxy to *original size*.

Instead of Open Mixers, you can Insert Mixers with new tracks at the timeline insertion point.

Options available in the *Mixer Windows*

There are several options you will see in each Mixer viewer that help with using them. These are shown in figure 6.9. To use these, RMB in the desired mixer window, and choose one of the options as described here.

1. Fullscreen / Windowed - will bring up a fullscreen display of that window or revert to the original size.
2. Resize Window - allows for resizing the window to a choice of different sizes.

Figure 6.9: *A Mixer window showing available options with the white X designating no playing*

3. Tile Mixers - makes it easy to get all of the mixers nicely tiled to a standard size. This is also available in the *Window* pulldown.

4. Playable - enabled by default so that you will see a checkmark next to it in the popup. The benefit of making a mixer window not playable is to save cpu time. When a specific mixer window is unchecked, so not Playable, a white cross appears in the upper left hand corner of the mixer. To switch back to playing again, RMB and choose Playable and this will turn playing back on. The program automatically updates the mixer image when the timeline cursor stops or when it is moved to a different spot even when play is disabled. You can see the white cross designating not playable in figure 6.9. Statistics that show the savings are as follows on a 16 cpu laptop:

```
5 Mixers all playing -- cpu usage = 1275
1 only playing       -- cpu usage =  405
2 only playing       -- cpu usage =  800
```

Mixers pulldown

From Program windows menu we can access the `windows` → `mixers...` there are several entries available (see figure 6.10):

Mixer Viewer: (Shift-M) opens a new empty *Mixer 1* window, which will then be associated with a track. We can open as many Mixers as we like.

Drag Tile mixers: (Alt-t) the default size of a mixer window is related to the std size of the *Viewer* window. If we want to modify it at our convenience, we can build a window of the desired size with the *Drag Tile mixers* function

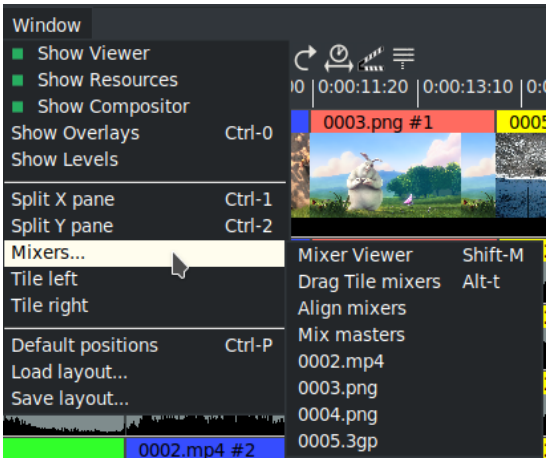


Figure 6.10: *Mixers... pulldown*

and then automatically order our mixers within it with the right click of the mouse.

Align mixers: see Audio/Video sync via Waveform

Mix masters: this feature makes it very easy to get into the multi-camera mixer mode after tracks have already been set up and edited. Before this addition, you could only *Open Mixers* from original media assets in the *Resources* window.

Entries: If the mixers of several sources have been opened, the entries of each assets will appear, useful for recalling a specific mixer window.

6.7.1 Recover Mixer Windows

It is a hazard that you might accidentally *undo* (`z or Ctrl-z`) too far and lose your mixer windows. Here are the steps to recover. It is recommended that you make a backup of your project before performing the recovery steps just in case there are other problems.

In the patchbay box to the left of the main timeline, there is a right pointing arrow on the right side. This is called the *Expander* (figure 6.11). When you click on each expander, another line appears below that expander arrow and the timeline track height is slightly increased. If you Shift-click on a single expander, the patchbay will expand for all of the tracks.

1. Expand all of the patchbay lines, either one by one, or Shift-click on one to do them all. This is so you can see the *mixer* right pointing arrow on the second expanded line.

Figure 6.11: *Mixer patchbay*

2. Use the *Window* pulldown and choose *Mixer Viewer* to bring up a new mixer window. Now you will be making an association between the mixer viewer and the track's video.

3. Click on the new mixer window to make sure it is highlighted with a white border. This designates it as the *in use* mixer viewer.

4. Set your play to the beginning of the video using the *Home* key or *Home* transport button.

5. In the patchbay for a video track click on the *mixer* arrow on the expanded 2^{nd} line which is a right facing arrow. Now the arrow will point up. If there are audio tracks with that video, click on each of its audio tracks *mixer* arrow until they point up also.

6. Next move your insertion pointer on the timeline where there is video. Some of the time this just helps so that the new mixer viewer window gets redrawn and you can see that the images appear; but the image may not appear until the program does a redraw later. Now the mixer viewer should be correctly associated. Note if you have large video, give it some time to update. You may have to click on the mixer viewer window if the image does not show. You can always start over with that mixer if you encountered any problems.

7. Click the arrows that are pointing up in that video and its audio so they go back to pointing right. That mixer viewer is complete so you need to do this to make sure the *mixer* arrows are off.

Repeat steps 2 through 7 for each of the mixer viewers you need going down the patchbay starting on step 2 first with Mixer 1, then 2 to 7 steps for Mixer 2, then again run 2 to 7 for Mixer 3 and so on.

Sometimes the association does not stick initially. If not, highlight the mixer viewer with the problem, change the mixer arrows to point up, and reassociate.

6.8 Audio/Video sync using Waveforms/Timecodes

6.8.1 Audio/Video sync via Waveform

Multi-camera footage of a single event can have various shots starting and ending at different times. So when the footage start times are different, you can use the mixer audio to synchronize the clips on the timeline. The program algorithm attempts to find and align automatically the waveforms of the media.

Synchronizing multiple camera videos based on audio tracks can be done with CINELERRA-GG easily enough with the `Window → Mixers... → Align mixers` option. Align mixers brings up a window displaying your mixers, the currently selected Master Track, and a list of all of the Audio Tracks (figure 6.12). There is a limit of 32 audio tracks per each mixer (that should be enough!)

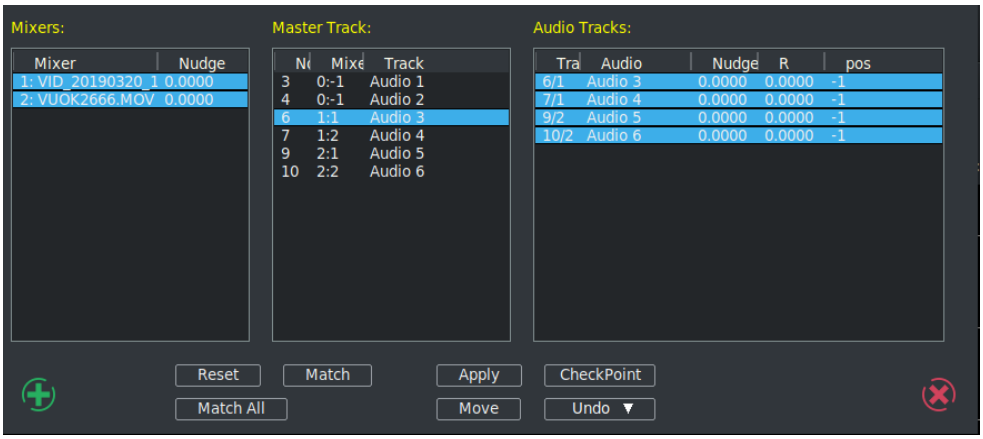


Figure 6.12: *Align Mixers window*

Two different methods of aligning the audio for mixers are available. They are most easily referred to by the button that is pressed – *Match* and *Match All*. There are also 2 methods of activating the alignment – *Apply* and *Move*.

Match consists usually of the next set of steps to take advantage of this feature:

1. Load your camera media with insertion strategy of *resources only*

2. Highlight in the Resources window, all of the media you want to mix.
3. Right mouse button on one of the media and click on Open Mixers; all mixer windows come up.
4. To arm the created mixer tracks and related audio tracks. They are automatically placed on the timeline disarmed. This is right for multi-cam but not for audio sync.
5. Provide a small target audio pattern on the Master Track for syncing by marking with the In/Out points ([and]). NOTE: When choosing between the audio tracks to align the one that will have to work as Master, we must rigorously use the one with the right-most synchronization peak. This is because the Master track will be started at the beginning of the timeline. If during alignment the other tracks should move further to the left than the Master, they will be truncated at 00:00:00:000, finding no more space available, and will be shortened with loss of footage for the next multi-cam
6. Make a selection on the timeline in which to look for the pattern. Left mouse click, then drag select and highlight a search time domain.
7. Use the `Window` → `Mixers...` → `Align mixers` to bring up its dialog window.
8. Highlight in the first listbox, the Mixer number you want to align. Click on *Match*. This will take a few seconds so watch the rendering time percentage on the lower right hand side zoom panel. The buttons will be ghosted out until finished. Now note the changed values in the Audio Tracks listbox.
9. If you are satisfied with the calculated Nudge values – that is they are very close to 1.0 – in the Audio Tracks listbox and the audio track selected as the Master Track in the Master Track listbox, hit the *Apply* button.
10. If you plan on performing more alignment tasks, click on Checkpoint so you can go back to a previous step in case you make a mistake.
11. Last, click on the OK green checkmark or to cancel click on the red X. Or just close the gui.
12. To disarm the mixer tracks and related audio tracks for the next multi-cam operations.

Reset is used to start over with the current session data, not an undo. This means you can use the match repeatedly to refine alignments. All of the Audio Tracks listbox values will be reset.

Checkpoint provides a method to create checkpoints that save the current state. This is especially helpful while learning or doing more complicated operations where you might make a mistake or do not like the results and need to get back to a previous state.

Undo is used to put the media back to a previous state on the timeline. If you choose *start over* the session will reload with the original, before any changes were applied. You can also go back to any of your previous checkpoints that you created earlier which are listed there, such as *chkpt 1*.

Match All is used when you have several mixers, instead of only picking 1 to match, it picks the best match for EACH of the mixer tracks based on a single master track. So when you hit Apply, each track might move differently. You do not set In/Out points but you have to make a selection within which to match.

Apply button will apply the nudges that were generated during the Match or Match All execution (figure 6.13).

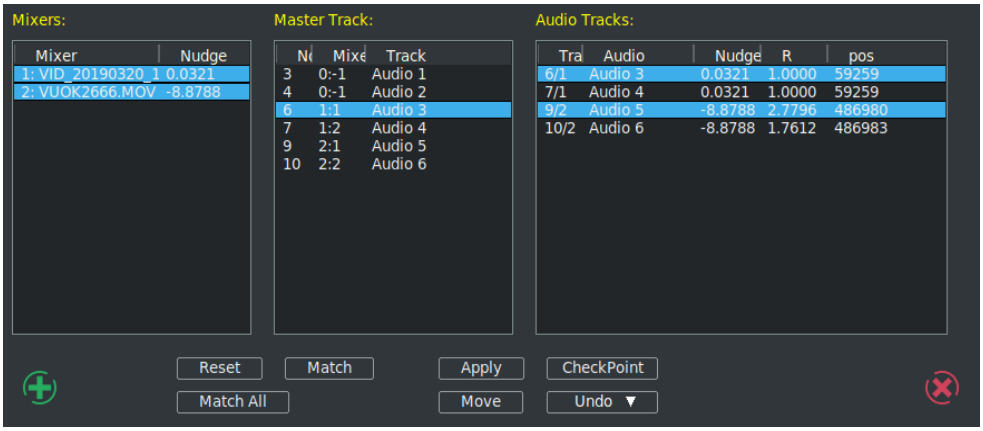


Figure 6.13: *Aligned mixer window after "Match". Note the Nudge amounts above.*

Move is very handy when you are using mixers, if you have an edit somewhere that you need to fix specifically without moving any of the other track pieces. In this case you have to select a section (like you do a group, but do not make a group), generate a match, and then you can just *Move* that set only – everything else stays where it is at its current location (figure 6.14).

More detailed information follows about how this all works and the information in the dialog window. It is important to know that the result of the calculation is *best match* but you can still override the selections if you decide there is a better one. The dialog window is split into 3 sections:

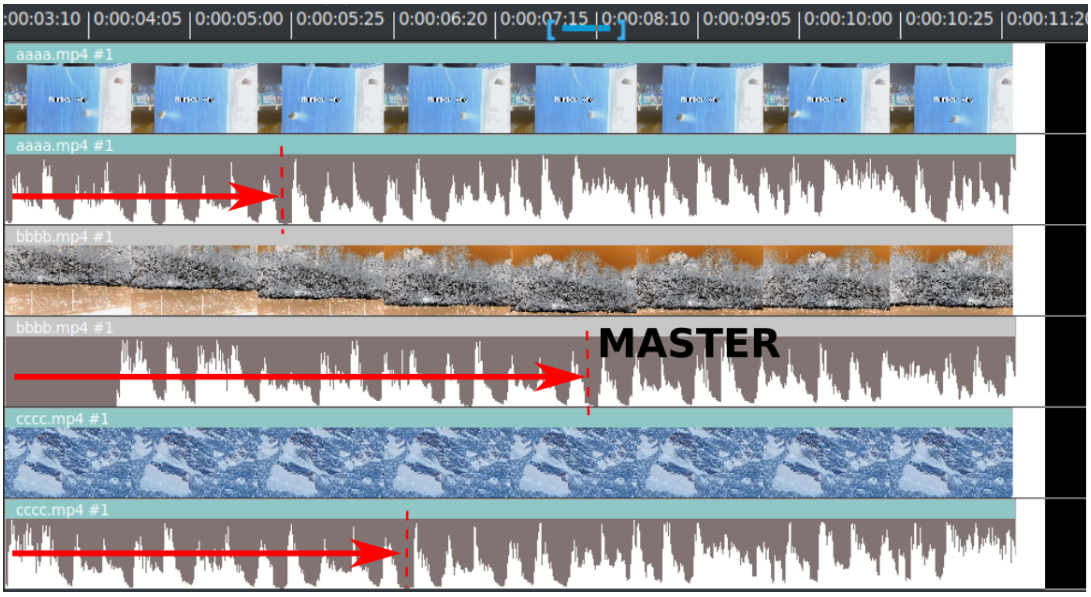


Figure 6.14: *Match setup for aligning by audio. Note that [ ] are set over a sample waveform highlighted selection that includes that.*

1. *Mixers* lists the mixers that are active by highlighting them all initially. You can decide that you do not want 1 or more mixers to be used in the correlation calculation by un-highlighting the one(s) that should not be used. In some cases you have to have at least 2 in order to align audio.

2. *Master Track* lists each of the audio tracks currently loaded for all of the mixers. You can decide to highlight a different audio track to be used as the master for correlation, but only 1 can be used. Remember to use the track with the right-most synchronization peak as the Master.

3. *Audio Tracks* lists each of the mixer audio tracks. Again, you can highlight a different set of which mixer audio tracks that you want to use for the waveform correlation. Any audio tracks that are not highlighted, that is *turned off*, will not be considered in the correlation calculation.

The corresponding input position is determined by track input correlation.

The letter "R" in the Audio Tracks listbox represents the correlation value. $R = 1.0$ designates that if both the pattern and the matching section were in the highlighted area, they are completely correlated – this is a good self-test to check. $Nudge = 0.0$ means just that.

The header *pos* stands for the timeline position. When the *Apply* button is pressed, only the Mixers listbox is relevant at that time.

The Mixer with the master track generally does not move, everything else will be lined up (figure 6.15).

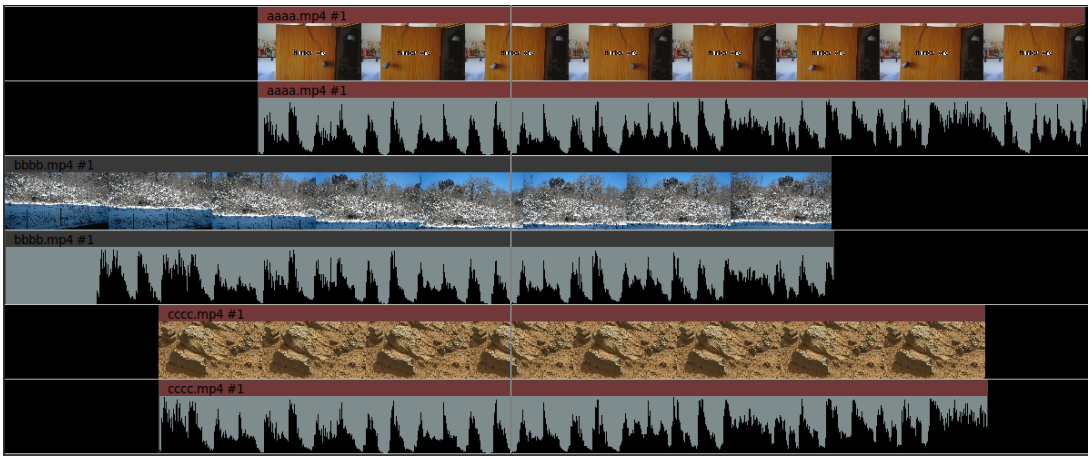


Figure 6.15: *An audio Match is complete. Note the waveform is aligned.*

Match All option basically consists of the following steps:

1. Highlight the Mixer to use in the Mixer listbox.
2. Highlight the Master Track you want to use in the Master Track listbox.
3. On the timeline, mark your selection on the Master Track.
4. Click on the *Match All* button.
5. Note the nudge values to see if they make sense, and if so, press *Apply*.

Match option basic steps (just for comparison with Match All):

1. Set the In/Out points [ ] of the target.
2. On the timeline, mark your selection.
3. Click on the `Match` button.
4. Note the nudge values to see if they make sense, then press *Apply* (or *Move* when doing a group).

6.8.2 Align Timecodes

Align Timecodes is especially useful in the case where you create video with multiple cameras capable of recording a timecode in the metadata of each file. Let's say we have recorded three videos at the same time at a concert with each camera set up at unique positions and at different angles. All of the cameras start recording at various times but were synchronized with the same master clock (*Timecode Sync Generator* or *Jam Sync timecode*) so that the recordings are timestamped with that synchronized time. For example, the 3 cameras have embedded metadata as follows: **camera 1** = 00:00:00:01 **camera 2** = 00:00:10:07 **camera 3** = 00:00:17:22

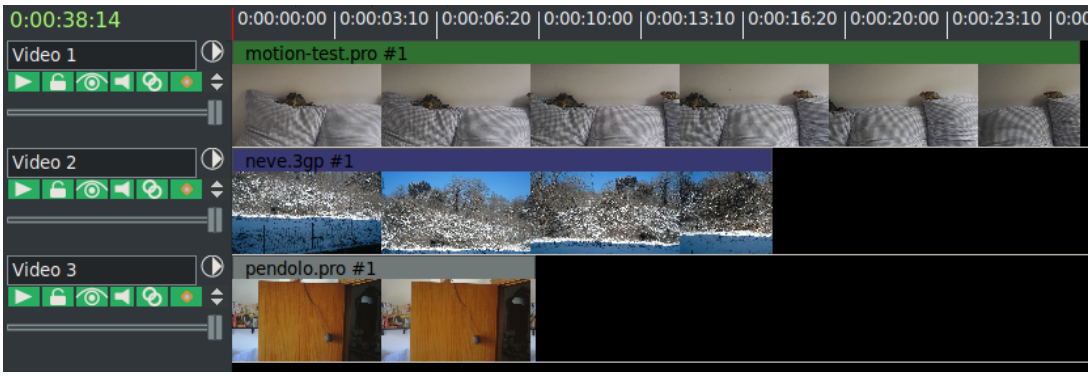


Figure 6.16: *Before sync*

Timecode synchronization is performed by the program when you choose the option *Align Timecodes* and works in 2 steps. The first automatic step is to locate the earliest timecode and the second step is to align the edits on the armed tracks using that time on the timeline.

When you load in the three files to different tracks, they'll be placed on a timeline that starts from 00 : 00 : 00 : 00 as usual. But, by middle mouse clicking (MMB) on any of the video tracks, you can view the timecode embedded at that point in that file.

To make it easy to align the videos onto the timeline manually, set the *origin point*, that is the leftmost time index, of the timebar to the timecode of the earliest video. For the 3 aforementioned cameras this would be camera 1 with a timeecode of 00 : 00 : 00 : 01. By using the *clapboard* button on the main timeline, you can set 00 : 00 : 00 : 01 as the start timecode. Then positioning the camera 1 video at the start of the timeline will line its frames up with the times on the timebar.

Similarly, using the middle mouse button to view the other 2 camera's timecodes, you can position them so their timecodes line up and now all three tracks will be

Figure 6.17: *Clapperboard button*

in sync. You can do this manually or simply do it automatically with the option under: `Tracks` → `Align Timecodes`. See figure 6.18

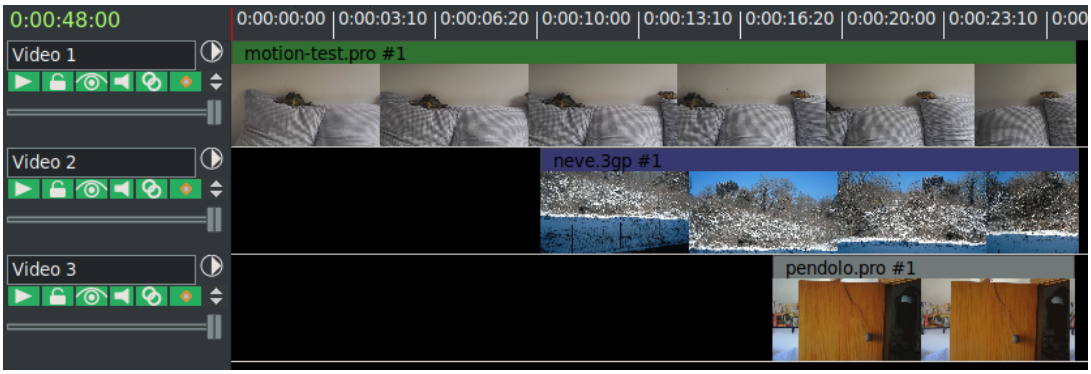


Figure 6.18: *After sync*

Timecode synchronization in CINELERRA-GG is based on the inclusion of an actual *timecode* saved in the media as can be seen using the `ffprobe` or `mediainfo` command line. Examples are shown here:

```
# ffprobe tut1.mov
Metadata:
handler_name: VideoHandler
timecode:  00:00:20:01

# mediainfo mtb.mp4
Time code of first frame: 11:05:49:41
Time code, striped:  Yes
Title:    GoPro AVC
```

Many newer cameras and media in a MOV container have valid timecodes often with SMPTE compliant start Timecode in Quicktime format, whereas others do not. Most low end cameras do not. When no timecode is present, various clock time values may be used to synchronize instead.

There are various types of timecodes and they are written in metadata in various ways. Because it is not clear which timecode in the file is the one to use or there is none, the precedence as described next is followed. The contexts, packets, streams, and data refer to the corresponding ffmpeg data or metadata objects:

1. if the format context has a timecode use: `fmt_ctx → timecode - 0`
2. if the layer/channel has a timecode use: `st → timecode - (start_time -nudge)`
3. find the first program stream with video stream, and if the video stream has a timecode use: `st → timecode - (start_time-nudge)`
4. find timecode in any stream use: `st → timecode`
5. read 100 packets, save `ofs=pkt.pts*st → time_base - st → nudge`:
   - decode frame for video stream of the first program
   - if `frame → timecode` has a timecode use: `frame → timecode - ofs`
   - if side_data has gop (group of pictures) timecode use: `gop → timecode - ofs`
   - if side_data has smpte timecode use: `smpte → timecode - ofs`
6. if the `filename/url` scans `*date_time.ext` use: `date_time` (yr, month, day, hr, min, sec, nanosecond)
7. if stat (display file status) works on the `filename/url` use: `mtime` (modification time)
8. finally, if none of the above are usable, then return `failure`

Operation of Align Timecodes includes the following options to help in your setup and review (figure 6.19):

- *Align Timecodes* under the *Tracks* pulldown
- *Clapperboard* icon on the main timeline to set a timecode.

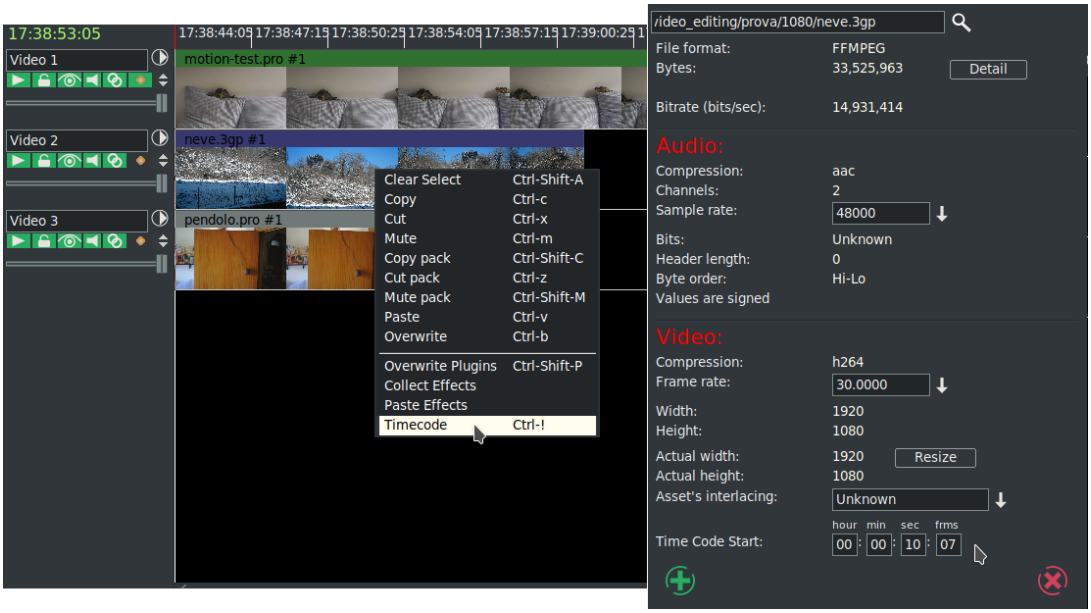


Figure 6.19: *Timecode via MMB on track and via Resources windows/Info*

- Timecode option added to timebar choices (in timecode $hh : mm : ss : ff$) with use of the session frame rate as a time standard.
- Time Code Start in the Resources window on the bottom of *Info* for the media if the timecode for that asset has been scanned and is known. You can scan the asset's Timecode by using the middle mouse button on its track which then displays the timecode for that on the timebar (`MMB → Timecode`). Or you can use the shortcut `Ctrl+!`, after selecting the edit/track on the timeline.

Notes

- *Align Timecodes* is a good first step to quickly sync media up either perfectly or roughly.
- Currently this is not known to function for syncing audio recorded on a separate audio device, because no test material was provided to ensure that this works.
- Disarmed tracks will not be affected by using *Align Timecodes*.
- No edits or sections will get cut off on the left or elsewhere, unless the moved edits overlap.
- You can add a timecode to video using an `ffmpeg` command if you want to align a particular video which has no timecode to ones that do. For example:

```
ffmpeg -i originalfile.mp4 -metadata timecode="14:36:08:29" -c copy newfile.mp4

# where "-i" is followed by the input filename
# where "-metadata timecode" is followed by the timecode key
# where "-c copy" just copies the video/audio to the following output filename
```

6.9 Subtitles

CINELERRA-GG can create subtitles directly in the timeline with the Subtitle tool. Subtitles are added by using the main window pulldown `File` → `Subtitle` (Alt-y) which brings up a window allowing you to type the filename of a previously generated text file containing the desired words/lines, the script. After entering the filename, click `Load` to read in your script. By creating a script file ahead of time, it lets you easily add dialog that was already written out and carefully edited for spelling and proper grammar.

The file must be plain text; a .srt or .sub can also be used, but only the text lines will be used and not the timecodes or comments. The format of the script/text input file has specific requirements as listed below:

- Lines can be any length but they will be broken up to fit according to some criteria below.
  - Running text used as script lines will be broken into multiple lines.
  - The target line length is 60 characters.
  - Punctuation may be flagged to create an early break.
  - Single carriage return ends an individual script line.
  - Double carriage return indicates the end of a entry and helps to keep track of where you are.
- The "=" sign in column 1 indicates a comment seen in the script text to assist you in location.
- An "*" at the beginning of the line is a comment and not a script line.
- *Whitespace* at either the beginning of a script line or the end will be removed.

Figure 6.20 shows the Subtitle window you will see.

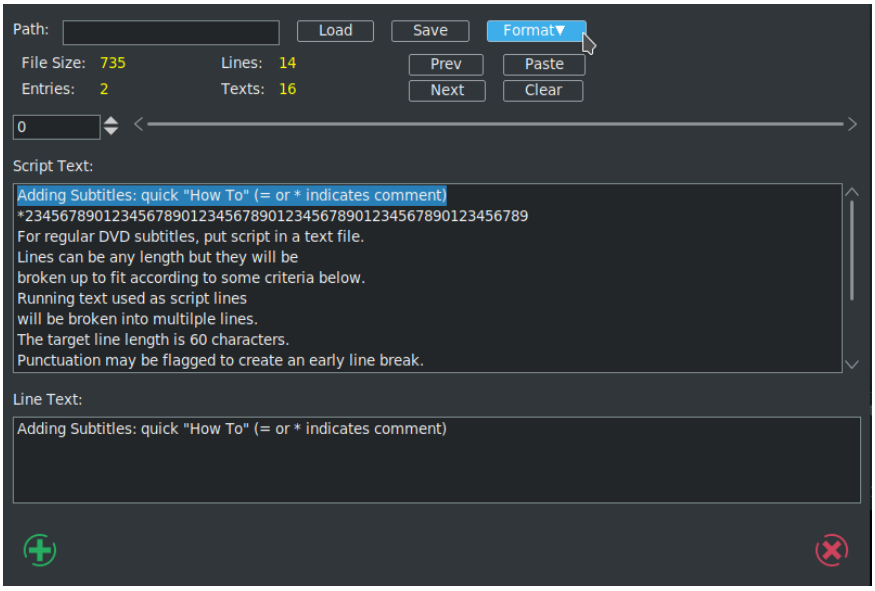


Figure 6.20: *Subtitle window*

To put the subtitles onto your media, first add a subtitle track via the pulldown `Tracks` → `Add subttl` (Shit-Y). In the Subtitle window, note that there are 2 major textboxes. There is the *Script Text* textbox showing the current entry of text from your input file and there is the *Line Text* textbox showing the currently active text. In your subtitle track, select a timeline region (in/out or drag select with hairline cursor/highlight or via labels or the *selection start/length/end time* textboxes in the Zoom Panel) to indicate the region where you want the active Line Text to be pasted. Then click the `Paste` button in the Subtitle window to paste the line onto the subtitle track. Silence will be added to the subtitle track in the places in the media where there are gaps.

Editing in the Line Text box can be used to change the active script line. By double clicking the timeline over the subtitle track, you can reselect the active script line. The subtitle text will be reloaded into the Line Text box and can be edited and re-pasted as the new active subtitle text. You can also highlight multiple lines in the Script Text box and paste them (using the usual window paste methodology) into the Line Text box. After pasting to the timeline, the Line Text box will be updated with the next script line. In addition, if you triple click a line in the *Script Text* box, it will automatically become the current line in the *Line Text* box.

When you are finished, before clicking on *Save*, you can specify the output format using the *Format* drop-down button. You can choose between the classic `.udvd` (micro DVD) and the more universally supported `.srt` (subrip) and `.sub` (sub-viewer). The next step is to provide a legitimate filname in the *Path* box; your current directory will be used if only a filename but no directory path is supplied.

The filename used will automatically have a "–" after it followed by the *track label* and then *udvd/srt/sub* extension added; any extension in the filename will be removed.. If you click OK before saving, the subtitle script position is saved with the session. This is convenient for continuing where you left off.

To reposition the script, use the slider or tumbler buttons:

Slider bar to move through the text entries quickly.
Prev or *Next* buttons to go to the previous or next script line.

Figure 6.21 shows what the pasted subtitle script looks like in a portion of the main window.

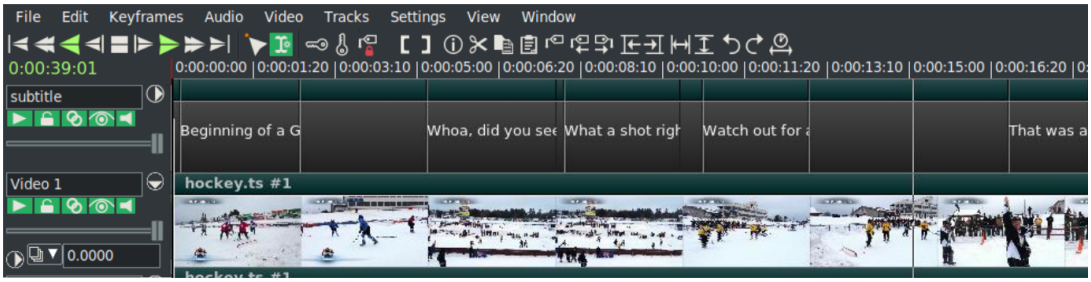


Figure 6.21: *Subtitles on timeline*

CHAPTER 7

Rendering

To configure the render according to your needs and to obtain the desired output, you will need to check on what your settings are for these three items:

1. **Timeline:** is the main starting point because, in addition to determining the range of frames that will be rendered, it is on the timeline that plugins and other editing features are applied, which will then be included in the rendered file.
2. **Set Format:** is used to configure the fps, aspect ratio and frame size of the project (= output). This can be found in the pulldown: `Settings` → `Format...` (`Shift + F`). See: Project and Media Attributes.
3. **Render dialog:** this is under: `File` → `Render...` (`Shift + R`). It contains all the rendering options that are documented in this chapter.

Rendering takes a section of the timeline, performs all the editing, effects and compositing, and creates a new media file. You can then delete all the source assets, play the rendered file, or bring it back into CINELERRA-GG for more editing. All rendering operations are based on a region of the timeline to be rendered. You need to define this region on the timeline. The rendering functions define the region based on a set of rules. When a region is highlighted or in/out points are set, the affected region is rendered. When no region is highlighted, everything after the insertion point is rendered. By positioning the insertion point at the beginning of a track and unsetting all in/out points, the entire track is rendered. But you also have the choice to render *one frame*. Reminder, CINELERRA-GG does not do remuxing without rendering - see Transcode.

7.1 Single File Rendering

Use the *File* pulldown and select Render to start the render dialog (figure 7.1). Then choose the desired parameters.

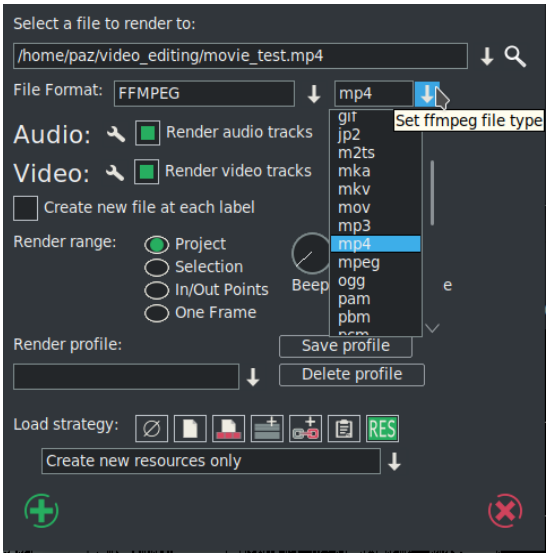


Figure 7.1: *Example of the Render menu*

Select a file to render to: enter the path and filename to write the rendered file to in the textbox below.

File Format: use the down arrow to see file format options. For ffmpeg, which has its own set of options, you will then have to select an ffmpeg file type from the down arrow choices. The format of the file determines whether you can render audio or video or both.

Render audio tracks: check this toggle to generate audio tracks

Render video tracks: check this toggle to generate video tracks. The Render window will sometimes automatically update the Render Audio Tracks or Render Video Tracks checkbox as allowed by the chosen file format, but you should always check (figure 7.2). For example, if the PNG file format is selected, only the *Render Video Tracks* will be checked. Or if an ffmpeg format is chosen and the file format does not render audio, the *Render Audio Tracks* will be unchecked. The invalid choices will be ghosted out.

Wrench: select the *wrench* next to each toggle to set compression parameters. If the file format can not store audio or video the compression parameters will be blank. If *Render audio tracks* or *Render video tracks* is selected and the file format does not support it, trying to render will result in an error message. There are several hardware related formats that will only work on specific computers that support that hardware/software implementation and will result in an error message if not available. Pick a different one. This is especially relevant if using an AppImage which is generated on a

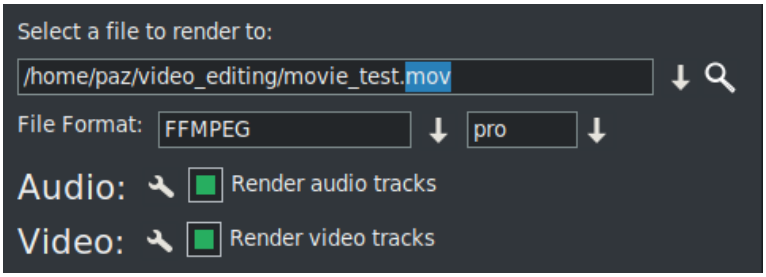


Figure 7.2: *Audio and Video tracks automatically checked for Pro file type*

computer with few extra hardware implemented features. More details on differing parameters of the format in the section: Extra "cin_" Options for Render with FFmpeg.

Create new file at each label the option causes a new file to be created when every label in the timeline is encountered – a separate file for each. This is useful for dividing long audio recordings into individual tracks. When using the Render Farm (described later), *Create new file at each label* causes one render farm job to be created at every label instead of using the internal load balancing algorithm to space jobs. If the filename given in the render dialog has a 2 digit number in it, the 2 digit number is overwritten with a different incremental number for every output file. If no 2 digit number is given, CINELERRA-GG automatically concatenates a number to the end of the given filename for every output file. For example, in the filename `/movies/track01.wav` the 01 would be overwritten for every output file. The filename `/movies/track.wav`; however, eventually would become `/movies/track.wav001` and so on. Filename regeneration is only used when either render farm mode is active or creating new files for every label is active.

Render range: choices are *Project* , *Selection*, *In/Out points*, and *One Frame* for single images like Tiff. For these images, Render range will have *One Frame* automatically checked and all of the others ghosted since nothing else makes sense (figure 7.3). This makes it easy to set the insertion point where you want the 1 frame to be rendered rather than having to precisely zoom in to set the in/out pointers. Note that whichever Render range is checked, remains checked so that if *One Frame* gets automatically checked, the next time you render it will still be checked and you will have to select a different one if desired. That is why you should always check the settings.

Beep on done: as a convenience when a render is complete, check this box. It gives you the chance to work on something else while waiting and still be immediately notified when the render is complete.

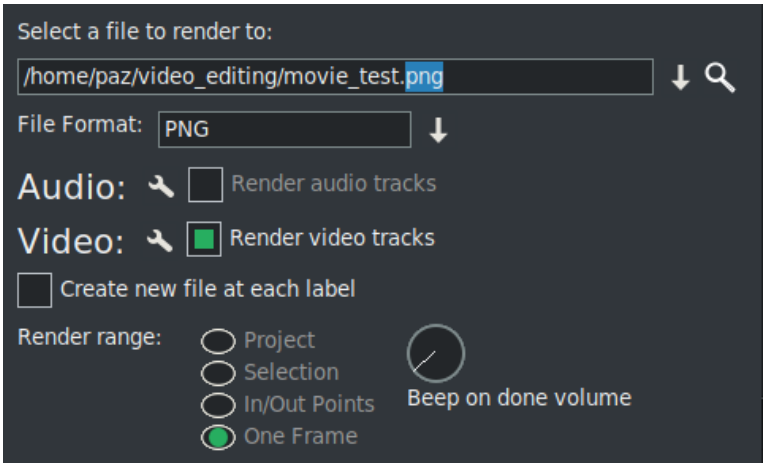


Figure 7.3: *Render menu displaying a PNG one frame option*

Render Profile: another convenience feature to take advantage of if you use specific render formats frequently, is to save that profile for future usage without having to set it up again.

Save Profile: after setting up your render preference formats, do not forget to type in a format name and then use the save profile button to save it. The named/saved Profiles will be saved in your $HOME/.bcast5/Cinelerra_rc file where it can be carefully modified.

Delete Profile: if you want to delete a saved profile, highlight the one you no longer want and delete.

Insertion strategy: select an insertion mode from the available choices as seen when you click on the down arrow on the right hand side of the option. The insertion modes are the same as with loading files. In the case if you select "insert nothing" the file will be written out to disk without changing the current project. For other insertion strategies be sure to prepare the timeline to have the output inserted at the right position before the rendering operation is finished.

Even if you only have audio or only have video rendered, a paste insertion strategy will behave like a normal paste operation, erasing any selected region of the timeline and pasting just the data that was rendered. If you render only audio and have some video tracks armed, the video tracks will get truncated while the audio output is pasted into the audio tracks.

7.1.1 Extra "cin_" Options for Render with FFmpeg

There are several special parameters that can be used in the ffmpeg options file to pass values to the codecs that are not normally available. They're called Global Options. These are explained below.

cin_pix_fmt The Render menus allows you to choose the codec input pixel format (figure 7.4). The Pixels selection provides the available pixel format options for the chosen codec type; valid choices vary for the different file types. This list represents the formats that the codec advertises. It is not always complete, and it may include options that are not legal with all parameter configurations.

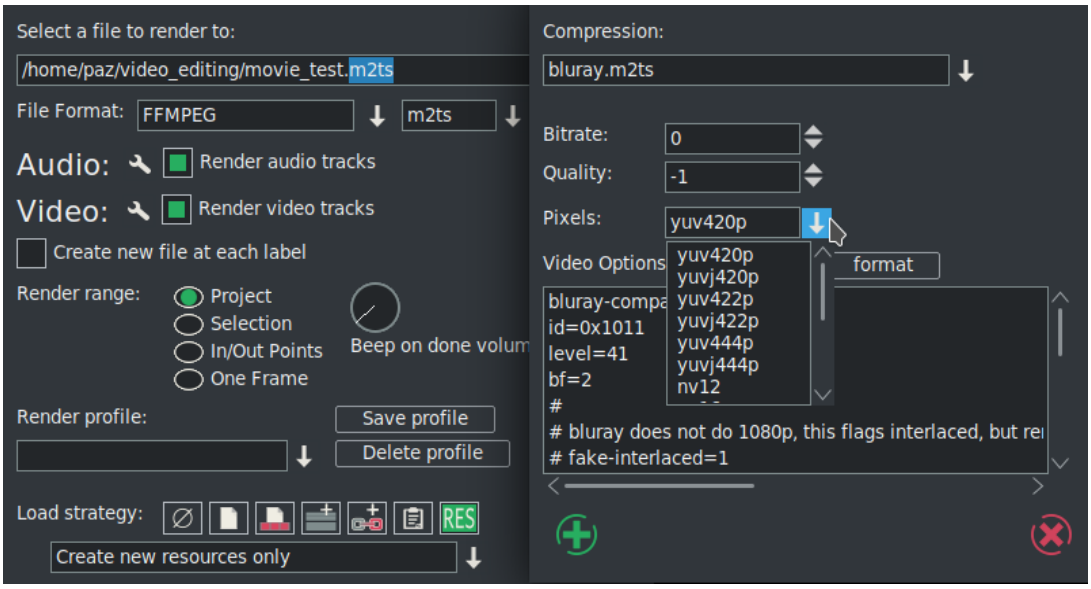


Figure 7.4: *Render & Video Preset menus displaying Pixel choices*

- The *Bitrate*, *Quality*, and *Pixels* fields are only updated when the Video Options are reloaded. This occurs when you either change the ffmpeg file format, or video presets compression fields.
- If the video options preset has *cin_pix_fmt* defined, its value will be loaded as the default. If you override the default, the value you specify will be used.
- If the video options preset does not have *cin_pix_fmt*, the default pixel format will be computed by ffmpeg (*avcodec_find_best_pix_fmt_of_list*), using the session format as the source choice. The *best* is usually the format which is most similar in color and depth.
- If no choices are available, yuv420p for video will be used.
- You can also specify ffmpeg pixel formats which are not in the list. The list is provided by ffmpeg as input selection, but is more like suggestions than fact. For example, the raw formats can take almost any format, but the rawvideo codec actually specifies no legal formats. Note that if you want a very specific Bitrate you must make sure there is not conflicting parameter values set such as Quality or CRF.

Some option files provide *cin_pix_fmt* to suggest a choice for good quality output or to prevent parameter errors when the other provided parameters conflict with the *best* pixel format. This is the case in `faststart_h264.mp4` where the *profile=high* parameter dictates pixel format must be `yuv420p`.

cin_bitrate If you specify the bitrate, you can not specify the quality or CRF.
Example: *cin_bitrate=2000000*

cin_quality If you specify the quality, you can not specify the bitrate.
Example: *cin_quality=7*

cin_stats_filename This parameter is useful for 2 pass operations.
Example: `cin_stats_filename /tmp/cin_video_vp9_webm`

cin_sample_fmt For audio the preset sample format default is computed in a similar way as stated above for video or can be set with the *cin_sample_fmt* parameter (figure 7.5). If no choices are provided, s16 will be used.
Example: *cin_sample_fmt=s16*

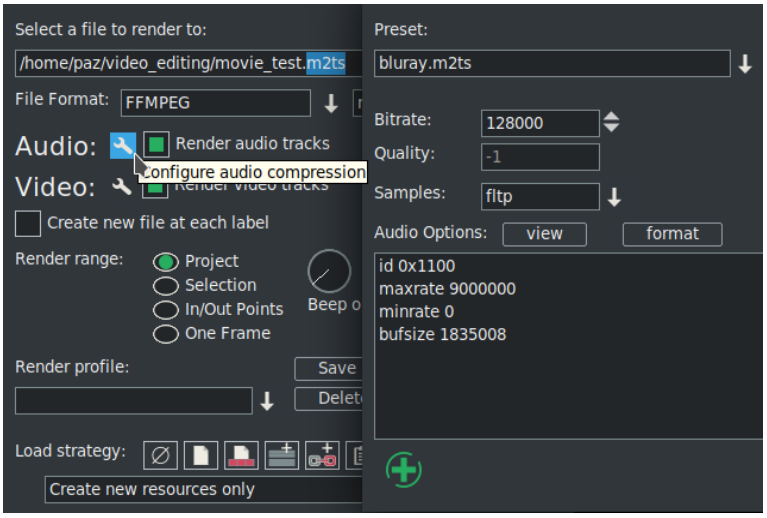


Figure 7.5: *Render menu showing where Samples is*

Private Options (muxers). In the window of the *wrench* in addition to the *View* button, which allows more global options and changes to the formats, there is an additional *Format* button that allows you to modify the Private Options, i.e. relating to specific muxing formats. You can list private options available in a encoder with:

```
$ /path/to/cin/thirdparty/ffmpeg-X.Y/ffmpeg -h encoder='libcodec' (i.e. libx265)
```

More information on all these options can be found at ffmpeg.org sections 19 and 21. See also Modifying FFmpeg Format Options inside CINELERRA-GG.

Render presets in CINELERRA-GG should work Out Of the Box. You can still configure the *Global Options* and *Private Options* manually. Finding the combination of parameters that best suits your needs, or simply finding working (*legal*) combinations, requires studying each codec in depth. You can start by looking in Wikipedia until you get to download and study the *white papers* of the codecs of interest. In any case, you must then start a long experimental phase, trying presets with different configurations or creating new ones, until you get satisfactory results. If you create new presets it is a good idea to make them known on the mailing list (`https://lists.cinelerra-gg.org/mailman/listinfo/cin`) or on the MantisBT Bug Tracker (`https://www.cinelerra-gg.org/bugtracker/my_view_page.php`) so that they can be integrated into subsequent versions of CINELERRA-GG. For an introduction see Overview on Formats and Codecs.

7.2 Some Specific Rendering

The next few pages relate to rendering for specific common cases. Also see: `https://gist.github.com/dexeonify/ed31c7d85fcf7297719e2ec4740fafda`)

7.2.1 FFmpeg Common H.264 Rendering

Because H.264 is so widely used, the method in CINELERRA-GG Infinity is outlined below. These setup steps make it easy to just get started.

- File → Render
- File Format → FFMPEG + mp4
- Video Wrench → Preset → h264.mp4 + bitrate: 6000000 (or whatever) + OK
- Audio Wrench → Preset → h265.mp4 + bitrate: 224000 (or whatever) + OK
- Set your target path in: Render → Select a file to render to
- Set your timeline in: Render → Render range + click Project
- Set your insertion strategy: Replace project (or whatever)
- Press OK to start rendering.

7.2.2 Lossless Rendering

Lossless means that in the compression of a file, all of the original data, every single bit, can be recovered when the file is uncompressed. This is different than *lossy compression* where

some data is permanently deleted so that when uncompressed, all of the original data can not be exactly recovered. Lossy is generally used for video and sound, where a certain amount of information loss will not be detected by most users or the playback hardware does not reproduce it anyway – it is a trade-off between file size and image/sound quality. The files created will be more than 10 times larger than usual. Most players will not be able to decode lossless as the bitrate will overwhelm the device.

For x264 lossless compression to work, the only color model allowed here is yuv420p. Any other specification will be converted to yuv420p and the data will be modified. Also, keep in mind that the YUV color model has to be converted to RGB, which also modifies the data.

To use x264 lossless rendering – choose File format of ffmpeg, m2ts in the Render window. Click on the Video wrench, which brings up the Video Preset window and scroll down in the Compression filebox and choose `lossless.m2ts`. *Preset=medium* is the default, but can be varied from *ultrafast* (least amount of compression, but biggest file size) to *veryslow* (most amount of compression, but still HUGE) in the parameter box where you see $qp = 0$. This option is also available for bluray creation.

7.2.3 Two-pass Encoding with FFmpeg

In CINELERRA-GG for two-pass, you need to run ffmpeg twice, with the same settings, except for designating the options of pass 1 for the first pass and then pass 2. In pass 1, a logfile that ffmpeg needs for the second pass is created. In pass 1 the audio codec should be specified that will be used in pass 2. For more information on ffmpeg 2-pass, check ffmpeg.org. Different libraries may have different requirements and you will probably have to determine this by looking online at ffmpeg or looking directly at that code.

This 2 line ffmpeg 2-pass operation can be functionally duplicated in CINELERRA-GG in the steps below them:

```
ffmpeg -y -i $INPUT \
-c:v libx264 -b:v 2600k -pass 1 \
-c:a aac -b:a 128k -f mp4 /dev/null && \
ffmpeg -i $INPUT \
-c:v libx264 -b:v 2600k -pass 2 \
-c:a aac -b:a 128k $OUTPUT.mp4
```

1. After you have completed your editing, do a Save Session with `File → Save as...` Before starting, be sure your session is ready for batch render. That is, positioned at the beginning and nothing selected.

2. Bring up `File → Batch Render...` where you will do the setup.

3. Click on the *Delete* box to remove old jobs highlighted in the bottom listbox.
    - For the *File Format* choose ffmpeg and mp4 for the type.
    - Set *Output path* to the path and filename for the render output file.
    - Click on *Use Current EDL* to use the designated EDL Path file.
    - Click on *New* and you will see a new highlighted job show up in the listbox at the bottom.
    - Use the Audio wrench to set bitrate to 128000 (128*k* as in ffmpeg example above).
    - Click checkmark OK. Open the video tools with the video wrench.
    - Set the Video Compression to *h264.mp4* (as seen in the example).
    - Set the bitrate to 2600000 (2600*k* as in ffmpeg example above).
    - Add the following 2 lines after the first line:

```
flags +pass1
passlogfile /tmp/"{temporary log file name}.log"
```

    Click checkmark OK.

4. Click on *New* to create the second pass job. You will see this second job in the listbox below. Use the Video wrench and change pass1 to pass2 as follows.

```
flags +pass2
```

5. Click checkmark OK.
6. Click on the *Start* box and watch it go!
7. You can now check the output file for results. At the time this was documented, *rc=2pass* will be in the output.

If you need to re-render this, the Batch Render will still be set up but you have to click on the *Enabled* column in the listbox to re-enable the jobs to run which puts an X there. Click Start again. You can reuse batch job using the *save jobs* and *load jobs* buttons in the batch render dialog.

Render shortcuts for webm, h264, h265 are available by using the option files that are already set up for this purpose. Use the render menu as usual, with ffmpeg/mp4, choose h264 or h265 *pass1of2_h26x* for the video and *passes1and2_h26x* for the audio; with ffmpeg/webm, choose *pass1of2_vp9*. When that is finished, you will have to use the render menu again and this time for video, choose *pass2of2_h26x* or *pass2of2_vp9*. The logfile is hard coded in the options file so will write over any currently existing logfile if you do not change it before you start the render.

Requirements for some other libraries (used instead of *flags +pass1* & *passlogfile*):

x265: add this line:

```
x265-params=pass=1:stats=/tmp/{temporary-log-file-name}.log
```

at the time this document was written, you should see in the output: *stats-read=2*

libvpx-vp9, xvid, and huffyuv:

```
cin_stats_filename /tmp/{temporary-log-file-name}.log
flags +pass1 (or flags +pass2 for the second pass)
```

NOTE: for vp9, the best Pixels is *gbrp*

7.2.4 Use case: High Efficiency Video Coding (HEVC)

An example of video profile based on CRF, a quality-controlled variable bitrate, instead of fixed quality scale (ABR). HEVC (H.265) was developed as a successor to AVC (H.264) to more efficiently compress the future large amounts of data from 2/4/8k videos. In comparison to AVC, an average saving of around 30 percent can be assumed for the same quality. Because HEVC is not bound to any size format, it is suitable for virtually any image size.

The following example is HD and FullHD oriented and produces a picture quality similar to the Blu-ray with some limitations. As container Matroska (`.mkv`) is used, but also mp4 and others are possible.

```
matroska libx265

# CRF 16 creates a balanced compromise
# between quality and file size.
crf=16

# Preset changes encoding speed and generally
# degrades the overall result. Medium (default)
# always fits.
preset=medium

# Additional parameters that are passed on to the codec.
# me=star improves the search for very fast
```

```
# movements, but slows down the encoding.
#x265-params=me=star

# Keyint does FFmpeg automatically, otherwise
# the setting must match the frame rate.
#keyint_min=25

# Profile does FFmpeg automatically.
#profile=high

# Source sRBG and retention of color space.
# 720/1080=bt709 if no profile set. Useful
# for formats smaller than 720 if no lossy
# conversion is desired.
colorspace=bt709
color_trc=bt709
color_primaries=bt709

# Output in 10 bit, prevents 8-bit step formation
pixel_format=yuv420p
```

NOTE:

A CRF of 16 delivers satisfactory results in most cases. However, if the video material is really *grainy*, a CRF 16 can lead to unwanted large files. In this case, a trial export of perhaps one minute should be performed. The resulting bit rate can be used to correct the CRF to 17, 18, 19. . . – remember, a CRF of 0 (zero) means lossless, the higher the number the stronger the lossy compression. The approximate calculation of the final file size can be extrapolated from the sample export.

The color space information must be used explicitly so that it can be included in the video. CINELERRA-GG or FFmpeg does not write it by itself. Without this information the players (e. g. mpv) stick to the dimensions of the video and take the assumed color model from a table. With videos in the dimensions from 720 to 1080 this is bt709. For smaller dimensions, e. g. DVD, bt601 is assumed and for 4k and above it is bt2020. Normally this is not a problem, but if you want to export a FullHD without color loss to a smaller size like 576 for example, you have to inform the encoder as well as the decoder of the player. This also applies if the videos are to be loaded on video platforms, where they are then converted into videos of different sizes. It is a security measure to prevent false colors, such as the color profiles in digital photos and the copies made from them.

The HEVC tuning has not been considered here, because it is is rarely used and requires background knowledge.

Further links:

- x265 Documentation
- x265 Command Line Options
- x265 Presets/Tuning

7.2.5 YouTube with CINELERRA-GG

To create a youtube or dailymotion video, you can easily follow the steps below. You will have to learn a lot more about CINELERRA-GG to take full advantage of its capabilities and make some really special videos, but this is just to get a start and to see the possibilities.

1. Start CINELERRA-GG; usually you can do this by clicking on CINELERRA-GG icon or key in `cin_path/bin/cin`.
2. In the Program window on the lower left side of your screen, left mouse click the *File* pulldown.
3. You will see *Load files* as the second choice so left mouse click this and find your video file to load, highlight it, and check the green checkmark in the lower left hand corner to get it loaded.
4. Edit your video in the Program window using the basic commands of:
    - play and then stop using the space bar
    - move the mouse and then left click to move the insertion (location) pointer
    - cut a section out by holding down the left mouse and drag, then key in “x” to cut or “c” to copy
    - paste a copy or cut section by moving the insertion pointer, then key in “v”
5. Add a title by highlighting the *Video Effects* in the right hand side Resources window; then highlighting the *Title* icon and dragging it to the Program window video track and dropping.
6. Click on the middle icon button (looks like a magnifying glass) on the brown colored Title bar to bring up the Title window bottom text box and key in a title.
7. Use the *File* pulldown to select *Render* to create the desired video. In the *Render* window just next to the empty box to the right of the *ffmpeg* file format, click on the down arrow shown there to see the choices and pick *youtube*. Then move back up to key in the path and filename to render to. It will pick all of the defaults automatically for you so then just click on the green checkmark to have it start. There is a progress bar in the main window, very bottom of the right hand side.
8. Key in “q” in the main window to get out of CINELERRA-GG and yes or no to save your edit session.

Youtube will allow the upload of the resulting rendered file as named. However, Dailymotion requires that the file be named with an acceptable extension so you must rename the output file to have the extension of .webm instead of .youtube

There are currently 6 specific variations within the ffmpeg (file format) / youtube (file type) for different video options. You see these when you click on the wrench to the right of the word Video and then the Compression down arrow in the Video Preset window. The first 3 are based on Webm/Vp9[1] and contain basic comments of usage and where to find more information.

The first 3 below, plus any of the VP9 files under the file type of *webm* are the recommended options to use because they are freely usable in any circumstance.

sd.youtube	Standard Definition use with default audio/Opus stereo.youtube
hd.youtube	High Definition " "
uhd.youtube	Ultra High Definition " "

For more details and options on VP9, see: https://developers.google.com/media/vp9

Alternatives based on h264 and for non-commercial use are listed below. For Dailymotion, these must be renamed to have a different extension of .mp4 instead of .youtube before uploading.

sd_h264.youtube	Standard Definition – must change to audio stereo_with_h264.youtube
hd_h264.youtube	High Definition - " "
uhd_u264.youtube	Ultra High Definition - " "

These same steps have been verified to work for creating Dailymotion videos – however, the created files must be renamed before uploading to change the youtube extension to webm instead for Dailymotion.

7.2.6 VP9 parameters

VP9[2] is a video codec licensed under the BSD license and is considered open source, the WEBM container is based on MATROSKA for video and OPUS for audio. There are some common VP9 rendering options files that support creation of video for YouTube, Dailymotion, and other online video services. YouTube easy startup steps are documented above.

[1] credit by Frederic Roenitz
[2] credit Frederic Roenitz

Note: For VP9 and AV1 presets, two pass rendering is recommended, which provides higher quality.

Below is one of the VP9 rendering options file with documentation for specifics:

webm libvpx-vp9

from https://developers.google.com/media/vp9/settings/vod/

1280x720 (24, 25 or 30 frames per second)

Bitrate (bit rate)

VP9 supports several different bitrate modes:

mode:

Constant Quantizer (Q)	Allows you to specify a fixed quantizer value; bitrate will vary
Constrained Quality (CQ)	Allows you to set a maximum quality level. Quality may vary within bitrate parameters
Variable Bitrate (VBR)	Balances quality and bitrate over time within constraints on bitrate
Constant Bitrate (CBR)	Attempts to keep the bitrate fairly constant while quality varies

CQ mode is recommended for file-based video (as opposed to streaming). The following FFMpeg command-line parameters are used for CQ mode:

FFMpeg:

-b:v <arg>	Sets target bitrate (e.g. 500k)
-minrate <arg>	Sets minimum bitrate.
-maxrate <arg>	Sets maximum bitrate.
-crf <arg>	sets maximum quality level. Valid values are 0-63, lower numbers are higher quality.

Note 1: Bitrate is specified in kbps, or kilobits per second. In video compression a kilobit is generally assumed to be 1000 bits (not 1024).

Note 2: Other codecs in FFMpeg accept the *-crf* parameter but may interpret the value differently. If you are using *-crf* with other codecs you will likely use different values for VP9.

```
bitrate=1024k
```

`minrate=512k`
`maxrate=1485k`
`crf=32`

Tiling splits the video into rectangular regions, which allows multi-threading for encoding and decoding. The number of tiles is always a power of two. 0=1 tile; 1=2; 2=4; 3=8; 4=16; 5=32
`tile-columns=2`

(modified from https://trac.ffmpeg.org/wiki/EncodingForStreamingSites)

To use a 2 second *GOP* (Group of Pictures), simply multiply your output frame rate × 2. For example, if your input is *-framerate 30*, then use *-g 60*.
g=240

number of *threads* to use during encoding
`threads=8`

Quality may be set to good, best, or realtime
`quality=good`

Speed: this parameter has different meanings depending upon whether quality is set to good or realtime. Speed settings 0-4 apply for VoD in good and best, with 0 being the highest quality and 4 being the lowest. Realtime valid values are 5-8; lower numbers mean higher quality
`speed=4`

7.2.7 Piping Video to a Command Line

You can pipe a video to any command line on the computer, such as ffmpeg. This can be especially useful with raw video files. Next is an example usage.

1. on a terminal window create a named pipe file, for example:

   ```
   mknod /tmp/piper.yuv p
   ```

   load your video and do your editing
2. set up your Render (`Shift-R`), you can choose a raw format such as *yuv* or *rgb*
3. for the filename *Select a file to render to*, use the named pipe as created in step 1 (`/tmp/piper.yuv`)
4. for *Insertion Strategy*, you will want to make sure to select *insert nothing*

5. click for OK on the green checkmark.(the CINELERRA-GG gui will look like it is hanging while waiting for a command line to use the pipe.)

6. on the terminal window, keyin your command, for example:

```
/mnt0/build5/cinelerra-5.1/thirdparty/ffmpeg-3.4.1/ffmpeg -f \
rawvideo -pixel_format yuv420p -video_size 1280x720 \
-framerate 30000/1001 -i /tmp/piper.yuv /tmp/pys.mov
```

A slightly different option can be used instead that may be more familiar to some. In the render menu after choosing the File Format of *ffmpeg*, use the pulldown to choose *y4m* as the file type. This choice results in putting a header on the rendered output with some pertinent information that can be used for ffmpeg processing thus alleviating the requirement for *pixel_format*, *video_size*, and *framerate* on the ffmpeg command line. In this case the format is *yuv4mpegpipe* instead of *rawvideo*. An example command line would look as follows (assuming the created pipe is called `piper.y4m`):

```
ffmpeg -f yuv4mpegpipe -i /tmp/piper.y4m -vcodec libx264 /tmp/test.mp4
```

7.2.8 Faststart Option for MOV type files

If you have mov video and want to be able to start playing without having to first load the entire video, *-movflags=+faststart* is needed for ffmpeg to put the meta-data, known as the *moov atom*, at the beginning of the file. Otherwise, ffmpeg puts this atom at the end of the video file which means you have to wait to play until the whole video is loaded. Or worse yet, if the file becomes damaged in the middle and you can not get to the end, you won't be able to play anything.

Now you can have the *moov atom* put on the front of the file (automatically via a second pass). To do this, when rendering using ffmpeg & either the mp4 or qt format/container, click on the video/audio wrenches and choose *faststart_h264*. With the *qt* format, settings will just be the default whereas the *mp4* format uses the highest quality and lowest file size as possible, but you can easily modify these options in the associated Video Preset textbox.

7.3 About Image Sequences

CINELERRA-GG supports image sequences with both decoding and encoding.

CINELERRA-GG by default uses ffmpeg as encoding/decoding engine but we can disable it to have the specific internal engine available. See FFmpeg Early Probe Explanation on how

to switch between engines. With the internal engine we can create and load sequences of OpenEXR; PNG; TIFF; TGA; GIF; PPM and JPEG. There is also support for DPX sequences, but only in read and without rendering presets. With ffmpeg we can create and load DPX sequences or create a custom preset for any kind of image. Using these formats results in great timeline efficiency and high video quality at the cost of taking up a lot of space because they are uncompressed (or with lossless compression). By rendering, you will get as many still images as there are frames in the project, plus a *file-list* (or *TOC*) that indexes the images. A good practice is to create a folder to contain the images (for example `/tmp/img_seq/`) and then open the rendering window in CINELERRA-GG and set a serial and increasing number as the name (for example: `/tmp/img_seq/image %05d.png`). *image* is a generic name chosen at will; % creates a progressive sequence of distinct images; 05*d* indicates how many digits the image number will be, in this case 5 digits to go from 00000 to 99999. Once we have our folder of images, if we want to import it in a project just load the file-list, which includes the link to all the files of the sequence. To learn more about using and creating a preset with ffmpeg of an image sequence, see The FFmpeg Image2 Streams and/or Image Sequence Creation.

7.4 Data storage formulas

If we are dealing with large projects and poorly compressed formats, we will get large files that are difficult to manage and take up a lot of space on the HDD. We present some simple formulas to be able to calculate the space that will be occupied and the data rates we have to deal with:

Frame size

$$\frac{Width \times Height[pixels] \times BitDepth[bits/pixel] \times Color}{8[bit/Byte]} = ...[MB/frame]$$

File size

$$Framesize[MB/frame] \times frames[frame] = ...[MB]$$

Data Rate

$$Framesize[MB/frame] \times fps[frame/sec] = ...[MB/sec]$$

Data in 1 Hour

$$\frac{DataRate[MB/sec] \times 3600[sec]}{1024MB/GB} = ...[GB]$$

7.5 Batch Rendering

Batch Rendering as implemented in CINELERRA-GG is considered to be more of an advanced feature and careful usage is advised. It automates the rendering of audio/video files in that you can establish a set of job parameters, save them, and use them repeatedly

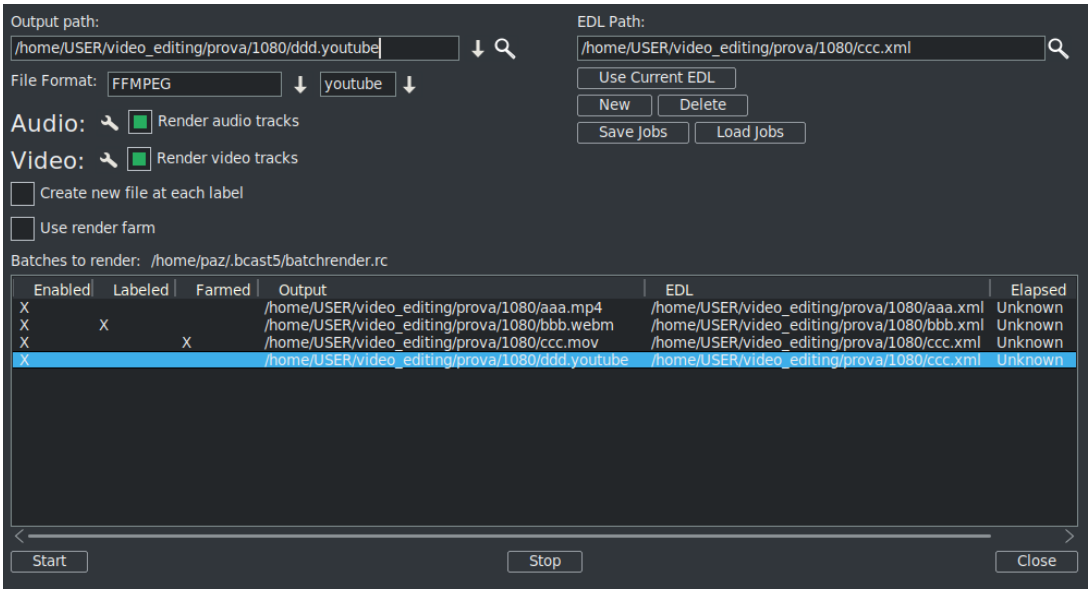


Figure 7.6: *Example of the Batch Render menu*

(figure 7.6). It also allows for CINELERRA-GG to be run by external programs, with no need for the user to manually interact with the user interface.

If you want to render many projects to media files without having to constantly set up the render dialog for each one, batch rendering is a more efficient method of rendering. To use this feature you need to understand certain concepts.

1. You must define a list of Batches (*Job*) before starting the rendering. This is created using the *New* button and displayed in *Batches to Render* dialog.
2. Each batch consists of a source project already created in CINELERRA-GG, e.g. `aaa.xml`, to which we assign the rendering parameters.
   - to associate `aaa.xml` to the batch we use the *EDL Path* input field.
   - we decide a name and path for the output file.
   - let's set the *File Format* of the output file.
   - We configure the file with the Audio/Video *wrench*.
   - we decide whether to create different files for each *label* and whether to use a *Render farm*.
3. Created the first batch, we will see it appear in the dialog *Batches to Render*.
4. Using the *New* button again we create a second batch for another source project (`bbb.xml`) and configure it at will.
5. We continue with the source projects `ccc.xml`, `ddd.xml`, etc. until we run out of projects that we want to render in batch.

6. Note that each batch has its own name, path and rendering parameters.

7. Now we have our *Job*, a list of batches. We can still configure it or modify it if we want to change something. In addition we can delete a batch from the list or we can disable it in the *Enabled* field so that it is not taken into account during rendering, but without deleting it.

8. Finally we start batch rendering with the *Start* button.

Let's see in detail how to set the Batch Rendering.

The first thing to do when preparing to do batch rendering is to create one or more CINELERRA-GG projects to be rendered and save them as a normal project, such as `aaa.xml`. The batch renderer requires a separate project file for every batch to be rendered. You can use the same CINELERRA-GG project file if you are rendering to different output files, as in an example where you might be creating the same output video in different file formats.

You do not have to render an entire projects. We can limit ourselves to an *active region* that we can set through a selection in Cut and Paste mode, with labels or In/Out Points. Or the rendering will start from the Insert Point position until the end of the project. Remember: if we want to render the entire project (and not just one active region) it is important to bring the Insertion Point to the beginning of the timeline. This is the only way we are sure to include the whole project.

With all the CINELERRA-GG xml project files prepared with active regions, go to `File` → `Batch Render`. This brings up the batch render dialog. The interface for batch rendering is more complex than for single file rendering. A list of batches must be defined before starting a batch rendering operation. The table of batches appears on the bottom of the batch render dialog and is called *Batches to render*. Above this are the configuration parameters for a single batch; a batch is simply a pairing of a project file with a choice of output file and render settings.

It may be advisable to start with a *Delete* so you don't have any problems. Set the *Output path*, *File format*, *Audio*, *Video*, and *Create new file at each label* parameters as if you were rendering a single file. These parameters apply to only one batch. In addition to the standard rendering parameters, you must select the *EDL Path* to be the project file (such as `aaa.xml`) that will be used in the batch job. In this case, *EDL Path* is not related in anyway with the EDL files as created by `File/Export EDL`. In batch render mode the program will not overwrite an existing output file and will simply fail, so make sure that no files with the same name as the output files exist before starting.

If the batches to render list is empty or nothing is highlighted, click *New* to create a new batch. The new batch will contain all the parameters you just set. Repeatedly press the *New* button to create more batches with the same parameters. When you highlight any batch, you can edit the configuration on the top of the batch render window. The highlighted batch is always synchronized to the information displayed. You can easily change the

order in which the batch jobs are rendered, by clicking and dragging a batch to a different position. Hit *Delete* to permanently remove a highlighted batch. In the list box is a column which enables or disables the batch with an `X` meaning the batch job is enabled and will be run. This way batches can be skipped without being deleted. Click on the *Enabled* column in the list box to enable or disable a batch.

The description of each of the columns in the batch list are as follows:

Enabled: an X in this column means the batch job will be run.

Labeled: an `X` in this column goes hand in hand with create new file at each label.

Farmed: to use or not the render farm.

Output: path and filename for the generated output.

EDL: the path and filename of the source EDL for the batch job.

Elapsed: the amount of time taken to render the batch if finished. If field is empty, it did not run.

The `File` → `Batch Render` pulldown brings up the Batch Render window to be used for batch rendering as well as DVD/BD creation. There are some additional buttons that can save time and mistakes. These are described next.

Save Jobs when you have set up the batch jobs the way you want and you think you may have to run them more than once, it is beneficial to save the jobs for later use so you easily run them again. It is recommended to use a filename with .rc as the extension so that it is obvious that it is a list of batch jobs to be run.

Load Jobs reload a previous set of saved jobs. This can come in handy if you did not have the time to render them when you originally set them up, if you need to rerun, or if you got interrupted.

To start rendering from the first enabled batch, hit *Start*. Once rendering, the main window shows the progress of the batch. After each batch finishes, the elapsed column in the batch list is updated and the next batch is rendered until all the enabled batches are finished. The currently rendering batch is always highlighted red. To stop rendering before the batches are finished without closing the batch render dialog, hit *Stop*. To stop rendering before the batches are finished and close the batch render dialog, hit *Close*. Or you can exit the batch render dialog whether or not anything is being rendered, by hitting *Close*.

You can automate CINELERRA-GG batch renders from other programs. In the batch render dialog, once you have created your list of batch render jobs, you can click the button *Save Jobs* and choose a file to save your batch render list to. It is recommended that you use

a filename with the extension of .rc in order to make it obvious that this is a list of batch jobs to render. Once you have created this file, you can start up a batch render without needing to interact with the CINELERRA-GG user interface. From a shell prompt, from a script, or other program, execute:

```
{path_to_cinelerra}/cin -r batchjob.rc
```

substituting your actual filename for `batchjob.rc`. **Warning** this file will be modified so if you use any filename that is not a legitimate list of batch jobs to render, that file will be overwritten and its previous contents destroyed. When invoked with these parameters, CINELERRA-GG will start up and run the rendering jobs in the list contained in that file starting at the defined *active region*, without creating its usual windows. If you do not specify a filename, the default will be $HOME/.bcast5/batchrender.rc. Possible messages you might see where you started up the job are as follows.

The following files exist: - filename - Won't overwrite existing files that batch job will not run in order to prevent writing over previous run.

"**filename**" **No such file or directory** the specified batch job file does not exist.

"**filename**": **Permission denied** the specified batch job file does not have write permission so can not be updated.

Render::run: filename the batch job with the name of filename will be processed.

** **rendered 0 frames in 0.000 secs, 0.000 fps** either you used a file that is not a list of batch jobs or the batch jobs within the file were not enabled.

7.5.1 Advanced features

Warning: *Save to EDL path* overwrites the current EDL thus destroying the original contents.

Although the operation of Batch Rendering in CINELERRA-GG is similar to that of other NLEs, there is one big difference that we need to take into account. The render setup is not done on a project-by-project basis, which are then brought into the Batch window to be rendered automatically. The setup must be done in the Batch rendering window, where various projects are loaded and set up. In the case of similar projects, derived from a single EDL with some variation, this mode offers the possibility of altering the projects without having to open each individual project, make the changes, set up the rendering, save and import into the Batch window. The procedure is to select the batch we want to modify in the Batches to render window; operate on the currently open timeline (even if it does not correspond to the one we want to modify) making the desired changes and then press the *Save to EDL path* button. Thus the chosen batch, while retaining its original name, will now contain the modified project. Since this possibility destroys the original

EDL overwriting it with the modified one, you must be very careful. This procedure is convenient in case the batches are similar, i.e. they are variations of the same EDL, where we want to experiment with other effects, other output formats or when trying out various cuts of a DVD/BD before the final production. It might also be useful to use an *active region* of the timeline, so as to speed up rendering times but still have an indicative result for comparison. Instead operating on different projects, we can do a *save as...* of the project on the timeline to have a new EDL with a new name and then replace it with the batch selected in the joblist using the *Use Current EDL* button. The new project (with its name) overwrites the original project.

The *Save to EDL Path* and *Use Current EDL* buttons can be valuable tools for advanced usage or for developers doing testing. Description of how you can expect them to work will help to illustrate how to take advantage of their capabilities (figure 7.7):

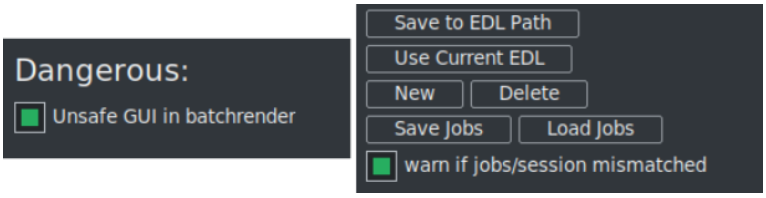


Figure 7.7: *New Buttons with Unsafe GUI in batchrender*

Save to EDL Path Warning: this function overwrites the contents of the original EDL with new data, keeping the name of the original. If we don't know exactly what we're doing we may lose the original project. If you have made a change to the EDL, use this button to save the changes so that they will be used in the render operation. Although you can get the same results by using `File` → `Save...`, this capability was initially added to assist developers in testing the batch jobs needed to create dvd/bluray media as it keeps the work focused in a single window and retains the original job name. An example –you have everything all set up with a new job in the Batch Render window using `generic.xml` for the EDL path and with a job name of `original_name.xml`. Then you realize that you forgot to cut out a section in the media that is not wanted in the final product. You can cut that out and then *Save to EDL Path* so your change will be in effect for the rendering. Without this button, you would be using the EDL you started with and the cut would be ignored. Alternatively, if the cut changes are saved via `File` → `Save as...` with a filename of `new.xml` and then you use *Save to EDL Path*, the current highlighted job displayed in the window as `original_name.xml` will be replaced with `new.xml`. However, it is important to note that the result will be saved with the name `original_name` – that is, the new content from `new.xml` but with the old name of `original_name.xml`. To have this functionality we have to enable the checkbox in `Settings` → `Preferences` → `Appearance` tab; section *Dangerous:* and unchecked (default) *Unsafe GUI in batchrender*.

Use Current EDL Warning: this function overwrites the contents of the original EDL with new project. If we don't know exactly what we're doing we may lose the original

project. if you are working on media and still testing out the results, you can take advantage of this click-box to quickly get results. Basically, you change the media, save that change with another name (in order to preserve the original name in case you don't like the changes), and press *Use Current EDL*. As an example, a user creates a new job in the Batch Render window using the current media, previously defined in generic.xml, with the EDL path of `generic.xml`. The user then changes the media on the timeline, saves the changes via `File` → `Save as...` with a new name, such as `new_name.xml`, and then clicks on *Use Current EDL*. In this case, the EDL path listbox will be automatically updated to the `new_name.xml` and the current existing highlighted job will be replaced with the `new_name.xml` in the EDL column.

Warn if Jobs/Session mismatched Warning: It is better to keep this function unchecked because it is only needed in case of changes on the original EDL. By default it is hidden and is shown only if we enable the checkbox in `Settings` → `Preferences` → `Appearance` tab; section *Dangerous:* and checked *Unsafe GUI in batchrender*. After you set up your render and press Start, the program checks to see if the current EDL session matches your Batch Render job. If the EDL has been changed since the batch job was created, it warns you so that you have the opportunity to *Save to EDL* path to record those changes. Otherwise, you can dismiss that warning box, disable the warning message by unchecking the box and use the original values. If you never want to be warned about the mismatches, leave the box unchecked (figure 7.8). It is advisable to keep it unchecked because it can cause problems.

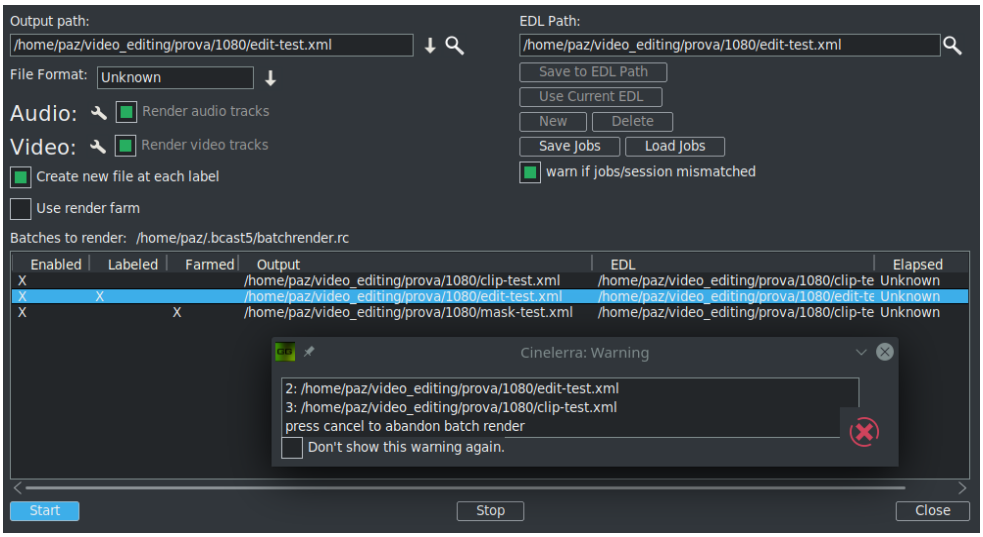


Figure 7.8: *Batch render with the 4 ghosted buttons on the right side + the Warning message below*

A very clear tutorial on these features can be found here[3]; in Russian but easily translatable with DeepL or similar.

[3] credit Igor Vladimirsky

7.5.2 Command Line Rendering

The command line rendering method consists of a way to load the current set of batch rendering jobs and process them without a GUI. This is useful if you want to do rendering on the other side of a low bandwidth network and you have access to a high powered computer located elsewhere. Setting up all the parameters for this operation is somewhat difficult. That is why the command line aborts if any output files already exist.

To perform rendering from the command line, first run CINELERRA-GG in graphical mode. Go to `File` → `Batch Render`. Create the batches you intend to render in the batch window and close the window. This automatically saves the batches in a file with the name of $HOME/.bcast5/batchrender.rc. Set up the desired render farm attributes in `Settings` → `Preferences` and quit out of CINELERRA-GG if you want to use the Render Farm capability. These settings are used the next time command line rendering is used to process the current set of batch jobs without a GUI. It is important to remember that the rendering will begin at the defined *active region* saved when the project was saved.

On the command line run:

```
cin -r
```

7.6 Background Rendering

Background rendering causes temporary output to be rendered constantly while the timeline is being modified. The temporary output is displayed during playback whenever possible. This is useful for transitions and previewing effects that are too slow to display in real time. If a Render Farm is enabled, the render farm is used for background rendering. This gives you the potential for real-time effects if enough network bandwidth and CPU nodes exist.

Background rendering is enabled in the `Performance` tab of the `Preferences` window. It has one interactive function `Settings` → `Toggle background rendering` . This sets the point where background rendering starts up to the position of the insertion point. If any video exists, a red bar appears in the time ruler showing what has been background rendered (figure 7.9). Because this creates a very large number of files, a Shell Command script is available to delete them if in the default location.

It is often useful to insert an effect or a transition and then select `Settings` → `Toggle background rendering` right before the effect to preview it in real time and full frame rates (figure 7.10).

Frames per background rendering job This only works if a Render Farm is being used; otherwise, background rendering creates a single job for the entire timeline. The

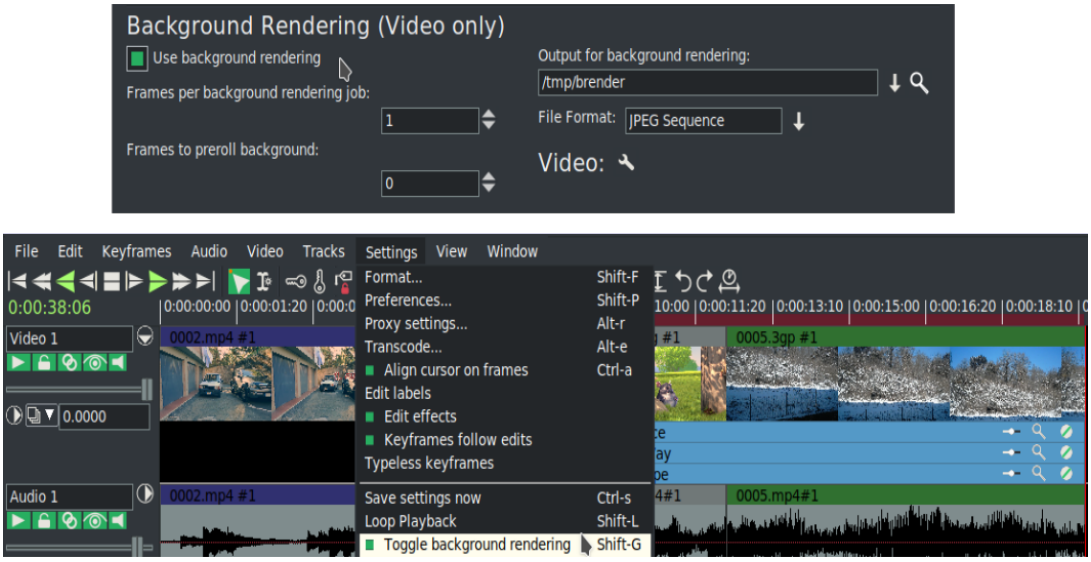


Figure 7.9: *Settings Background Rendering*

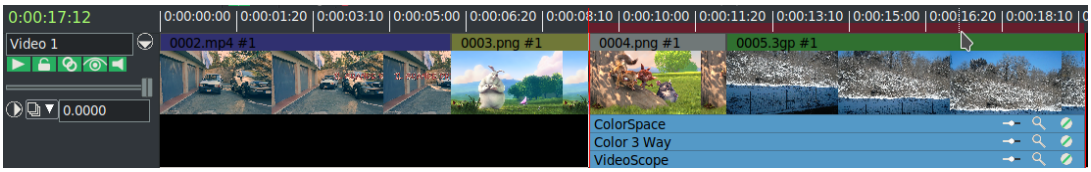


Figure 7.10: *Timeline with the top red bar*

number of frames specified here is scaled to the relative CPU speed of rendering nodes and used in a single render farm job. The optimum number is 10 - 30 since network bandwidth is used to initialize each job.

Frames to preroll background This is the number of frames to render ahead of each background rendering job. Background rendering is degraded when preroll is used since the jobs are small. When using background rendering, this number is ideally 0. Some effects may require 3 frames of preroll.

Output for background rendering Background rendering generates a sequence of image files in a certain directory. This parameter determines the filename prefix of the image files. It should be accessible to every node in the render farm by the same path. Since hundreds of thousands of image files are usually created, ls commands will not work in the background rendering directory. The browse button for this option normally will not work either, but the configuration button for this option works. The default value will be /tmp/brender . Because using background rendering creates a voluminous number of brender numbered files, a Shell Command script is available to delete them if they are in the default /tmp/brender format.

File format The file format for background rendering has to be a sequence of images. The format of the image sequences determines the quality and speed of playback. JPEG generally works well and is the default.

Tip: If you have rendered your whole project with *File format* set to JPEG and there are no missing numbers in the sequence, you can create a video from that sequence outside of CINELERRA-GG. For example, if using the default output so that your files are named /tmp/brender000000, /tmp/brender000001, ... in a window, you would type:

```
ffmpeg -f image2 -i /tmp/brender0%5d -c:v copy brender.mov
```

which would create the video file brender.mov - be sure to delete existing brender files before creating a new sequence to ensure there are no missing numerical values in the sequence.

7.7 Render Farm Usage

Render Farm uses background rendering, a feature of CINELERRA-GG where the video is rendered in the background, to speed up rendering significantly. Because rendering is memory and cpu intensive, using multiple computers on a network via a render farm is a significant gain. With CINELERRA-GG installed on all nodes, the master node and the clients communicate via a network port that you specify. The Master node is the one of the instance of CINELERRA-GG that we normally start with its gui; the other nodes are the instances of CINELERRA-GG that we decide to start in parallel from the terminal to create the Render farm (clients). CINELERRA-GG can distribute the rendering tasks over the network to the other computers of the Render Farm; or among all threads of a multicore CPU. The render farm software tries to process all of the rendering in parallel so that several computers can be used to render the results. The *Total jobs to create* in the setup or labels on the timeline are used to divide a render job into that specified number of tasks. Each background job is assigned a timeline segment to process and the jobs are sent to the various computer nodes depending upon the load balance. The jobs are processed by the nodes separately and written to individual files. You will have to put the files back together via a load with concatenation, or typically by using a command line tool from a script.

7.7.1 Basic Steps to Start a Render Farm

The following steps are just a guideline to start your render farm. It is assumed that you already have the master and client nodes communication, shared filesystem, permissions and usernames synched. Let's take the example of a network with 2 PCs: the master and the client. On the client we set 5 tasks; on the master we set 2 tasks.

1. On the master computer, use `Settings` → `Preferences` → `Performance tab` to set up a Render Farm:
    - check the *Use render farm* box;

- in the *Hostname* box, keyin your hostname or ip address such as 192.168.1.12 or *localhost* when using a single computer with a multicore CPU;
- enter in a port number such as 401–405 (only a root user can use privileged ports) or 10650... for non-root and click on *Add Nodes*. To find a range of free ports to use you can look at the file `/etc/services`;
- you will see something like the following in the Nodes listbox to the right:

On	Hostname	Port	Framerate
X	192.168.1.12	10650	0.0
X	192.168.1.12	10651	0.0
X	192.168.1.12	10652	0.0
X	192.168.1.12	10653	0.0
X	192.168.1.12	10654	0.0
X	localhost	10655	0.0
X	localhost	10656	0.0

- set the Total number of jobs to create. This number only pertains to client nodes, so we do not need to consider the master node;
- click OK on the bottom of the Preferences window.

2. For external network nodes, now we must join the nodes created to instances of CINELERRA-GG. On the client computers (192.168.1.12), on the terminal, start 5 background CINELERRA-GG tasks via:

```
cd {path_to_cinelerra}
cin -d 10650
cin -d 10651
...
cin -d 10654
```

   In practice, at each instance that we start, the cursor will be positioned in a new line ready to enter the next command, without exiting the task. If we have several PCs on the network, repeat these steps for each new client (with its own ip address).

3. Similarly, on the terminal, we must join the local nodes (internal to the Master node) created to instances of CINELERRA-GG. On the Master node (localhost), start the 2 background CINELERRA-GG tasks via:

```
cd {path_to_cinelerra}
cin -d 10655
cin -d 10656
```

   Similar to the previous point, the cursor positions itself in a new line ready to enter the next command, without exiting the task.

4. When your video is ready, setup a render job via `File` → `Render` or `File` → `Batch Render` and check OK.

5. The results will be in the shared file `path/filename` that you selected in the render menu with the additional numbered job section on the end as 001, 002, 003, ... 099 (example, `video.webm001`).

6. When finished, load your new files on new tracks via `File` → `Load` *concatenate to existing tracks* or if you used ffmpeg, run *RenderMux* from the Shell Scripts icon.

7. If you plan on doing more rendering, you can just leave the master/client jobs running to use again and avoid having to restart them. You can also close the terminal, but the jobs will remain active until you turn off the PC. Or you can kill them when you no longer are using them.

7.7.2 Render Farm Menu and Parameter Description

Below we describe the Performance tab for configuring a render farm (figure 7.11).

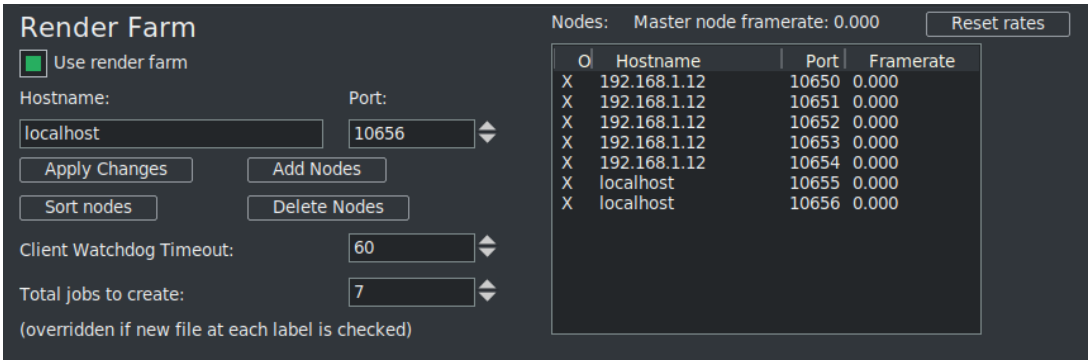


Figure 7.11: *Settings: Preferences: Performance tab, menu to set up your Render Farm*

Project SMP cpus although this field is not Render Farm specific, it is useful for CINELERRA-GG to have the CPU count and for using multiple threads.

Use render farm check this to turn on the render farm option. Once checked ALL rendering will be done via the farm including the usual Render (`Shift-R`). You may want to turn if off for small jobs.

Nodes listbox displays all the nodes on the render farm and shows which ones are currently enabled. The Nodes listbox has 4 columns – On, Hostname, Port, Framerate – which show the current values. An *X* in the *On* designates that that host is currently enabled; *Hostname* shows the name of the host; *Port* shows the port number that host uses; and *Framerate* will either be zero initially or the current framerate value.

Hostname this field is used to edit the hostname of an existing node or enter a new node.

Port keyin the port number of an existing or new node here. You can also type in a range of port numbers using a hyphen, for example 10650—10799 when you need to add many.

Apply Changes this will allow you to edit an existing node and to then commit the changes to hostname and port. The changes will not be committed if you do not click the OK button.

Add Nodes Create a new node with the hostname and port settings.

Sort nodes sorts the nodes list based on the hostname.

Delete Nodes deletes whatever node is highlighted in the nodes list. You can highlight several at once to have them all deleted.

Client Watchdog Timeout a default value of 15 seconds is used here and the tumbler increments by 15 seconds. A value of 0 (zero) disables the watchdog so that if you have a slow client, it will not kill the render job while waiting for that client to respond.

Total jobs to create determines the number of jobs to dispatch to the render farm. Total jobs is used to divide a render job into that specified number of tasks. Each background job is assigned a timeline segment to process. The render farm software tries to process all of the rendering in parallel so that several computers can be used to render the results. You should adjust this for each different video if your work varies considerably in time so that each task time is not too small.

To start, if you have computers of similar speed, a good number for *Total jobs to create* is the number of computers multiplied by 3. You will want to adjust this according to the capabilities of your computers and after viewing the framerates. Multiply them by 1 to have one job dispatched for every node. If you have 10 client nodes and one master node, specify 33 to have a well balanced render farm.

(overridden if new file at each label is checked) instead of the number of jobs being set to *Total jobs to create*, there will be a job created for each labeled section. If in the render menu, the option *Create new file at each label* is selected when no labels exist, only one job will be created. It may be quite advantageous to set labels at certain points in the video to ensure that a key portion of the video will not be split into two different jobs.

Reset rates sets the framerate for all the nodes to 0. Frame rates are used to scale job sizes based on CPU speed of the node. Frame rates are calculated only when render farm is enabled.

Framerates can really affect how the Render Farm works. The first time you use the render farm all of the rates are displayed as 0 in the `Settings` → `Preferences`, Performance tab in the Nodes box. As rendering occurs, all of the nodes send back framerate values to the master node and the preferences page is updated with these values. A rate accumulates based on speed. Once all nodes have a rate of non-zero, the program gives out less work to

lower rated nodes in an effort to make the total time for the render to be almost constant. Initially, when the framerate scaling values are zero, the program just uses package length – render size divided by the number of packages to portion out the work (if not labels). If something goes wrong or the rates become suspect, then all of the rest of the work will be dumped into the last job. When this happens, you really should *reset rates* for the next render farm session to restart with a good balance.

```
{path_to_cinelerra}/cin -h # displays some of the options.
```

7.7.3 Multi-core CPU Setup (Localhost)

If you are lucky enough to have a computer with a large cpu core count, setting up a render farm can really take advantage of using all of the cores at 100%. This is much faster than the default automatic threading capability. Since you don't need to communicate with other computers, you will not have to be concerned about TCP communication or shared disks/files; only localhost nodes. On the terminal you will need to open many instances of CINELERRA-GG by connecting them to the jobs created. The number of such jobs can be the total number of CPU threads (-1) or not. When you are going to be doing other work simultaneously while rendering a large job, you will want to leave some of the cpus available for that. Be sure to set *Project SMP cpus* in the `Settings → Preferences, Performance` tab to your CPU count. Follow the steps outlined in 7.7.1 but skip the step "For external network nodes".

7.7.4 Detailed Setup Description

CAUTION , any exact command lines worked as of 01/2018 on a Fedora system. These can change over time and on different operating systems/levels. Always check/verify any command line before using.

Set up CINELERRA-GG A CINELERRA-GG render farm is organized into a master node and any number of client nodes. The master node is the computer which is running the gui. The client nodes are anywhere else on the network with CINELERRA-GG installed and are run from the command line. Before you start the master node for CINELERRA-GG, you need to set up a shared filesystem on the disk storage node as this is the node that will have the common volume where all the data will be stored. The location of the project and its files should be the same in the client computers as in the master computer and to avoid problems of permissions, it is better to use the same user in master and clients. For example, if you have the project in `/home/<user>/project-video` you must create the same directory path on the clients, but empty. Sharing the directory of the location of your project on the master computer can be done with NFS as described next. Alternatively, you can look up on the internet how to use Samba to share a directory.

Create a shared filesystem and mount using NFS (only for Network) All nodes in the render farm should use the same filesystem with the same paths to the project files on all of the master and client nodes. This is easiest to do by setting up an NFS shared disk system.

1. On each of the computers, install the nfs software if not already installed. For example, on Debian 9 you will need to run: (be sure to check/verify before using any command line):

```
apt-get install nfs-kernel-server
```

2. On the computer that contains the disk storage to be shared, define the network filesystem. For example to export `/tmp`, edit the `/etc/exports` file to add the following line:

```
192.168.1.0/24(rw,fsid=1,no_root_squash,sync,no_subtree_check)
```

3. Next reset the exported nfs directories using:

```
exportfs -ra
```

and you may have to start or restart nfs:

```
systemctl restart nfs
```

4. Each of the render farm computers must mount the exported nfs target path. To see the exports which are visible from a client, login as root to the client machine and keyin:

```
showmount -e <ip-addr> #using the ip address of the storage host
```

5. to access the host disk storage from the other computers in the render farm, mount the nfs export on the corresponding target path: (be sure to check/verify before using any command line):

```
mount -t nfs <ip-addr>:/<path> <path>
```

where `<path>` is the storage host directory, and `<ip-addr>` is the network address of the storage host. Because all of the computers must have the same directory path, create that same directory path with the same uid/gid/permissions on each storage client computer ahead of time.

6. To make this permanent across reboots on the client nodes, add the following line to `/etc/fstab`:

```
{masternode}:/nfsshare /mnt nfs defaults 0 0
```

You can make this permanent on the disk storage host BUT the command lines shown, which were correct in January 2018 on Fedora, may be different for your operating system or in the future. In addition if your network is not up, there may be numerous problems. If you make a mistake, your system may not boot. To make permanent, add the following line to `/etc/fstab`:

```
192.168.1.12:/tmp /tmp nfs rw,async,hard,intr,noexec,noauto 0 0
```

You will still have to mount the above manually because of the *noauto* parameter but you won't have to remember all of the other necessary parameters. Depending on your expertise level, you can change that.

Later, to remove access to the storage host filesystem:

```
umount <path>
```

Be aware that you may have to adjust any security or firewalls you have in place. *Most firewalls will require extra rules to allow nfs access*. Many have built-in configurations for this.

Configure Rendering on Master Node There is 1 master node which is running the CINELERRA-GG gui and where the video will be edited and the command given to start up the rendering. Any number of client computers can be run from the command line only, so they can be headless since no X or any graphical libraries are needed. Of course, the CINELERRA-GG software must be installed on each of the client computers.

1. Assuming you already have CINELERRA-GG installed on the master node, start CINELERRA-GG by clicking on the icon or by typing the following command on the terminal screen: `/cinelerra_path/cin`.
2. Use the *File* pulldown `Settings` → `Preferences`, the Performance tab, to set up your Render Farm options in the Render Farm pane.
3. Check the *Use render farm* option. By default, once you enable the option of Render Farm, rendering is usually done using the render farm. Batch rendering can be done locally, or farmed.
4. Add the hostname or the IP address of each of the client nodes in the Hostname textbox and the port number that you want to use in the Port textbox. You can make sure a port number is not already in use by keying in on the command line:

```
netstat -n -l -4 --protocol inet
```

Next, click on the *Add Nodes* button and then you will see that host appear in the Nodes list box to the right. The `X` in the first column of the nodes box denotes that the node is active. To review the *standard* port allocations, check the `/etc/services` file.

5. Enter the total jobs that you would like to be used in the *Total job* textbox.
6. The default watchdog timer initial state is usually just fine but can be adjusted later if needed.
7. Click OK on the Preferences window when done.

Create Workflow While working on the master computer, it is recommended that you keep all the resources being used on the same shared disk. Load your video/audio piece and do your editing and preparation. Add any desired plugins, such as a Title, to fine-tune your work. You want to make sure your video is ready to be rendered into the final product.

Start the Client Nodes To start up the client nodes run CINELERRA-GG from the command line on each of the client computers using the following command:

```
/{cinelerra_pathname}/cin -d [port number]
# for example:
/mnt1/bin/cinelerra -d 401
```

This starts CINELERRA-GG in command prompt mode so that it listens to the specified port number for commands from the master node for rendering. When you start each of the clients up, you will see some messages scroll by as each client is created on that computer, such as:

```
RenderFarmClient::main_loop: client started
RenderFarmClient::main_loop: Session started from 127.0.0.1
```

As it completes its jobs, you will should see:

```
RenderFarmClientThread::run: Session finished
```

Render Using Render Farm After you have followed the preceding steps, you are ready to use the render farm. Click on `File` → `Render...` which opens the render dialog. The most important point here is to use for *the Output path / Select a file to render to* a path/file name that is on the shared volume that is also mounted on the clients. Click on OK to render. The CINELERRA-GG program divides the timeline into the

number of jobs specified by the user. These jobs are then dispatched to the various nodes depending upon the load balance. The first segment will always render on the master node and the other segments will be farmed out to the render nodes. Batch Rendering, as well as BD/DVD rendering, may use the render farm. Each line in the batchbay can enable/disable the render farm. Typically, video can be rendered into many file segments and concatenated, but normally audio is rendered as one monolithic file (not farmed).

Another performance feature which can use the Render Farm is *Background Rendering*. This is also enabled on the `Preferences` → `Performances` tab. The background render function generates a set of image files by pre-rendering the timeline data on the fly. As the timeline is update by editing, the image data is re-rendered to a *background render* storage path. The Render Farm will be used for this operation if it is enabled at the same time as the *background render* feature.

Assemble the Output Files Once all of the computer jobs are complete, you can put the output files together by using the shell script, *RenderMux* (from the menubar *scripts* button just above FF), if the files were rendered using ffmpeg (see Menu Bar Shell Commands).

If you want to remain within the open project in CINELERRA-GG, you can load these files by creating a new track and specifying concatenate to existing tracks in the load dialog in the correct numerical order. File types which support direct copy can be concatenated into a single file by rendering to the same file format with render farm disabled as long as the track dimensions, output dimensions, and asset dimensions are equal.

Finally if you want to use ffmpeg from terminal externally to CINELERRA-GG you can go to the directory where the rendered files are, it creates a text file `list.txt` containing the list of all our files:

```
file 'name.webm001'
file 'name.webm002'
...
file 'name.webm00n'
```

and finally give the command

```
ffmpeg -f concat -i list.txt -c copy output.webm
```

7.7.5 Quick and Easy Render Farm Setup – The Buddy System Way

These steps are for quickly setting up render farm with the least amount of additional system work, but it is non-optimal. It is useful in situations where a few people all show

up with their laptops to work together on the same video/audio file and you don't want to bother setting up NFS for a shared disk.

1. Make sure the CINELERRA-GG program is installed on all of the computers and the network between the main computer and the client computers is working. Use the same version if possible.
2. Load your video file on the master node and use `File → Save as...` to save it to `/tmp`.
3. Move that same file with the same name to `/tmp` on all of the client computers via rsh or sneaker net – the ONLY reason you are doing this is to avoid having to set up NFS or Samba on the buddy client laptops that show up!
4. Edit your video/audio file to get it the way you want it and add the plugins, such as a Title, etc.
5. Check for a set of unused ports in `/etc/services` file, if username is root usually 401 — 425 are available; if non-root, then 10650 — 10799.
6. On the master computer, in `Settings → Preferences, Performance` tab:
   - check the box *Use render farm*
   - keyin localhost for the hostname or an ip address of the buddy client node
   - keyin the desired port number for each client; and use *Add Node* for each host
   - set total jobs to the number of client computers +1 multiplied by 3 (or proportion to client speeds)
   - check OK
7. On each buddy client, create a job for each port:

```
/{cinelerra_pathname}/cin -d port#
```

8. On the master, bring up the render menu and name the output files, for example `/tmp/myoutput.webm`.
9. The client nodes output results will be on their local `/tmp` filesystems so you will have to again use *rsh/ftp* or *usb sneaker net* to move them over to the main computer. File names will be the render job output file name with port number tacked on (e.g. `/tmp/hb.webm001...webm005`).
10. Load the files by concatenate to existing track on the master node or use RenderMux shell script.

7.7.6 Troubleshooting Tips and Warnings

If you have problems running the Render Farm. Here is a list of items to check.

- CINELERRA-GG must be installed on the master node and all client machines.
- It is best to have the same username available on all nodes to avoid problems with access rights.
- Check file permissions and ownership to ensure that the clients all have access.
- If a node does not have access to an input asset it will not die, but just display error messages.
- If a node can not access an output asset, the rendering will abort.
- A port in use when stopped may take up to 30 seconds to time out before you can restart the jobs.
- Each of the port combinations have to be unique across clients, and not already in use in the network.
- CINELERRA-GG load balances on a first come, first serve basis. If the last section of the video is sent to the slowest node, the render job will have to wait for the slowest node to finish. It would be better to start on the slowest node with the earlier section of the video so keep that in mind when designating port numbers.
- If not running as root, a port number in the higher range of 10650 and above must be used instead of the 401+ range.
- The master and client jobs on the ports do not go away so if you want to stop them, you will have to kill them via: `kill PID#`.
- Check to see if there are services listening on the ports to use: `netstat -n -l -4 -protocol inet`
- There is a watchdog timer in CINELERRA-GG and if there is no response from a client in the designated number of seconds, it will kill the render job.
- The *localhost* should exist as 127.0.0.1 in `/etc/hosts` and as the `lo` network device in ifconfig.
- If the job loads become unbalanced, you may want to *reset rates* to start over for new framerates.
- If jobs are split in a key section on the timeline, you may wish to *use labels* to prevent this.
- Adjust the *Total jobs to create* for different videos that are considerably different in size. You should not create so many jobs that each one is very small.

- For testing purposes, you may want to start a client in the foreground using `-f` instead of `-d`.
- If one of the client computers is unavailable, check to see if there is an `X` to the left of the `nodename` in the Nodes listbox. Check the `X` to disable it which sets ON to OFF.
- A red message in the lower left hand corner of the main timeline that reads *Failed to start render farm* often means that the client CINELERRA-GG programs were not started up.
- A message of `RenderFarmWatchdog::run 1 killing server thread #address#` means that the client did not respond in time. You can adjust the timer in `Settings → Preferences, Performance` tab.
- When you get the message `RenderFarmClient::main_loop: bind port 400: Address already in use`, use a different port. See `/etc/services` for free ports.
- A message of `RenderFarmServerThread::open_client: unknown host abcompany` means that the hostname of abcompany is not in `/etc/hosts` so you will have to add it or use the ip address instead.
- There are numerous error messages associated with file *open/close/status* or problems with the file that should be dealt with according to what is printed out.
- Other illustrative messages may be shown such as: `RenderFarmClientThread::run: Session finished`.

And here are a couple of more tips for making Render Farm specific for your setup.

- Because *index files* speed up displaying the video you may want to share these files with the clients on a shared filesystem. More information on index files configuration is outlined in 16.5.3.
- Or, one of the convenient features of CINELERRA-GG is the redirection of the path via `CIN_CONFIG` as in:

```
CIN_CONFIG="/<shared_file_pathname>/<filename_such_as_.bcast5>" cin
```

This means that you can make project related configurations that do not impact the default `$HOME` config. You can either export your default `$HOME` config or the `CIN_CONFIG` config to use on the render farm.

Warnings If one of the render farm computers is connected to the internet, you should use a firewall to maintain the safety of all of the computers. The ports have to be reachable for the intranet but you do not want the ports to be open to the outside.

CHAPTER 8

Keyframes

The word *keyframe* has at least 3 contextual meanings in the NLE environment. First, the oldest meaning, is the *I-Frame* definition used in codecs algorithms. These are *key* frames that begin a new sequence of pictures (GOP= Group Of Pictures), and are anchor points for repositioning (seeks) . Next are the automation parameter data points. These are usually input to primitive math forms, like translation and zoom. And last are blobs of data that are chunks of parameters to plugins that can do almost anything. The data can be a simple value, like a fader value, or more complex like a group of points and colors in a sketcher plugin keyframe. The word keyframe has changed a lot in meaning. In the context of CINELERRA-GG, keyframes are data values that have been associated to media on the timeline which affect the media presentation. So a keyframe no longer refers to a frame, but to a position on the timeline (timecode).

In CINELERRA-GG, there are two general types of keyframe data, *automation keyframes* (autos) which are drawn as colored lines and box icons overlayed at a point on a media track, and *plugin keyframes* which are drawn as gold key symbols on a plugin bar of a track. Auto ← Keyframe → Plugin

It is important to understand the fundamental difference between CINELERRA-GG *Autos* (zoom, pan, etc.) is that *Autos* are interpolable in any way (linear, bezier, etc.) while *Plugin keyframes* may be interpolable in various ways or not be interpolable at all.

8.1 Automation Keyframes / Autos

The *autos* are created by clicking on an *automation curve* to establish the time position for the new keyframe anchor point. The basic nature of these simple auto values make them primitive operations that are easy to apply when needed.

There are many automation curve types, and most are not normally visible or clickable. To make them visible, use the `View` pulldown, or open the `Window → Show Overlays` . This window allows toggling of the parameters in the *View* pulldown but is more convenient because you can leave the window up to change values quickly. If all of the automation curves are turned on, the timeline will be quite cluttered, and so usually only the parameters of interest are enabled during use. When keyframes are selected, they are drawn on the timeline over the tracks to which they apply. The keyframe is represented on the timeline as a little square on the curve, for example as in fade, or as a symbol as

in a mask. This square, timeline attachment point, can be used for positioning by clicking on a keyframe anchor and using drag and drop to set the new position.

The automation keyframes include:

mute/play audio; camera translation x,y and zoom; projector translation x,y and zoom; fade blending; audio panning; overlay mode; mask point sets and sampling speed.

Except for the mask auto, the values are all simple numbers. Mute is different from the other autos in that it is simply a toggle of either on or off. Mute keyframes determine where the track is processed but not rendered to the output. An example usage would be to use auto keyframes to fade in a clip by setting the transparency to 100% at the first keyframe and adding another keyframe 5 seconds later in the timeline with a transparency of 0%.

The *Keyframes* pulldown on the main timeline is used for Cut, Copy, Paste, Clear, Change to linear, Change to smooth, Create curve type of Smooth, Linear, Tangent, or Disjoint, Copy default keyframe or Paste default keyframe. If you right click on a curve keyframe on the timeline, a set of options popup including the choices *keyframe type* (such as Fade, Speed, etc.), Hide keyframe type, Delete keyframe, Copy keyframe, smooth curve, linear segments, tangent edit, or disjoint edit.

Usually, the use of the keyframe values are more pleasing when the data varies smoothly between keyframe anchors on the timeline. This is useful in many cases that are familiar, like a video fade in/out, or audio pan between channels. To make the auto value change smoothly as the media is played, the keyframes auto values are points on curves that are created according to the design of the effect. Most of the primitive types can create anchor points on curves that are piecewise linear, smooth, sloped, or broken at the keyframe anchor points.

Curve smoothing is called *interpolation* and it uses keyframe point values and control values that determine how the curve will react at the time the media is played or rendered. Interpolation uses 2 keyframes to create a set of intermediates which are used as active values between the *previous* and *next* keyframe anchors on the timeline. The way the intermediate data is generated depends on the type of curve used to invent these values. CINELERRA-GG interpolates the intermediate values making the change happen smoothly and gradually over time. The simple linear mathematical formula for interpolation is: $a \times (1-t) + b \times t$ where $0 \leq t \leq 1$ uniformly.

Autos are handled inside the main executable. Therefore CINELERRA-GG knows exactly what a particular Auto does, how to interpret it, and how it can be interpolated.

8.2 Using Autos

The first click on the curve, creates a keyframe which you can click drag on to reposition. The second click at a later position, generates the smoothing by creating a smooth ramp. Ctrl-dragging on a keyframe round control point handle changes the value of either the input control or the output control. This affects the sharpness of the curve. While the input control and the output control can be moved horizontally as well as vertically, the horizontal movement is only for legibility and is not used in the curve value. When you Shift-drag on a timeline curve, the keyframe snaps to the value of either the next or previous keyframe, depending on which exists. It will snap up or down depending on direction of movement. This lets you set a constant curve value without having to copy the next or previous keyframe.

To make it easier to navigate curve keyframes, since there is not much room on the timeline for a wide range of curve values, you need to zoom the curves in and out vertically to have any variability. This is done by 2 tools: the automation fit button, Alt-f, and automation zoom menu which is seen at the bottom of the main window (figure 8.1). The automation fit button scales and offsets the vertical range so the selected curve area appears in the timeline. If a region of the timeline is highlighted by the cursor, only that region is scaled. In/out points do not affect the zoomed region. The automation zoom menu manually changes the vertical scaling of the curves in multiples of 2. Click on its tumbler to change the zoom. Alt— ↑ and Alt— ↓ change the automation zoom from the keyboard.

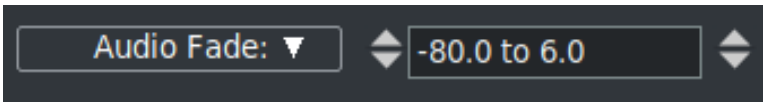


Figure 8.1: *Automation Audio Fade menu with tumbler*

Other mouse actions have the following effects:

- Double left mouse click on a curve Fade or Speed line will create ganged keyframes so that there is a keyframe on each of the tracks in the exact same position.
- Left mouse click on a keyframe position will show the numerical value in a colored tooltip-like box.
- Right mouse click on the curve type line will bring up the option of *Hide keyframe type*. This provides the same functionality as disabling the keyframe type in the *View* pulldown menu. Often it helps to use this in order to be able to see other things on the timeline once it gets cluttered.
- Fade and speed allow for setting a specific value for the keyframe using the following:

    1. click using the right mouse button on one of the auto speed keyframes on the timeline;

2. a popup menu comes up with speed or fade auto type as the first menu item;
3. click on speed or fade and a colored slider bar will appear (default speed = orange; fade = pink);
4. click the slider, press and hold the left mouse button and move the slider to update the value or simply type in a value in the textbox followed by pressing the Enter key or click the checkmark;
5. a tooltip shows the keyframe value;
6. release the button, the slider will be deleted from the canvas, and the value will be updated.

You can click mouse button 3 (RMB) on a keyframe box and a menu pops up with the first menu item showing the keyframe type. The top menu item can be activated for immediate access to update the automation keyframe value. Some keyframe types, which have values that can be manipulated in another way than by dragging the color coded line, now show up with a different colored background to make them more visible. Keep in mind that Zoombar ranges/values must be set to appropriate values when working with specific keyframe types, such as Fade or Speed. If you do not see the auto line in the visible area of the video track, try the key combination Alt-f or select the speed in the *Automation Type* drop-down menu at the bottom of the main window. To the right of this field is *Automation Range* where you can set the display ratio of these lines. Simply change the values until the lines are visible again.

Figure 2.39 and figure 8.2 shows several color coded lines for different keyframes and specifically the slider bar for the Fade keyframe. It is in the same color as the color coded keyframe type line which is the same color which would be shown in the *Show overlays* window figure 2.38.

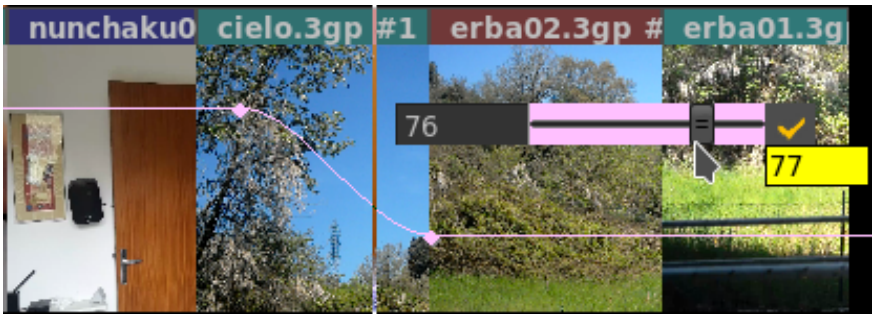


Figure 8.2: *Fade curve with pink colored box Keyframe and Slider bar*

In the *Editing* section of `Settings → Preferences, Interface` tab there is *Keyframe reticle* with options of Never, Dragging, or Always. This is used to help in checking edit alignment across tracks. (A reticle is a sighting line used to line up visual items, like cross hairs in a eyepiece.) The appearance and function of sighting lines can be changed when dragging auto keyframes. To see the effect, create some fader autos and drag a few to see the reticles drawn — you will see something similar to the next screencast (figure 8.3).

Always renders a line over all plugins, and *dragging* only over the drag icon. *Never* draws nothing. The default is *dragging*.

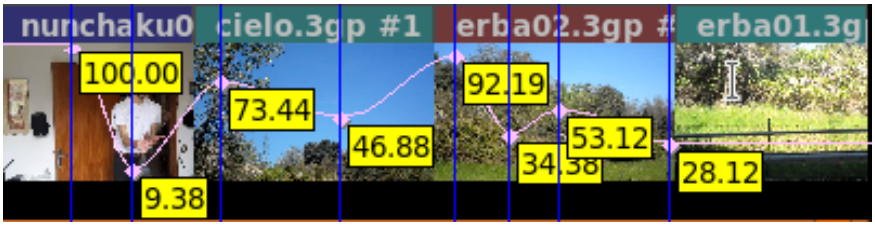


Figure 8.3: *Fade Auto with Always set for easy to see numeric values*

Control points allow for setting the slope of auto curves and then subsequently adjusting that slope (figure 8.4) . To modify a current keyframe you just right mouse it and change to either Tangent or Disjoint edit. In the screencast to the right, the Fade Auto has pink colored curves and control points are seen as dashed lines next to the keyframe box with black filled circles on each end of the line. Use the Ctrl key with the left mouse button to modify the control point lines.

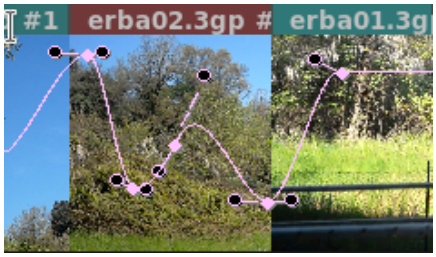


Figure 8.4: *Fade Auto with Controls*

See D.3 for a pratical example.

8.3 Speed / Fade Automation Usage and Auto Gang

Speed automation resamples the data at a higher or lower playback rate. Speed automation can operate on all tracks of the same type, either video or audio, with a single click; or all tracks, both video and audio, with a double click. The curves to be affected must be both armed and ganged in the track patchbays to be included in the operation. You can see the advantage of having different media types ganged the same so that they will remain synchronized. However, if as recommended you have Settings of *Align cursor on frames* set, video will always snap to a frame, whereas audio does not. Because the smallest pieces are audio and not video, when video is present in order to maintain synchronization be sure to do the double click on the video track. The single/double click feature also works in the same manner for Fade Automation.

The conversion algorithms that calculate the duration are approximate, not exact. Speed is not checked when inserting media on a track that already has speed adjustments so duration will not be corrected in that case. You can not do negative time and you can not play backwards. When working with edits and the speed auto, if you make a *cut* in a section, the speed curve values should not change over the remaining playable edits. It is possible to easily add *disjoint* curve points at the boundaries of the edit to preserve the curve shape.

Steps to demonstrate Auto Gang of Speed/Fade on all of the audio and video tracks are listed below:

1. Use the pulldown menu *View* on the main canvas and ensure that there is a checkmark left of the Speed or Fade selection.
2. Double click and hold button 1 (LMB) on any point of the Speed line on any track of the main track canvas and drag the handle where you want it, then release button 1 (LMB).
3. Note how all video and audio boxes move together simultaneously for synchronization.
4. You can adjust the sampling rate (speed/fade) at any time by doing the same single/double click on a keyframe.
5. To move the location left or right, hold down button 1 (LMB) while moving it back and forth.
6. Use the status bar in the lower left hand corner to see position and playback rate.

Releasing button 1 (LMB) ends the current dragging operation, but single clicking on a speed/fade keyframe handle again will restart the drag operation on tracks of the same type. Double clicking will select all armed tracks which have handles in the exact same cursor position. It is sometimes difficult to get the desired value for the speed and it may take a few tries to get to the desired endpoint, but you can use an auto slider bar to get better control for more precision. This is not ganged so you will have to do this on each track to achieve the same value.

An easy way to get an exact position is to set the *Automation range* in the bottom bar of the main window to the low and high ends of the desired range. For example, 1.000 to 50.000, which makes it so you can drag the speed/fade handle all the way to the left for 1.000 or all the way to the right for 50.

Figure 8.5 shows orange keyframes and lines for the ganged speed automation on all video/audio tracks. The figure shows the orange speed slider bar triangle with the current value displayed via the tooltip. Note the status bar numbers in the lower lefthand corner displaying 01:08 seconds as the location and 2.05 as the playback rate.

One other helpful item is a small colored indicator in the zoombar *Automation Type* box when you have clicked on an auto line. It will be the same color as that line so you can easily see out of the corner of your eye what that line refers to. This is especially useful for the Camera and Projector X,Y,Z lines which are often right on top of each other. In addition the status bar will have the colored indicator to the left of the last auto value when that status is visible.

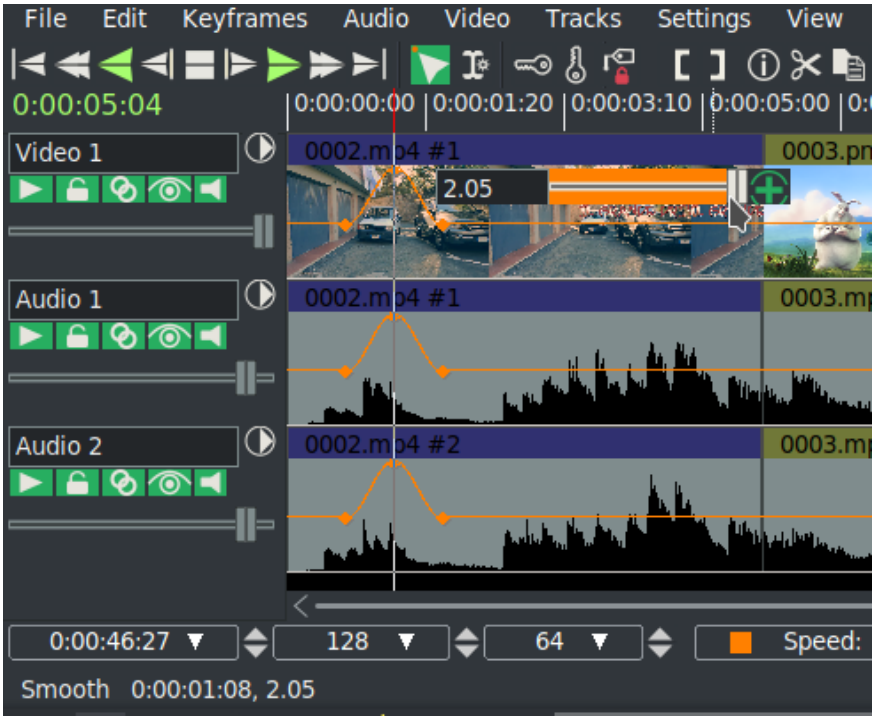


Figure 8.5: *Ganged Speed Auto on all tracks and Speed auto with orange Slider bar*

8.4 Bump autos

Bump autos are a kind of floating point keyframe that differs from the rest of the floating autos, like smooth and linear, in that they have two values:

a left edge and a right edge value.

The presence of a *bump* creates boundaries to regions on the timeline. The region begins at that bump, and extends to the next bump auto. If the bump *span* toggle is set, then adjusting the left/right edge value will raise or lower all of the auto values in the bump region. For example, if you put a speed bump in at 1.0 secs and another at 2.0 secs on the timeline, and adjust the right edge of the first auto to speed 3.0, then the first region of playback (between 0.0 and 1.0 secs) will play at normal speed, the next region (between the speed bumps) will play at 3x, and the last region (after the 2.0, which moved to 1.333 after speed up) will play at normal speed (figure 8.6).

Adjusting the *bump* edge values when *span* is on modifies all of the keyframes between edges of bumps, or to the beginning or end of media if there are no bumps to end the bump region. Turn *span* off if you only want to change the edge value without modifying the region values.

Technical Insight: The *bump* is a kind of FloatAuto, an element in an automation list that is one of the 12 automation types of keyframe lists:

Figure 8.6: *range from left keyframe to right keyframe*

mute, cam_x,y,z, proj_x,y,z fade, pan, mode, mask, speed

which are part of the Track. A FloatAuto is one of 4 auto *value* types that are stored in the auto at the keyframe position: `float, mask, int, pan`.

A FloatAuto stores a floating value for the automation at a position on the track. When a FloatAuto `get_value` is invoked at a track position, the position is used to access a previous and next auto list element values based on the *current* position. The returned value is generated using one of 5 modes of curve interpolation: smooth, linear, tangent, disjoint, bump.

When a FloatAuto is created, it gets a list default value. This varies by auto type speed=0, fade=100, ..., and at position 0, but this is almost always immediately changed to a value specified for the current position, a value/mode you specify in the sliders and drag values of the GUI updates.

The best way to create bump keyframes is highlight a region and then:

`Keyframes → Create curve type → bump`

`Keyframes → Create keyframes → speed` (or other)

However, you can also create them by clicking or double-clicking on the curve, then:

`RMB → bump`

`RMB → speed` (or other)

Make sure that the keyframes at the edges are bump type, span is active and that the left edge is set to the right (if we want to change the next interval). Changing the *Left/Right Edge* and *Span* options will involve all the autos of the curve. Actually the changes concern the whole curve, not the individual keyframes.

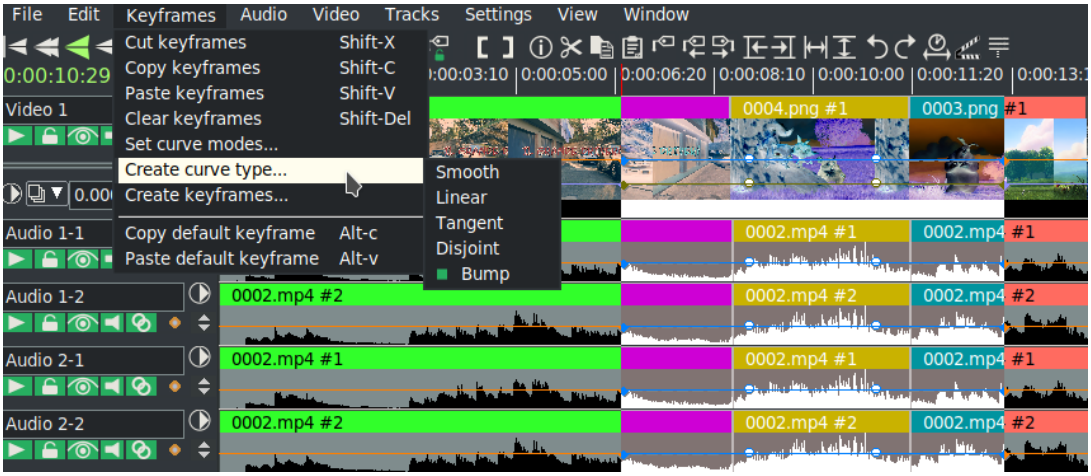


Figure 8.7: *Bump autos creation*

Warning: bump autos works only via slider. Dragging the keyframe with the mouse makes you lose the *right/left edge* and *span* settings and behave as normal autos (figure 8.8).

Figure 8.8: *Speed autos slider*

With using the sliders on a track, the keyframes ganged track are changed if the positions match; tracks can be excluded by switching the track gang mode off. Another way to edit all armed tracks of all types at the same time is as follows:

- disable *Master track* of all tracks except the first one, on which we will work.
- switch to *gang media* mode. Now only the Master track is visible, but every change made on its keyframes is also applied to the other tracks. (see Displaying tracks: Ganged mode)

In the Camera and Projector tools configuration window you will find the *Bump curve*, *Span* and *Rigth/Left Edge* buttons. The Camera-XYZ and Projector-XYZ curves can also create and use Bump autos. (see Camera and Projector Menu)

Note: do not confuse the autos bump span button with the keyframes span button of the plugins (on MenuBar); they are different and work differently (see Allow Keyframe

Spanning). With the plugin keyframes, the update is done by first creating a textual `diff` on the xml values of the plugin keyframes, then applying that diff to all of the keyframes selected in the *span* range. The bump FloatAuto `bump_update` subtracts the old and new values to create a `diff` that is added to all of the values in the [*right_edge* ... *auto values* ... *left_edge*] region created by bump autos and media endpts.

8.5 Plugin Keyframes

The Plugin keyframes are structured. The individual data values are named parameters to the keyframe function. For example, the hue plugin has keyframe parameters of hue, saturation, and value. Each plugin has its own parameters, and what they do depends on the plugin. Most of the time, it is pretty obvious what the value controls, like the audio gain plugin with the level parameter. Some of the plugins have a wide variety of controls, like the titler which can setup a wide number of controls, like formats, fonts, styles, placement, and so on.

Plugins also may use interpolation to smooth the data between the keyframe paramaters on the timeline. Just exactly how this is done varies with the plugin and is not always easy to predict, but this is usually due to the nature of the keyframe parameter data. For example, it does not make sense to interpolate which font is in use.

Plugin keyframes are handled inside its own plugin. CINELERRA-GG knows nothing about internal functionalities inside plugins, either external (such as ffmpeg) or native plugins. And has no chance to do any interpolation. Only the plugin can interpolate its own keyframes. And if you succeed to add plugin keyframe interpolation in some particular plugin, all the other (even similar in functionality) plugins will know nothing about it. So, each one out of dozen of plugin keyframes has to be implemented individually.

8.6 Default Keyframe

For plugins, there is a special hidden keyframe, called the *default keyframe*, that is used when no previous keyframe exists. It is like keyframe zero on the timeline, and it is persistent and shared on all sessions. The intent is to make a parameter set that is likely to be reused on all initial instances of the plugin. An example may be to color correct a set of media that was taken in low light, or needs resampling to be played correctly. The default keyframe is *off the bar on the left side* of the plugin title bar and can not be seen. It is used when there is no *previous* keyframe for its default values.

It may be useful to create a default keyframe which has specific desirable values for later use. To do this, set the timeline to position 0 and be sure to disable *generate keyframes while tweaking*. This will create a default keyframe at the beginning of the timeline which contains global parameters for the entire duration. Or if you have copied a non-default keyframe via *Keyframes* pulldown *copy default keyframe*, it can be stored as the default keyframe by calling `keyframes → paste default keyframe`. After using paste de-

fault keyframe to convert a non-default keyframe into a default keyframe, you will not see the value of the default keyframe reflected until all the non-default keyframes are removed.

The `keyframes → copy default keyframe` and `keyframes → paste default keyframe` allow conversion of the default key-frame to a non-default keyframe.

`Keyframes → copy default keyframe` copies the default keyframe to the clipboard, no matter what region of the timeline is selected. The `keyframes → paste keyframes` function may then be used to paste the clipboard as a non-default keyframe.

Typeless keyframes enabled under the *Settings* pulldown allow keyframes from any track to be pasted on either audio or video tracks. Ordinarily audio keyframes can only be pasted to another audio track and video keyframes can only be pasted to another video track.

8.7 Keyframe *Edit Params* for Plugins

Keyframe values can be set using the various plugin *plugins: show controls* (magnifying glass) icon on the plugin track (figure 8.9).

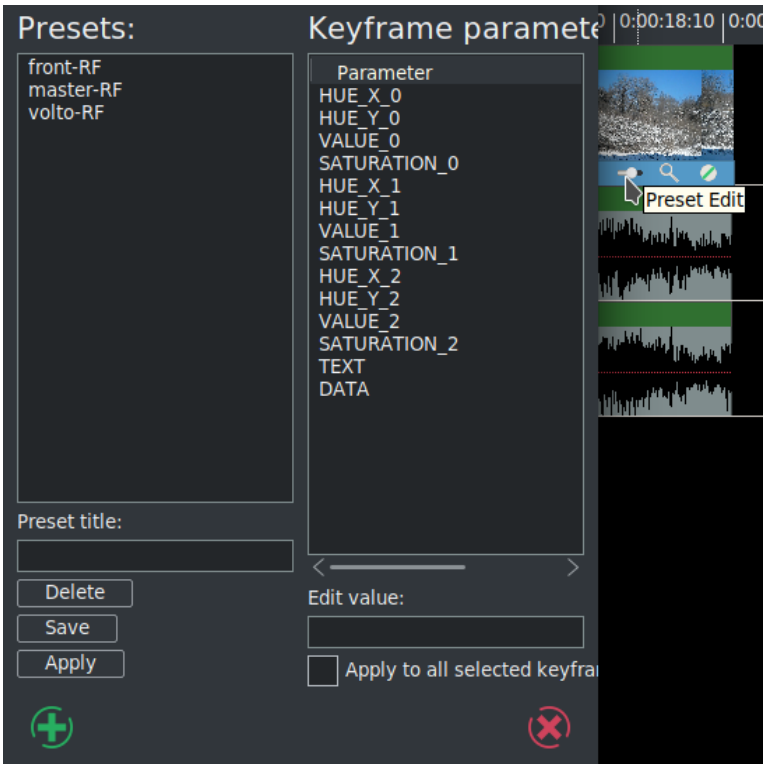


Figure 8.9: *Keyframe Parameters window for a Color 3 Way plugin added to the video track*

It is possible to see all of the keyframe data in a raw format using the *Edit Params* popup

menu item which you will see when you right mouse click the keyframe icon on the timeline. The keyframe data is stored in xml format, and the *edit params* feature allows you to view and modify the xml data directly. This normally should not be necessary since the plugin's control gui displays the intended parameters, but this will let you view and specify just about anything that can be specified in xml. There is no validation checking of the modified data.

8.8 Generate Keyframes while Tweaking / Automatic Keyframe Mode

Tweaking is defined as changing a parameter while playing – it can be small changes or small motion increments performed in a series. These changes are recorded as a series of new keyframes on the timeline. Enable automatic keyframe mode by enabling the automatic keyframe toggle, that is *Generate keyframes while tweaking*. In automatic keyframe mode, every time you tweak a key-framable parameter it creates a keyframe on the timeline (visible by activating the respective *Camera x, y, z* and *Projector x, y, z* curves). Since keyframes affect playback, you should enable generate keyframes just before you need a keyframe and disable when your parameter changes are complete. So turn on when ready to make changes and turn off when done!

Also, before making a change, be sure to check the *View* pulldown and make the desired parameter visible with the enable checkmark. The location where the automatic keyframe is generated is under the insertion point. If the timeline is playing back during a tweak, several automatic keyframes may be generated as you change the parameter. When automatic keyframe mode is disabled, adjusting a parameter adjusts the keyframe immediately preceding the insertion point. For example, if two fade keyframes exist and the insertion point is between them, changing the fader changes the first keyframe.

8.9 Compositor Keyframes

Camera and projector translation is represented by two parameters: x and y, making it difficult to adjust with curves. CINELERRA-GG solves this problem by relying on automatic keyframes. With a video track loaded, move the insertion point to the beginning of the track and enable automatic keyframe mode. Move the projector slightly in the compositor window to create a keyframe (visible by activating the *Projector x, y, z* curves). Then go forward several seconds. Move the projector a long distance to create another keyframe and emphasize motion. This moves the projector box to its new position. If you create more keyframes, we can move the box several times. Once all the desired keyframes are created, disable automatic keyframe mode.

Dragging the auto curve with tweaking off, adjusts the previous keyframe. With tweaking on, if no keyframe is at the timeline position, then a new one is created at that position and the modification value is applied. If you are halfway between two keyframes, the value from the previous box position will be altered, while the second is not changed. The

video does not appear to move in step with the first keyframe. This is because halfway between two keyframes, the projector translation is interpolated. In order to set the second keyframe you will need to move after the second keyframe.

Image translation, motion direction, and speed determine the results. Motion varies based on the interpolation type that is set, for example, if set to Linear, the result will be uniform straight line motion. Smooth is the default so that interpolation will generate curved lines with control points. Ctrl-drag the endpoint of the control handle to adjust the curved slope.

8.10 More about Editing Keyframes

Keyframes can be shifted around and moved between tracks on the timeline using similar cut and paste operations to editing media. Only the keyframes selected in the View menu are affected by keyframe editing operations.

An often used, keyframe editing operation is replication of some curve from one track to the other to make a stereo pair. The first step is to solo the source track's record patch by Shift-clicking on the *arm track* icon in the patchbay. Then either set In/Out points or highlight the desired region of keyframes. Go to `keyframes → copy keyframes` to copy them to the clipboard. Solo the destination track's record patch by Shift-clicking on it and go to `keyframes → paste keyframes` to paste the clipboard. Another common application for keyframe modification is to highlight a region on the timeline which contains multiple keyframes that you want to modify. Then when you adjust a parameter or set of parameters, the change will be applied to all keyframes within the selection instead of a new keyframe being created. This only works when the keyframe stores multiple parameters and only for mask and effect keyframes. Other types of keyframes are generated as usual.

And there is an easy way to delete keyframes besides selecting a region and using `keyframes → clear keyframes`. Click-drag a keyframe before its preceding keyframe or after its following keyframe on the track. This is the only way you can simultaneously delete keyframes on ganged tracks.

8.11 Allow Keyframe Spanning

Allow keyframe spanning is enabled on the timeline by clicking on the icon that is on the right side of the *Generate keyframes while tweaking*. With this enabled, when you create a drag selection and you modify a value in a plugin then everything in the selection gets modified the same. It uses the previous keyframe and if there is no previous keyframe, then the default keyframe in your `$HOME/.bcast5` definitions is used.

This is particularly useful in plugins that have the *Drag* box enabled. You can select the area on the timeline where all of the keyframes exist with the drag enabled, turn on *Allow*

keyframe spanning, and then modify the menu for that plugin to turn off the drag.

CHAPTER 9

FFmpeg Interactions

CINELERRA-GG uses ffmpeg for decoding and encoding media, thus there are many opportunities available to manipulate options.

9.1 FFmpeg Early Probe Explanation

When you open media, a series of libraries and codec functions are used to *probe* the data, to see if it can determine the type of file format and codec parameters needed to correctly decode the file. If ffmpeg probes early – *Try FFMpeg first* is in effect for the FF button (green icon) – it will usually find some way to try to decode just about any contemporary media file. But there are some times that the built in codecs are actually a better choice. A lot of this may fall into the category of personal preference. For example, some may prefer the mpeg library in the CINELERRA-GG code over the ffmpeg code because it has more decoding capability and seems to be more robust when the media is damaged. In that case you will want the FF button to read *Try FFMpeg last*.

To summarize, if ffmpeg probes early, you will never get to use the built in libraries, and if you want to skip over buggy old libraries, use ffmpeg early probe enabled so that the newest code will be tried first. The green icon is located in the upper right hand corner of the main window.

When the icon is green, ffmpeg probes early is enabled and you will see it reads *Currently: Try FFMpeg first* when moving over the icon button in the upper right hand corner of the screen. When the icon is capital "C", ffmpeg probes early is disabled so that ffmpeg probes late and it reads *Currently: Try FFMpeg last*. The initial default state of the icon is on, that is, ffmpeg probes first. Suggestion is to leave it on except in a few special cases where it may be better to have early probes disabled. When you mouse over the main menu icon toggle button, the text displays ffmpeg's *Currently* set position. Just left mouse click to change to the other setting. The ffmpeg early probe state is saved between sessions and is also affected by choices made in Probe Order (refer to section Probe Order when Loading Media). It is important to note that the various file indexes may need to be rebuilt if you change which codec is being used to decode the file. There is a warning popup to remind you when you change the default ffmpeg early probe state (unless you have checked the box to no longer show the warning). You can easily rebuild the index for a specific media file by going to the Resources window, right mouse click on that media, and choose `Rebuild Index` from the popup choices.

Figure 9.1 shows (1) greenish colored icon in upper right hand corner of main window indicating that ffmpeg early probes is enabled; (2) *Try FFMpeg last* indicator message for ffmpeg early probes enabled (note that the icon is different because you highlighted the icon); and (3) "C" icon indicates ffmpeg will be used last and you are changing the behavior so that CINELERRA-GG warns you accordingly.

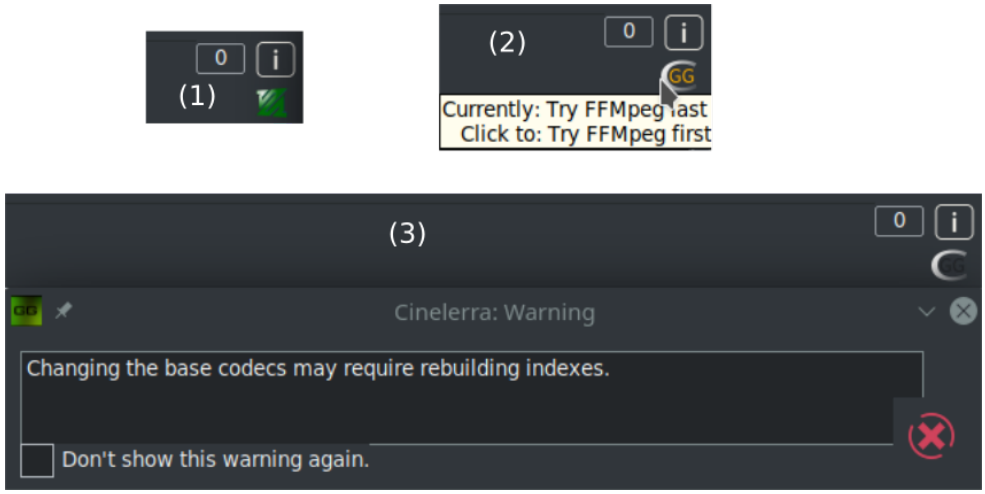


Figure 9.1: *The three icons of FF Probe*

9.2 How to Create FFmpeg Options Files

This section describes how the FFmpeg options files work for decoding and encoding and goes into great detail. It will make more sense if you look at CINELERRA-GG's ffmpeg config directory and the CINELERRA-GG menus at the same time. It is meant to include everything necessary for complete understanding. You will be able to personalize your own options files without knowing all of the information included below if you know the basics. The word encoding is used interchangeably with the word rendering. The possible combinations for ffmpeg options files are literally combinatorial – that is a lot (factorial!). The allowed media file format / codec choices are much more flexible than you might realize. When the ffmpeg design was initially added, some parameter files which describe the choices which the program uses had to be created. There are way too many to enumerate in the deliverable CINELERRA-GG package. Some quite detailed information for how ffmpeg options work is given here and hopefully, enough basics for simple understanding. It may all seem complicated at first, but will become obvious. AppImage does not provide this capability unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds.

9.2.1 File naming convention

In CINELERRA-GG's ffmpeg configuration directory you will see files as listed and described below. File type and extension names are the key for CINELERRA-GG's use of ffmpeg. Basically the `.opts` file extension represents options; `.dfl` represents defaults; and all the rest

are media types. For example one media type is quicktime so that `*.qt` file names would be the *quicktime* choices. In the file names below, *ext* refers to a set of files with file names matching the `*.ext` file extension. And *typ* refers to a type of format / codec combination used, that is, the media type. AppImage does not provide this capability unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds.

In the ffmpeg configuration directory there are a series of options files used when encoding or decoding audio or video. They are read in the order from top to bottom and only the files needed for the current operation are added to the active configuration.

ffmpeg/ffmpeg.opts	global ffmpeg options, always used
ffmpeg/decode.opts	global decoder options, used when opening existing files for decoding
ffmpeg/encode.opts	global encoder options, used when creating new files for encoding
ffmpeg/audio/audio.opts	audio encoder options, used when creating audio streams
ffmpeg/video/video.opts	video encoder options, used when creating video streams
ffmpeg/plugin.opts	parameters for ffmpeg filters as audio/video plugins

Decoder options: Normally, only `ffmpeg.opts` and `decode.opts` are used when reading/decoding files, but may be specialized if a `<path>/media.opts` exists for a given `<path>/media.ext` file. For example, if you want to only fail on fatal errors and to always use the video filter, edgedetect, when working with your media file `dreaming.y4m`, then create a file `dreaming.opts` in the same directory with the contents of *loglevel=fatal* on the first line and *video_filter=edgedetect* on the next. These specialized settings will override the defaults. The fatal loglevel is especially handy for lesser quality media.

Encoder Options: Within the audio/video subdirectories of the first level ffmpeg directory, the `typ.ext` files are for encoder (rendering) setups.

ffmpeg/audio	directory of audio encoder settings
audio.opts	options used by all audio encoders
typ1.ext, typ2.ext, ...	are all *.ext type choices for encoding audio
ext.dfl	contains the default selection used when ext is first selected
ffmpeg/video	directory of video encoder settings
video.opts	options used by all video encoders
typ1.ext, typ2.ext, ...	are all *.ext type choices for encoding video
ext.dfl	contains the default selection used when ext is first selected
ffmpeg/format	presets needed to initialize audio / video formats

9.2.2 Option File Format / Content

For the option files a specific format must be followed in creating the file content. AppImage does not provide this capability unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds. In `typ.ext` encoder parameter files, the first line is defined as:

```
      muxer codec
(or)  muxer codec | bitstream filter [ bitstream filter options ]
```

where the | represents piping the codec data through the bitstream filter. The rest of the lines in the file should look as follows:

```
      # in column one is a comment
      id1 value1
(or) id2 = value2
```

Only one equals sign is allowed and it is just for readability. There may be any number of id/value pair lines in a media definition, including zero. A typical line might be:

```
      bitrate 4000000
(or) bitrate = 5000000
```

There are 4 special id's recognized by CINELERRA-GG which cause special processing. They are:

duration overrides the probe duration when opening media for decoding

video_filter adds a video stream filter, eg. edgedetect,... at the stream level

audio_filter adds an audio stream filter, eg. echo,... at the stream level

loglevel sets the library logging level, as quiet, panic, ... verbose, debug

All other id's should be in the ffmpeg documentation, and correspond to the global, muxer, and codec option names and values used by ffmpeg. For example to set the aspect ratio to 4:3, use:

```
aspect 4:3
```

Below shows an example: `decode.opts` which is used when the ffmpeg decoder is initialized.

```
# apply at init decode
loglevel=fatal
formatprobesize=5000000
scan_all_pmts=1
threads=auto
```

The encoder options you see in the CINELERRA-GG menus depend on the files in these directories, NOT THE CODE. If you add files, you will get to use more variety.

In the CINELERRA-GG directory, which contains the ffmpeg configuration folder, there are the choices the program uses. When you open an ffmpeg format popup dialog, the listbox contains all of the codec types which are identified by the `file.ext` extensions. Decoding has only a few options, since the ffmpeg file probes determine most of the options by looking at the media being opened, but encoding media requires a lot of setup. Below are some of the folders and files used to determine the configurations used by ffmpeg to decode and encode files.

These extensions create audio / video media classes:

`dvd  m2ts  mkv  mp3  mp4  mpeg  qt  pro`

which become the choices in the render pulldown menu.

So if you want to create a *mov* codec class, add two new files to the ffmpeg configuration directory:

`audio/aud.mov` and `video/vid.mov`

Now you will see this as what you can choose in the rendering choices for ffmpeg. Inside the file you will see that the first line is special. It is the muxer and codec. For example:

```
h264 libx265
```

The contents may be something like:

```
# <path>/video/vid.mov
mp4 libx265
bitrate 4000000
```

This will code an mp4 formatted file using the *lib264* codec encoder.

For audio and video together, the mux format must agree between the aud.mov and vid.mov files when they are to be used together. The stream muxer must be the same for all the streams in the file being written. For example:

```
# <path>/audio/aud.mov
mp4 pcm_mulaw
```

This will create mp4 media using audio format *pcm_mulaw* coding.

Both the audio and the video are using mp4 mux format, and so there will be 2 streams:

1. x265 video
2. pcm_mulaw audio

When the menu popup is created, there may be many choices for that class type, so you may want defaults. That can be specified as:

`audio/<class>.dfl` and `video/<class>.dfl`

```
# audio/mov.dft
aud.mov
```

```
# video/mov.dft =
vid.mov
```

The above will be the default choice when the menu opens.

When you see problems in using the new options files you have created and put into place, add the following line to `ffmpeg/encoder.opts`:

```
loglevel=verbose
```

sometimes that will be enough to see what has caused a failure, or even catch unexpected results.

There is an EXCEPTION to all of the above because of a conflict between ffmpeg and the x264 person making the detection of default ffmpeg settings terminate with an error. If you get this error, you must workaround this termination by including parameters that don't match 5 or more of the normal expected values. So you just have to change a few parameters to avoid the probe detection. Here is an example where you will notice the *x264-params* line tweaking values to throw off the detection/error termination code.

```
# <path>/ffmpegvideo/test.mp4
mp4 libx264
preset=slow
x264-params keyint=25:min-keyint=4:qpmin=3:qpmax=33:qp_step=4:merange=8
crf 20
```

For more examples, look around the ffmpeg directory for examples which may be close to what you are trying to use, and see if the parameters look usable.

This is quite complicated, but that is because ffmpeg has a lot of parameters and history. Good results are not that hard to create. Initially you should mostly use the defaults. If you send any new options files to cin@lists.cinelerra-gg.org, it will be given consideration to being added to the baseline for future deliverables.

To get a listing of the current ffmpeg supported formats and codecs that can be made to work with CINELERRA-GG, provided there are option files added, run the following commands. This should be done from the `<build>` directory substituting the location of `<build>` where you have installed CINELERRA-GG on your system and the ffmpeg may be a different version than 4.2 as used below. Then look at the output created in `/tmp/ff-formats.txt` and `codecs.txt`.

```
/<build>/cinelerra-5.1/thirdparty/ffmpeg-4.2/ffmpeg -formats > /tmp/ff-formats.txt
/<build>/cinelerra-5.1/thirdparty/ffmpeg-4.2/ffmpeg -codecs > /tmp/ff-codecs.txt
```

9.2.3 Complete Options File Example

For illustrative purposes, here is an example of the options files that need to be added for using the ffmpeg *ProRes 422* format. This makes it possible to transcode to `h264.mov` with FFmpeg retaining *10-bit yuv422p* from the source to the target output video.

Add the file named `./ffmpeg/audio/aac256k.pro` which contains the following lines:

```
mov aac
strict -2
b 256000
```

(Note that in the example above, even though the bitrate is set here to 256000, it can be overridden by the render menu settings).

Add the file named `./ffmpeg/audio/pro.dfl` which contains the following lines:

```
aac256k.pro
```

Add the file named `./ffmpeg/video/prores.pro` which contains the following lines:

```
mov prores
profile=2
# lines of comments
```

Add the file named `./ffmpeg/video/pro.dfl` which contains the following lines:

```
prores.pro
```

Then to use and to get 10 bit depth and preserve depth from decode to encode:

1. load media
2. use `settings` → `format` to set the frame rate, sample rate/channels, aspect ratio, color model = rgb_float or rgba_float if blending
3. press Shift-R and select FFmpeg format type *pro*
4. select target path
5. check OK, and watch for messages in the terminal window

9.2.4 Modifying FFmpeg Format Options inside CINELERRA-GG

There are thousands of options for using ffmpeg. Now it is possible to *view* the available options for a particular video and audio choice by using the *wrench icon* and then clicking on the *view* box. FFmpeg has to be the selected file format for this feature to be visible. It makes it a lot easier since only the applicable options show up as opposed to everything that ffmpeg can do. These options are just *Hints* and some may be missing due to the way that ffmpeg options are coded – CINELERRA-GG shows the option data ffmpeg has exposed.

As an example, instead of reading the entire 264 library information, you only have to look at the shown available options. Both the video and the audio are browsable. The options visible in the *Audio/Video Preset* textbox are the final values which are used when rendering once you have checked OK. For assistance in choosing the options you want, use the view popup to see the objects that go with the selected format tool, highlight the option, modify the parameter value in the box at the top of the *Options* window based on what you want, and then click apply. Updated parameter values or new parameters will be appended at the bottom. Note that when you highlight an option, a tooltip will show up when available in the lower right hand corner which describes the option. Also note that the Format and Codec types are shown on the top line of the Options window.

Parameters exist in 3 layers: ffmpeg, codec, and an interface layer. You can apply parameters to each layer. The top 2 layers are accessed with the Kind popup menu. The ffmpeg layer is topmost, and is selected as Kind: ffmpeg. It can specify many of the more common parameters, such as the bitrate, quality, and so on. The middle layer is selected as Kind: codec. These options can specialize your choices, and frequently includes presets and profiles useful for coding well known parameter sets, like *profile=high422*, *preset=medium*, or *tune=film*, etc. The interface layer may or may not be available. It is usually accessible only by an *opts* parameter, like `x264-params key=value:key=value:...` These options are passed directly to the low level codec library.

With the *format* button we can access further parameters, called *Private Options*. A window opens that contains a new view button. More details in the section: Extra “cin_” Options for Render with FFmpeg

Figure 9.2 shows *ffmpeg* video as the Kind. Note the x264opts (should actually be x264-params now) in the Video Preset window immediately below. There is also the *format* window with private options.

Figure 9.3 shows *ffmpeg* video for the Kind. Note the colored tooltip in the lower right hand corner describing the highlighted option. Also note the allowed *Range* values above the box provided for keyins. There is also the *format* window with private options.

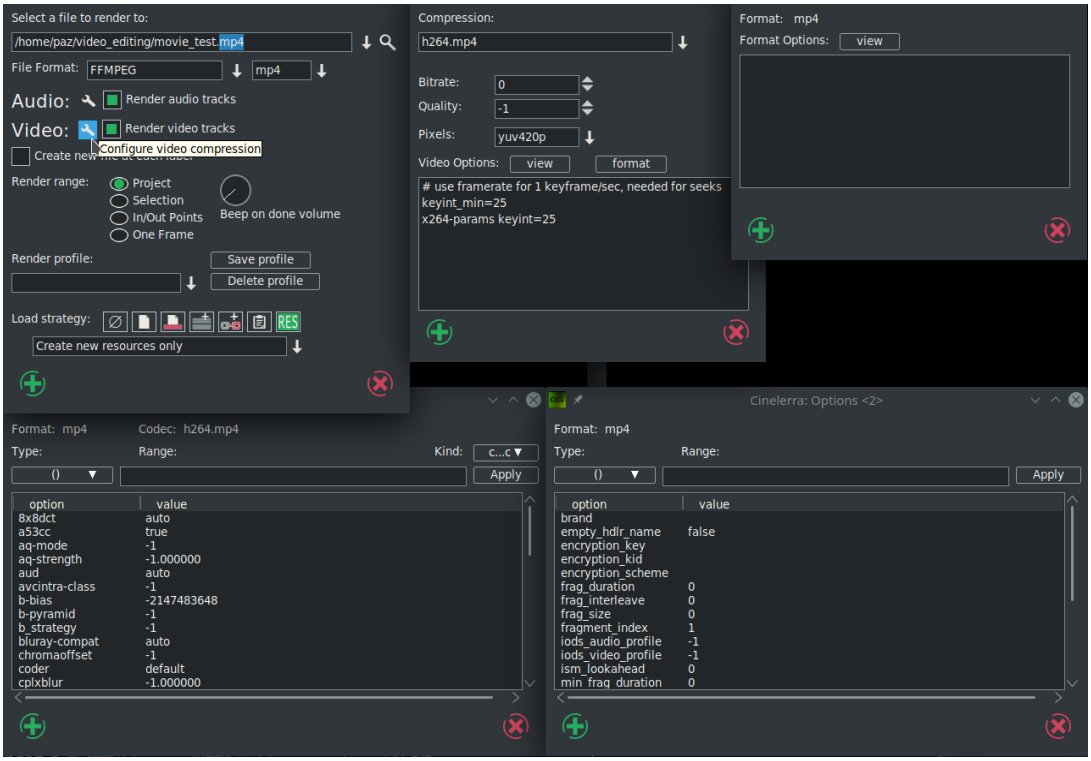


Figure 9.2: *FFmpeg wrench, video preset, view and format options*

9.3 The FFmpeg Image2 Streams

Another feature gained from using ffmpeg in CINELERRA-GG takes advantage of what is being referred to as the *%d trick*. This trick uses the ffmpeg muxer image2 and a filename template to create a series of image files of a given type. A specific example is described below. AppImage does not provide this capability unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds.

To encode a series of 48 bit tiff output image files, add a file to the CINELERRA-GG data ffmpeg/video subdirectory as in:

```
# {path_to_cinelerra}/bin/ffmpeg/video/tiff.dfl
tiff48.tif
```

Then create an ffmpeg video encoder parameters file in the same directory:

```
# {path_to_cinelerra}/bin/ffmpeg/video/tiff48.tiff
image2 tiff
pixel_format=rgb48
```

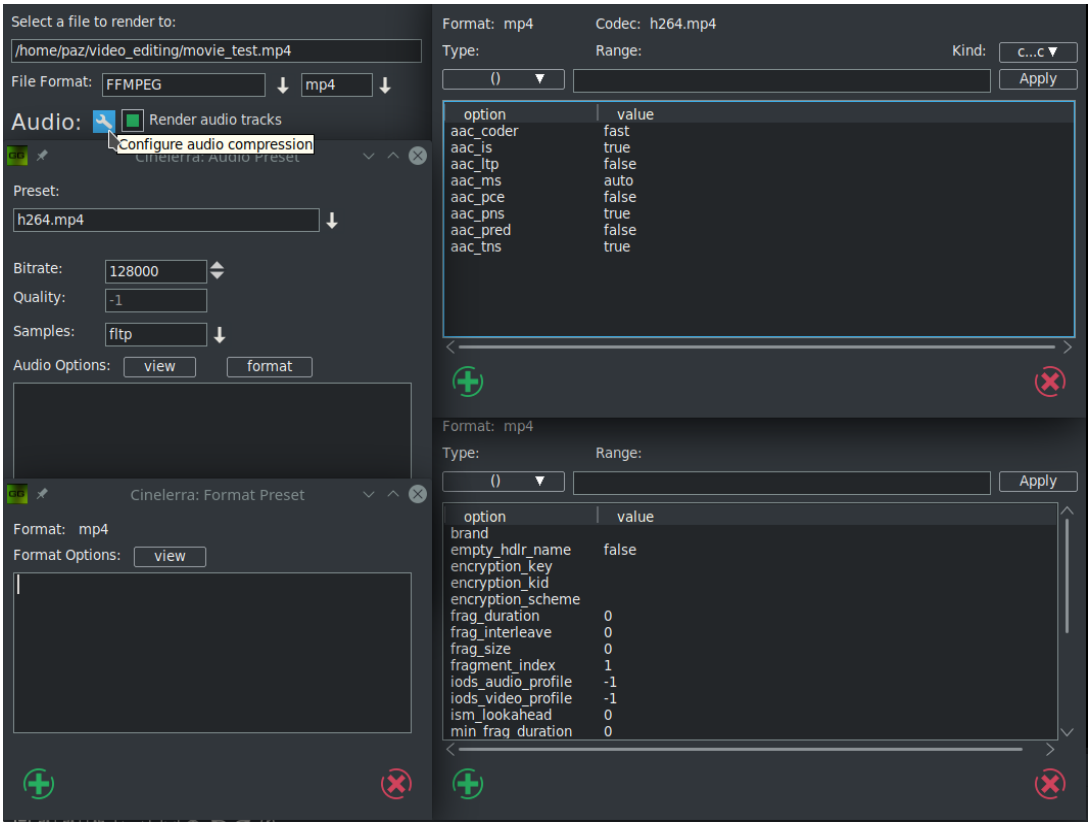


Figure 9.3: *FFmpeg wrench, audio preset, view and format options*

This will define a new ffmpeg encoder format which is a video image file format that uses the tiff codec for encoding, and a pixel_format of *rgb48* (or a similar equivalent such as rgb48le). Next load up your project and set up for a Render using `File → Render` in the usual way. Now the tricky part; the output file name should contain a `%d` which will be the frame number used in the image output filename as in: Select a file to render to `/tmp/tiff_images/img%03d.tiff`. You will get multiple files as output – one for each frame!

The resulting directory of images can be opened for reading by simply opening the template path. As in: `File → Load Files /tmp/tiff_images/img%03d.tiff`. You will notice a file named the same as the template, which has been automatically created, is empty, is needed, and has to remain with the set.

9.4 Raw Input Opts File for Video/Audio

Raw video is not affected by decoding on read in. This makes it very attractive to provide raw image data for editing and rendering media. A wide variety of raw formats are available via the ffmpeg file interface. To load media in a raw format, select *try ffmpeg first* and create an accompanying *opts* file. The opts files must be in the same directory as your

media, with the same base name, and the `.opts` extension. The opts file contents should reflect your video setup. An example follows:

```
# Video file name: /tmp/buddy.mov
# Opts file name: /tmp/buddy.opts
# Contents of opts file:

format=rawvideo
codec=rawvideo
video_size=352x240
pixel_format=yuv420p
duration=90.25
```

9.5 FFmpeg Items of Note

Quality Option when rendering: FFmpeg responds variably to the quality option in the render option but seems to respond well to bitrate. The subranges used by quality even seem to vary somewhat depending on how old the codec is. Some use 0 to 35, some use 0 to 500 or so. The quality is supposed to cause the codec to output data until the noise level is below a limit determined by the quality setting. Your specific results may vary.

Previous Changes when rendering: With ffmpeg there are 2 cases that the defaults will be used. The first time when you have nothing set up and any other time when you reset the render File Format in the Render Menu. Otherwise with ffmpeg, if you change a video compression type for the render (for example `h264.mp4` to `h265.mp4`), the settings will be from the previous session settings.

Outstanding Issues with ffmpeg: There are some problems that need to be addressed by the ffmpeg developer group that adversely affect CINELERRA-GG. These are stated below with the hopes that that group will fix them as time permits.

- Make all the default parameters operational. When they are not, the CINELERRA-GG plugins can't be initialized since the initial state of the filter is not operational. If that is not possible, then provide a set of nominal parameters for each plugin, so that they can be used as the plugins initial default state.
- Make the filter config function project the new parameter data into the filter function at any point during filter operation. This is so that continuous updates can be done as the plugin operates.
- Improve seek codec restarts. The past predictor must be reset or reconstructed after a seek. The only documented way to seek is open/seek/play. Reopening the format layer is very expensive.

Chapter 10

Plugins

There are realtime effects – these are the most useful and probably all you will ever need – and rendered effects. The rendered effects are discussed separately in the Rendered Effects section. Effects that begin with the characters F_ are effects that are part of the FFmpeg software as opposed to those coded within the CINELERRA-GG program. These are discussed separately in FFmpeg Audio and Video Plugins. Effect plugins modify the track when played, according to how they are set, with no permanent storage of the output except when the project is rendered. There are many Plugins in CINELERRA-GG Infinity which are actually quite easy to use just by experimenting with them. The plugins are shown and selected from the *Resources window* (figure 10.1). They are described in more detail later.

Figure 10.1: *Screencast of the native Video plugins in the default Cinfinity icon set.*

There is a choice of plugin icons which can be displayed.

In `Settings→ Preferences→ Appearance` tab, there is a pulldown for *Plugin icons* where the user can choose between the *original* icons, *regular* or *smoother*, *cinfinity*[1] – the default modernized set, or *cinfinity2* (figure 10.2).

[1]Cinfinity /2 icon set is credited to Sam - Creative Common By – https://creativhecommons.org/licenses/by/3.0/

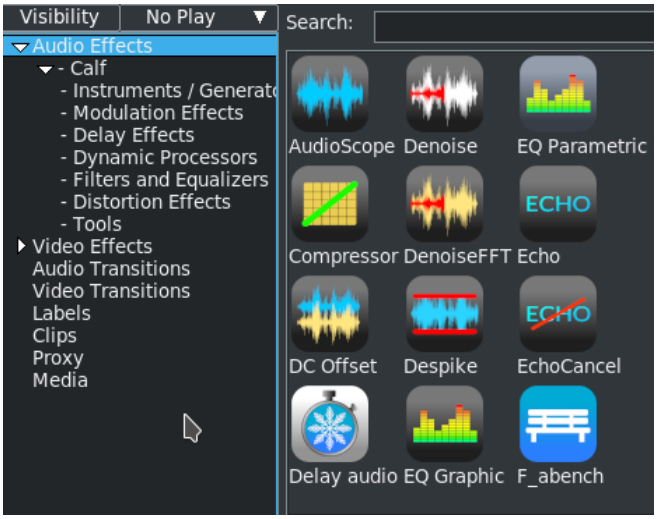


Figure 10.2: *Cinfinity2 audio plugins*

Note that when you change the plugin icons, your session will automatically save the backup, stop, restart, and reload (figure 10.3).

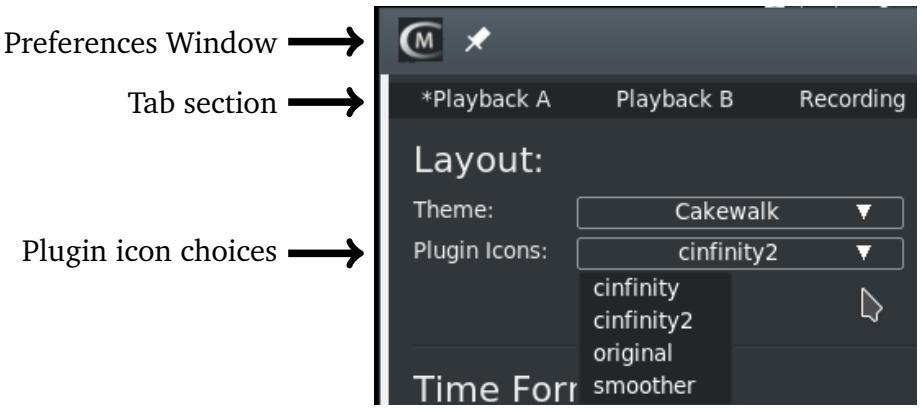


Figure 10.3: *Screencast showing the screen to change your plugin icons set*

10.1 How to Use Plugins

Realtime effect plugins are listed in the Resources window as *Audio* Effects and *Video* Effects. Effect plugins are used by dragging them from the Resources window onto an audio track if it is an audio effect or video track if it is a video effect. You will see a colored bar appear beneath the track with the plugin name on it. If there is data on the destination track, the effect is applied to the entire track, unless a edit or a region of the track is selected (wipe selection or In/Out points) in which case the effect is pasted into that region only. If there is no data on the track or no selected region set, the effect is not added.

Plugins are layered under the track they apply to in what is referred to as the effect bar. Do a double click on the effect bar to select the effect region. When dragging more than one effect onto an armed track, you will see the effects layering from *top to bottom*, on the bottom of that track. When the track is played back, effects are processed from *top to bottom*. The output of the top effect becomes the input of the bottom effect and so on. "The pipeline in Cinelerra's[2] plugin design is the *pull* method. The rendering pipeline starts at the final output and the final steps in the rendering pipeline are reading the data from disk. Every step in the rendering chain involves requesting data from the previous step. When the rendering pipeline eventually requests data from a plugin chain, each plugin requests data from the plugin before it. This is less intuitive than the push method but is much more powerful. Realtime plugins written using the pull method can change the rate data is presented to the viewer and the direction of playback. The pull method allows plugins to take in data at a higher rate than they send it out." Side note: if you want to avoid the automatic *top to bottom* application of plugins, you can take advantage of *nested assets* to manipulate the order the plugins are applied (see 6.6).

Instead of dragging from the Resources window, effects may be applied to a single armed track or a region via a popup menu. On the entire track or on a region determined by a selection wipe or by the In/Out points, right click and select *Attach effect* from the popup. The attach effect dialog gives you more capability than just dragging and dropping. For example, the attach effect dialog lets you attach two more types of effects: *shared effects* and *shared tracks* which are explained in a later section. Select a plugin from the Plugins column and hit the green colored checkmark under the plugins column to attach it. The result is the same as if the effect was dragged from the Resources window.

One more method to apply an effect on all armed tracks of the same type (video or audio) is to use the *Video* or *Audio* pulldown *Attach Effect* option. This brings up a menu which has a useful checkbox to *Attach single standalone and share others* (see 10.4). The default setting is checked on.

After attaching an effect to a track, it often needs to be configured. There are two ways to get to the configuration controls. Click on the magnifying glass symbol on the right side of the effect bar – this is the middle symbol on the bar as you can see in the picture below. Alternatively, you can right click on the effect bar to bring up the effect popup which has a *Show* option. Either method causes the GUI for the effect to appear in a separate window. There will not be a popup if the plugin has no GUI.

Besides the magnifying glass, for Show Controls, on the effect colored bar beneath the track, there are two more symbols.

The rightmost knob is used to Turn Off/Turn On the effect where the default is On. This is useful to easily see that the plugin is doing what you expect. The leftmost symbol that looks like a gear is for *Preset Edit* and its usage is described in the section Saved Plugin Presets.

[2] credit to Original Creator

10.2 Editing Effects

Many operations exist for manipulating effects once they are on the timeline. Because mixing effects and media is quite complex, the methods used in editing effects are not as concise as cutting and pasting. Some of the editing happens by dragging in/out points, some of the editing happens through popup menus, and some of it happens by dragging effects.

When enabled, which is the default, and you edit tracks, the effects follow the editing decisions. If you cut from a track, the effect shrinks. If you drag edit in/out points, the effect changes length. This behavior can be disabled by selecting `Settings→ Preferences→ Interface tab→ Editing section`, see figure 10.4 .

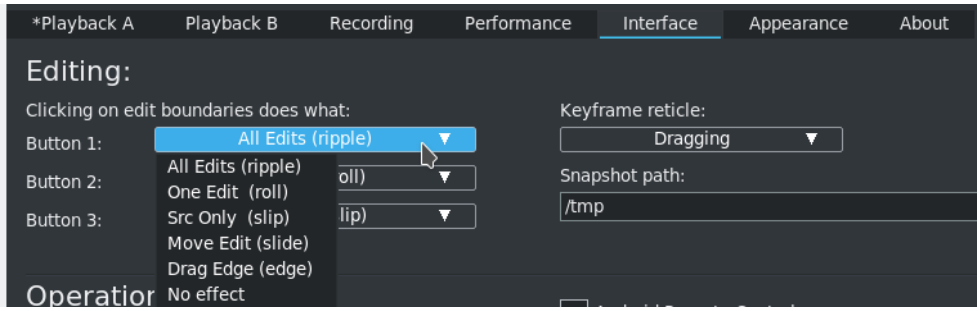


Figure 10.4: *Screencast showing the menu where Editing Effects behavior is changed.*

To edit effects, you can move the timeline cursor over the effect borders until it changes to a resize left or resize right icon. In this state, if you drag the end of the effect, it performs an edit just like dragging an edit edge. The five editing behaviors of track trimming apply to effect trimming and they are bound to the mouse buttons that you set in interface preferences as shown in the previous screencast. *Trimming* simply means changes the duration dragging the edges.

When you perform a trim edit on an effect, the effect boundary is moved by dragging it. Unlike track editing, the effect has no source length. You can extend the end of an effect as much as you want. Also unlike track editing, the starting position of the drag operation does not bind the edit decision to media. The media the effect is bound to does not follow effect edits. Other effects, however, do follow editing decisions made on an effect. You can disable effects from being subject to the edit decisions by using the pulldown *Settings* and toggling off Edit effects. If you drag the end of an effect which is lined up to effects on other tracks, the effects on the other tracks will be edited while the media stays the same. When you drag an effect in from the Resources window you can insert the effect in the portion of the row unoccupied by the trimming operation. In some cases you will want a trimming operation to change only one row of effects. This can be achieved by first positioning the insertion point on the start or end of the effect. Then press the shift key while beginning the trimming operation. This causes the operation to change only one row of effects.

You can move effects up or down. Every track can have a stack of effects under it. By moving an effect up or down you change the order in which effects are processed in the stack. Go to an effect and right click to bring up the effect menu. The *Move up* and *Move down* options move the effect up or down. When you are moving effects up or down, be aware that if they are shared as shared effects, any references will be pointing to a different effect after the move operation.

Finally, there is dragging of effects . Dragging effects works just like dragging edits. You must select the arrow in the main window transport buttons line to enter drag and drop mode before dragging effects. Dragging a plugin causes a highlight outline to be drawn over a targetable timeline region, and the plugin can be re-positioned into any plugin track. The effects snap to media boundaries, effect boundaries, and tracks. If you drag a reference to a shared effect, the reference may point to the wrong effect afterwards. It is recommended that you re-construct shared effect track references.

Figure 10.5 showing 3 plugins, two still plugin, two have already been dragged and the *Color 3 Way* in the process of being dragged. Note the gold-colored frame which enables allow *drag and drop* editing mode.

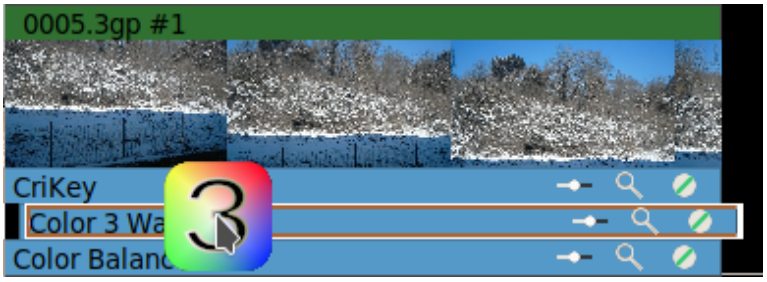


Figure 10.5: *Dragging the Color 3 way effect*

10.3 Copy/Paste Effects

In *Drag and Drop* mode we can copy and paste edits containing plugins (see also Copy/-Paste Behavior). Start by selecting an edit (that contains plugins) with LMB and open the context menu with MMB. The possibilities offered are:

Ctrl - c : copy edits and plugins to buffer.

Ctrl - v : paste edits and plugins into the timeline (splice) at the Insertion point.

Ctrl - b : paste edits and plugins into the timeline (overwrite) at the Insertion point.

Ctrl - Shift - p : paste only the plugins into the timeline, at the Insertion point, keeping the original length (overwrite).

Collect Effects : copy only plugins into buffer.

Paste Effects : pastes the plugins into a selected target edit. The length of the plugins adapts to the length of the edit.

10.4 Shared Effects and Shared Tracks

Two other effect types available in the Attach Effect dialog are *Shared effects* and *Shared tracks*. In the case of a shared effect, the following conditions must be true:

- There must be other effects in the timeline.
- The other effects must be of the same type as the track you are attaching an effect to. That is for audio tracks, effect must be audio and for video tracks, effect must be a video effect.
- The insertion point or selected region must start inside the other effects.

Before going into further detail about how to use *Shared effects*, an easier alternative method of application which is especially useful for Audio tracks is available. In this method, all you have to do is use the *Audio* pulldown and choose *Attach effect* , highlight the effect you would like and make sure the default of *Attach single standalone and share others* is checked on. It will automatically be a "Shared Effect" on all audio tracks (be sure to disarm any audio tracks that you do not want to have the effect shared on). This method also works for Video tracks using the *Video* pulldown choice of *Attach effect*.

In the case of a shared track, there must be another track on the timeline of the same type as the track you are applying an effect to. If you right clicked on a video track to attach an effect, there will not be anything in the shared tracks column if no other video track exists. The same applies equally to audio tracks in that another audio track must exist. Shared tracks are often used as layers for titles, curves and keyframes.

If shared effects or shared tracks are available, they appear in the shared effects and shared tracks columns when you used the *Attach effect* option (RMB on a track). When the green colored checkmark is clicked OK, anything highlighted in the column is attached under the current track.

Shared effects and shared tracks allow very unique things to be done. In the case of a shared effect, the shared effect is treated like a copy of the original effect, except that in the shared effect the GUI can not be brought up. All configuration of the shared effect is determined by the GUI of the original effect and only the GUI of the original effect can be brought up.

When a shared effect is played back, it is processed just like a normal effect except the configuration is copied from the original effect. Some effects detect when they are being shared. These effects determine what tracks are sharing them and either mix the two tracks together or use one track to stage some value.

When an original track has a shared track as one of its effects, the shared track itself is used as a *realtime* effect. This is more commonly known as *bouncing tracks* but CINELERRA-GG achieves the same operation by attaching shared tracks. The fade and any effects in the shared track are applied to the original track. Once the shared track has processed the

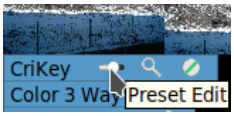
data, the original track performs any effects which come below the shared track and then composites it on the output.

In order to prevent the shared track from mixing the same data as the original track on the output, enable the output `mute` toggle in the patchbay next to each track for which you do not want to mix on the output. If you are making a video and you do want the shared track to composite the original track's data on the output a second time, the video from the shared track would always appear under the video from the original track, regardless of whether it was on top of the original track. This is because shared tracks are composited in order of their attachment. Since it is part of the original track it has to be composited before the original track is.

10.5 Saved Plugin Presets

Presets and *Factory Presets* for Plugin settings are now combined with the Preset Keyframe Parameters allowing you to choose, apply, delete, and edit your own Presets which can then be easily saved in the file `$HOME/.bcast5/Cinelerra_presets`. In addition to your own saved presets, there are automatically available Factory presets for some plugins, for example the Lens video plugin. The Factory presets are preceded by an asterisk (*) and can not be modified permanently.

Note that using this is directly changing a keyframe object so you will only want to modify parameters you are familiar with. Most of the data is obvious and safe to change.



A Presets button on the plugin bar to the left of the Controls and On/Off button allows for quick access to this feature. The symbol resembles a slider (figure 10.6).

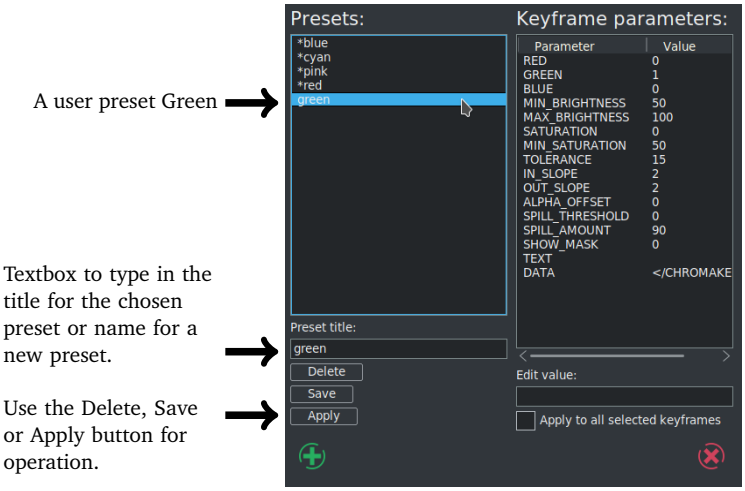


Figure 10.6: *Screencast shows 4 Factory presets as preceded by an *.*

10.6 Some specific details concerning Plugins

These next few sections explain some details about the plugins that are not directly related to actually using them but help to work with them.

10.6.1 How to see short Description of a Plugin

To get a short one or a few lines description of a plugin, mouse over that plugin in the Resources window and a popup appears. You can disable the popup by right-clicking and choosing *Info Off* (shortcut "i"). And again enable it with *Info On*. Some of the plugins may not have any description included. An example screenshot is next (figure 10.7).

Figure 10.7: *Effect Info for Color 3 Way*

Context Help provides more detailed information about the plugin in 4 possible ways using the Alt/h hotkey combination.

1. Pressing `Alt/h` with the mouse in the dialog window of that plugin's settings menu.
2. Pressing `Alt/h` while pointing with the mouse on the plugin bar in a track, transition bar, or transition icon.
3. Pressing `Alt/h` while pointing on the plugin name (icon) in the Resources window. If plugin tooltips are on, help for the highlighted plugin under the mouse is shown. If plugin tooltips are off, help for the selected plugin is shown.
4. Pressing `Alt/h` while pointing on the plugin name in the *Attach Effect* dialog menu.

10.6.2 Delete Plugins to save Resources Space or make them Unavailable

Maybe you just don't ever use certain plugins or would prefer to only find the ones that are useful to you. To save space in the Resources Window so you don't have to scroll to find the plugins you want as much, a feature to delete others is available. If you have a System install, you will have to be root for this function to be usable. The plugins will be permanently deleted, but only until you rebuild or download a new set of CINELERRA-GG binaries. To delete a plugin, highlight the plugin you no longer want in the Resources window then press Ctrl-Shift-delete. A small window will come up allowing you to change your mind and red-X out or check-OK to remove plugin. AppImage does not provide this capability unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds.

This feature may come in handy if you have personnel working on media for you and you only want them to exercise certain functions. Or maybe you can't remember which is the good *deinterlace* plugin out of the available five or so and want to delete the extras so as not to be confused. The ffmpeg, *ladspa*, and *lv2* plugins can not be deleted in this manner but, of course, you can always turn them off from view by clicking on *Visibility* and unchecking them (figure 10.8).

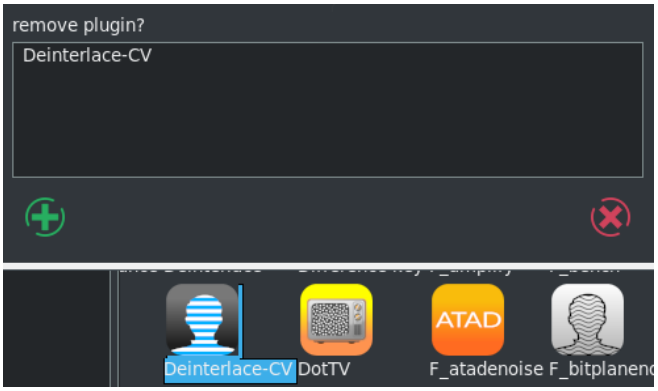


Figure 10.8: *Remove Deinterlace-CV plugin*

10.6.3 Updatable Icon Image Support

When running CINELERRA-GG Infinity builtin icons are loaded before the program starts. Png files in the path:
`<target_directory>picon/picon_set_name`
are searched before the images loaded into memory. Override `icon.png` files must be put into the path:
`<target_directory>/picon/picon_set_name`
There are currently 4 sets of icons and their directory names are *cinfinity* (the default) and *cinfinity2*, *original* (the long-time original set), and *smoother* (generally was in use by some of the themes). An example, to replace the cinfinity icon of Blue Banana with a red

apple instead, create your .png file as desired, and replace the file in:
<target_directory>/bin/plugins/picon/cinfinity/bluebanana.png.

For most User installs, the <plugin_name>.png file will be located at:

<cinlib_path>/bin/plugins/picon/cinfinity (or cinfinity2, original or smoother)

For some System installs, the files might be located at:

/usr/lib/cin/plugins/picon/cinfinity

(or cinfinity2, original or smoother – ubuntu distros)

/usr/lib64/cin/plugins/picon/cinfinity

(or cinfinity2, original or smoother – Leap distro)

AppImage does not provide this capability unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds.

10.6.4 Details on where to put your own Plugin Icons

In order to make the icons available to all themes, which would thus be the default when no theme-specific icon is available, put the png file in the:

<cinlib_path>/bin/plugins/picon/cinfinity (or cinfinity2, original or smoother)

The CINELERRA-GG program looks for a plugin icon in two places:

1. First, it tries to find a png file in <cinlib>/plugins/picon/cinfinity(2) or original, smoother directory.
2. If there is no corresponding .png file for a plugin, the program uses a built-in default:
   - ordinary video plugins use 3 vertical color bars as a default;
   - ffmpeg plugins use the words *FF* on yellow colored background as a default icon;
   - audio and ladspa plugins use a green-colored audio wave for a default.

Keep in mind these points for newly created plugin icons:

- All included icon images become part of open source, in the public domain, and not proprietary.
- The preferred format is 52 × 52, 8 *bit* /color RGB or RGBA, non-interlaced.
- Since plugin icons are used by different themes, it is recommended that a *transparent background* be used. Otherwise some color background that looks good for one theme may not for another.
- In order to test a new icon, you have to have write permission in the: `<cinlib_path>/plugins` directory so you may have to become the root user to copy the `.png` file to the correct location.
- If there is currently no theme-specific `.png` files present, it may be necessary to first create the theme directory in `<cinlib_path>plugins` as `<theme_name>` in order to put the `.png` files in that subdirectory.
- Make sure that the *ownership* and file *permissions* match the existing directory and files.
- All ffmpeg icons must begin with `ff_<plugin_name>.png` (Resources window title will still be F_...)
- For ladspa, check in the `<cin_config>` directory (`$HOME/.bcast5` normally) and look for the text file `$HOME/.bcast5/ladspa_plugins...` for the names of the ladspa libraries which correspond to plugin names where the needed name is the basename of the `.so` file. For example `phasers_1217.so` would need to have a `phasers_1217.png` file. There may be multiple plugins in a single "so" file which means that you can only have 1 icon to represent all of the plugins in that file; again as in phasers.
- Once you have placed the .png file in the correct spot, you will have to restart CINELERRA-GG to test it.
- To submit your `.png` file for inclusion into CINELERRA-GG Infinity for all to enjoy, it is best to upload it to any datafilehost and notify the community via email with any informative documentation.

10.6.5 Example of new Plugin Icon Testing

For a simple test just copy an existing `<plugin_name>.png` file into the cinfinity directory with the name `bluebanana.png` to write over the existing file. This icon will now show up in CINELERRA-GG and still execute the Blue Banana function. AppImage does not provide this capability unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds.

For an ffmpeg plugin, create `ff_loop.png` and copy it to: `<cinlib_path>/plugins/picon/original`. This icon will show up in CINELERRA-GG if original is selected and execute the *F_loop* function.

For a ladspa plugin, the text line in `$HOME/.bcast5/ladspa_plugins...` as seen below:
`2 am_pitchshift_1433.so`
`AM pitchshifter 15049223210100101010 0` indicates that you would create the icon:
`<cinlib_path>/plugins/picon/cinfinity/am_pitchshift_1433.png`
For your own personal plugins, you can create a directory on your system and put any plugin png files you like into that directory. For example, if you want a specialized picon for *F_aeval*, create a picon named `ff_aeval.png` in:
`<cinlib_path>/plugins/picon/yournamehere.`

```
cd <cinlib>/plugins # go to the correct directory
mkdir -p picon/yournamehere # create subdirectory if does not exist
ls -l picon/* # list the picon directories
                                    # check for existence (and permissions)
cp yourpicon.png ff_aeval.png # Copy your example .png file
```

Restart cin by changing `Settings→ Preferences→ Appearance` and in *Plugins icons* choose a directory.

10.6.6 Plugins/Effects Visibility

CINELERRA-GG contains many plugins, especially with the addition of ffmpeg, and it is somewhat difficult to find the one you are looking for in the Resources window. In CINELERRA-GG Infinity, the plugins have been categorized into the following subsets in the *Visibility* section of the Resources window to make it easier to locate a particular one:

Audio Effects, Video Effects, Audio Transitions, Video Transitions

The *Visibility* tool in the Resources window (figure 10.9) gives you the ability to turn off or on any of several sets of plugins. If you left-click the Visibility box, you will see the various categories of plugins, such as *ladspa*, *ffmpeg*, *audio*, *lv2*, and *video* (figure 10.10).

Highlight the set you want to turn on and a check mark appears to show it is active. Highlight again to toggle it off. See the next screenshot which illustrates that all of the plugins are turned off (not visible) except for audio. There is also the ability to add your own personal directory of plugins which will show up here. All you have to do to have these plugins become visible is to create a directory, with some name that is meaningful to you, and put your .png files in your:
`cinelerra_path bin/plugins/<your_directory_name>`.

10.6.7 Expanders for Plugin Subtrees in the Resources Window

To accentuate a set of common plugins, there are *expander* arrows on the left side of the Resources window. You will see these expanders only when in *Display text* mode, not *icon*

Figure 10.9: *Screenshot showing on the left hand side the Visibility box with Audio Effects highlighted.*

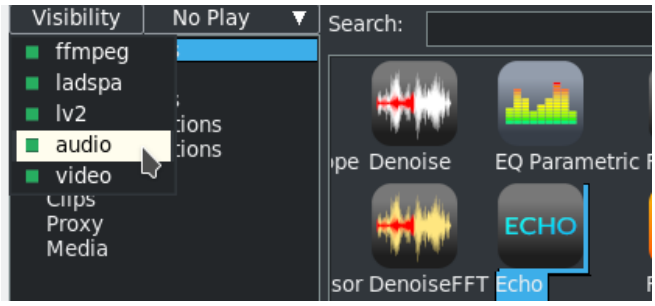


Figure 10.10: *Screenshot showing the Visibility categories of plugins with all toggled on and audio highlighted.*

mode. CINELERRA-GG's default setup is in the file `$CIN_DAT/expanders.txt` but if the user wants their own specific setup and if the file in `$HOME/.bcast5/expanders.txt` exists, it will take precedence. If there are recommendations for other relevant categories, they can be added. The subtree structure is applicable to any of the *Video Effects/Transitions* or *Audio Effects/Transitions*. You can not sort once an expansion is in effect (figure 10.11).

Figure 10.11: *Down facing triangle, Right facing triangle = expander;* "-" *= options*

The `expanders.txt` file has very specific requirements. The most specific is that there are no blanks – you must use tabs only. A # (pound sign) can be used in column 1 to indicate a comment. A short example of what the txt file looks like is shown below.

```
Video Effects
    -Favorites
        Brightness/Contrast
        Title
        VideoScope
    -Color_Correction
        Blue Banana
        Color 3 Way
        Color Balance
Audio Effects
    -Calf
    -Instruments / Generators
        L2_Calf Organ
        L2_Calf Monosynth
        L2_Calf Fluidsynth
```

NOTE: The capability to put the expanders.txt file in `./bcast5` (see .bcast5) allows for customizations such as grouping favorite plugins in one section so that they can be recalled more quickly. For this purpose, the *Favorites* section has been included at the top of Video Effects, the contents can be changed as desired. Once your modified file is saved to hard disk as $HOME/.bcast5/expanders.txt it will remain in use on every update as long as you do not delete it or remove the .bcast subdirectory. For users running CINELERRA-GG using the AppImage, to make an initial copy of expanders.txt, execute the following lines on /tmp:

```
git clone --depth 1 git://git.cinelerra-gg.org/goodguy/cinelerra.git cinelerra5
```

```
cp /tmp/cinelerra5/cinelerra-5.1/expanders.txt $HOME/.bcast5/expanders.txt
```

10.6.8 Speed-up of Ffmpeg plugin usage with OPTS files

You can speed up some ffmpeg plugins that are quite time-consuming and use a lot of CPU. For a specific color-based example, CINELERRA-GG uses 6 primary rendering color models. All of them have 3 components at full scale. Direct usage of a particular ffmpeg plugin from the ffmpeg command line might handle the planar at less than full scale chroma (yuv420), which means there is less data to manipulate. But when cinelerra loads a video it uses full scale color models. In other words:

- CINELERRA-GG uses *yuv444*
- ffmpeg uses *yuv420*

if using an ffmpeg plugin that uses filters and many passes over the data, the amount of data is a big factor. If you load a file in CINELERRA-GG with a

`same_directory_path_and_filename.opts`

file containing the following line, the full scale color modeling upgrade will not be performed until after any plugin, and then the render is faster:

```
video_filter=xxxxxx=threads=8 # where xxxxxx is the desired filter
```

When the file loads, however, it will initially take longer because it is running through the video filter. The format *rgb* in ffmpeg uses more cpu time. For comparison, ffmpeg line that might be used:

```
ffmpeg -i /tmp/filename.mpeg -threads 15 -vf format=rgb24,xxxxxxs=threads=8 -
    ↪ acodec ac3 -vcodec libx265 -y /tmp/x.mp4
```

This converts the input to rgb before xxxxxx runs, and so it too is slower (because there is more color data). You would ordinarily avoid this conversion by omitting the `format=rgb24` parameter. An example ffmpeg plugin that could easily take advantage of an auxilliary opts file is *nlmeans*.

10.6.9 Floating-point pipeline and plugins

CINELERRA-GG runs internally in 32-bit floating-point and thus can read and process values outside the range $(0-1.0f)$ (*illegal* values, they need the final *conforming* step to comply

with the *legal* values). However, there are plugins and tools that can introduce clipping that must be taken into account to keep the whole pipeline which preserves the original values. The limited plugins are: *Blur; Color 3 Way* (partially); *Color Space; Foreground; Histogram* (partially); *Histogram Bezier; Title; Videoscope* (partially).

10.7 Audio Effects - Native

10.7.1 AudioScope

Effect rewritten and improved. Convert input audio to video output representing the audio power spectrum. Shows you the sound wave.

10.7.2 Chorus

It is a multitrack effect, where each track is a channel. For example if you have 4 voices per channel and 2 channels, you will have a total of 8 tracks. It is an effect that modulates the signal, varies the pitch up and down (instead of modulating the phases as for example in the *Flanger* plugin) and creates voices from the original signal and adds them to the Output. You then get a chorus effect, with multiple voices *singing* the same song but with slightly different modulations. Voices not only modulate the original signal but also start with a certain delay. There are two components of delay, *constant delay* and *oscillating delay* (figure 10.12).

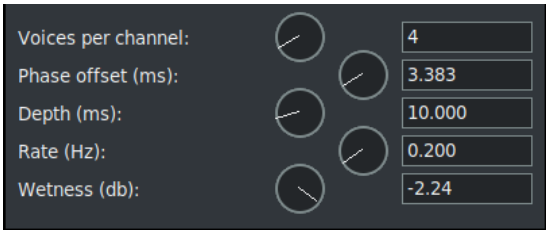


Figure 10.12: *GUI of configuration for Chorus plugin*

Voices per channel : number of items we want to put in the effect. Using more than 4 voices creates sound artifacts that lose the feel of a human voice choir, but can still be used as an artificial sound effect.

Phase offset (ms): is the constant delay, i.e. the amount of delay of the voices compared to the original signal.

Depth (ms): is the oscillating delay, i.e. the delay in the oscillation of the various voices from the original signal.

Rate (Hz): is the speed at which we apply the oscillating delay. In other words, the speed at which the oscillations occur.

Wetnwss (db): Indicates how much of the original (dry) signal is taken into account compared to delayed voices.

10.7.3 Compressor (Single Band)

The audio compressor reduces the dynamic range of the audio, not the amount of data required to store the audio. In CINELERRA-GG the compressor actually performs the function of an expander and compressor of the signal's dynamic range. A third and more sophisticated use serves to highlight the voice with respect to the sound background. It is a multitrack effect and can also be applied as a Shared Effect. (figure 10.13).

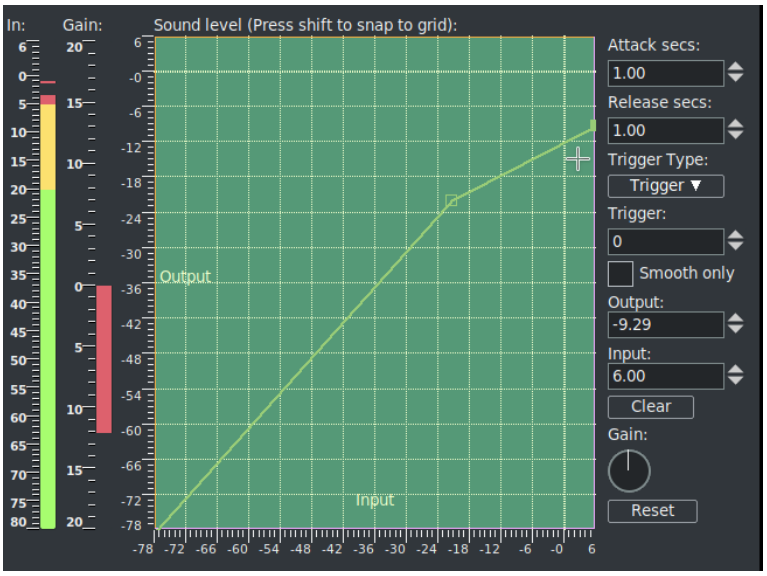


Figure 10.13: *GUI of configuration for Compressor plugin*

The compressor works by calculating the maximum sound level within a certain time period of the current position. The maximum sound level is taken as the input sound level. For every input sound level there is an output sound level specified by the user. The gain at the current position is adjusted so the maximum sound level in the time range is the user specified value. The compressor has a graph which correlates every input sound level to an output level. The horizontal direction is the input sound level in dB. The vertical direction is the output sound level in dB. The user specifies output sound levels by creating points on the graph. Click in the graph to create a point. If two points exist, drag one point across another point to delete it. Moving the point horizontally is equivalent to the *makeup gain* parameter. Note that it is impossible to create a vertical curve; points would be deleted. Put a part of the curve in horizontal means clamping the signal to the same, unique value (as Limiter). The most recent point selected has its values displayed in textboxes (Output and Input) for more precise adjustment. To have the compressor reduce the dynamic range of the audio, make all the output values greater than the input values except 0 dB. To make the compressor expand the dynamic range of the audio, make all the output values except 0 dB less than the input values. The algorithm currently limits all sound levels above 0 dB to 0 dB, so to get an overloaded effect put a gain effect before the compressor to reduce all the levels and follow it with another gain effect to amplify all the levels back over 0 dB. The volume (in Db) of the input signal is shown in the *In meter* on the left. Next to it is the *Gain meter* which indicates the gain added (green) or subtracted (red) to the original signal according to our settings.

Attack secs : determines where in relation to the current position the maximum sound level is taken and how fast the *gain* is adjusted to reach that peak. It is in seconds. If the reaction time is negative the compressor reads ahead of the current position to get the future peak. The gain is ramped to that peak over one reaction time. This allows it to hit the desired output level exactly when the input peak occurs at the current position. If the reaction time is positive the compressor scans only the current position for the gain and ramps gain over one reaction time to hit the desired output level. It hits the output level exactly one reaction time after detecting the input peak.

Release secs : if the peak is higher than the current level, the compressor ramps the gain up to the peak value. Then if a future peak is less than the current peak it ramps the gain down. The time taken to ramp the gain down can be greater than the time taken to ramp the gain up. This ramping down time is the decay seconds.

Trigger type : the compressor is a multi-channel effect. Several tracks can share one compressor. How the signal from many tracks is interpreted is determined by the *trigger type*. The Trigger type uses the value supplied in the Trigger textbox as the number of the track to use as input for the compressor. This allows a track which is not even heard to determine the loudness of the other tracks. The maximum trigger takes the loudest track and uses it as the input for the compressor. The Total trigger type adds the signals from all the tracks and uses the total as the input for the compressor. This is the most natural sounding compression and is ideal when multiple tracks are averaged into single speakers.

Trigger : This parameter is used in conjunction with trigger type as described previously. Normally only one track is scanned for the input peak. This track is specified by the Trigger. By sharing several tracks and playing with the trigger value, you can make a sine wave on one track follow the amplitude of a drum on another track, for example.

Smooth only : for visualizing what the compressor is doing to the sound-level, this option causes it to replace the sound wave with just the current peak value. It makes it very easy to see how *reaction secs* affects the detected peak values.

Gain : moves the curve in the vertical direction only, to change the gain of the Output.

Clear and **Reset**: Reset the parameters to the default values (no compression).

10.7.4 Compressor Multi (Multi Band)

Refer to Compressor (Single Band) for common theory and options. The normal compressor acts over the entire frequency spectrum. The multi-band allows us to distinguish three frequency ranges (low, med and high) on which to intervene separately and in a more sophisticated way. In other plugins there are four bands instead of three, but we can make very precise adjustments so the three present are enough because they are not fixed. Finally the value of the three corrections are added together in the Output.

The three bands and their range (adjustable as desired) are shown in the frequency graph visible below (Bandwidth:). As a first step you choose the band and adjust its range in the Bandwidth graph, then you go to adjust the compression in the top level graph. In this graph we can copy/paste the curve in the other bands via RMB; so we have a similar basis on which to make the adjustments.

In figure 10.14 we can see the three bands; the one currently active (*Current band: 2*) presents the waveform of the sound signal. The interval goes roughly between 300 Hz and 4000 Hz, the edges are vertical because the *stepness* is set to zero. So we have a clear separation between the bands, but by varying the slope we can have some overlapping of the bands for smoother effects. In the top level graph we can see the yellow curve of the active band, but we can also see in the violet the soft lines of the curves of band 1 and 2.

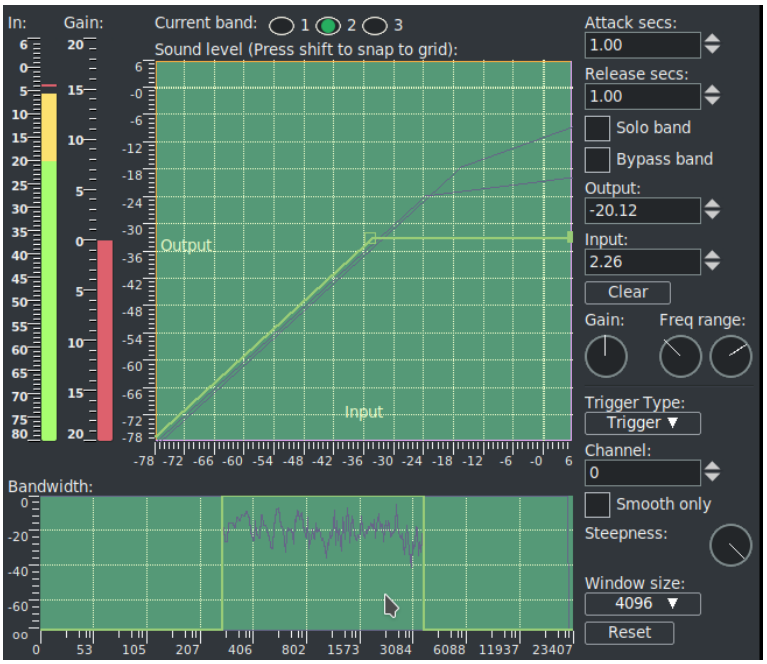


Figure 10.14: *GUI of configuration for Compressor Multi plugin*

Solo : brings only the active band to the Output, silencing the others. So we can make more precise adjustments without listening to the whole effect.

Bypass band : In contrast to Solo, it only brings the sound signals of the two inactive bands to the Output.

Freq range : is used to set bandwidth limits. You act separately on the left and right edge.

Steepness : varies the slope of the edges of the band range. Creates some overlap in the band boundaries.

Channel : is the equivalent of the *Trigger* option in Compressor (single band). Allows you to choose the channel or track to act on.

Window size : Determines the number of samples used in the compression calculation. The more they are, the higher the quality of the result, but the more CPU usage.

10.7.5 DC Offset

Use this to remove *DC Offset*, which is usually an undesirable characteristic of a recording normally caused by defective equipment. This effect works like a *high pass filter* and has no controls. DC stands for *Direct Current* which is the average amplitude of the waveform. It sounds best when it is absent, represented by zero, so that there is no imbalance in the audio.

10.7.6 Delay Audio

In the Delay Audio effect you can specify the number of seconds you want to delay the video track. An alternative ffmpeg plugin is F_adelay.

10.7.7 DeNoise

Reduce audio background noise. There is only 1 parameter which is used to regulate the level dial with a range of 0 to 1.

10.7.8 DenoiseFFT

Audio sampling (time data) is converted to a frequency spectrum using FFT (*Fast Fourier Transform*). Since the *reference* contains only the background noise while the rest of the clip contains noise + audio to clean up, you get separate frequencies so you can subtract them to get the clean audio only. Set the Denoise Power dial in dB and choose the number of reference samples.

For example, if you record a voice with a microphone[3]:

1. You record 15 seconds of background (without talking). This is represent the background noise that the filter uses as a reference.
2. After 15 seconds, start the voice. This is represents the recording whose filter should remove the background noise.
3. You stop recording.
4. You place the FFT effect on the whole length of the recording (including the 15 seconds of background noise).
5. Then place your cursor at the beginning of the 15 seconds. Enter the number of samples in the effect presets, e.g. 65536.

[3] credit fary54

This has the effect of creating a keyframe that marks the beginning of the reference range, mentioned in point 1, which in our example ends at 65536 samples later.

During playback, the background noise is then automatically removed along the entire length of the effect.

10.7.9 Despike

Detect and eliminate out of range impulse values.

Maximum level: slider to set the maximum value in dB above which the frequency cutting takes place.

Maximum rate of change: to adjust peak delete in dB.

10.7.10 EQ Graphic

Graphic equalizer sets the output levels for specified frequency bands. This effect works by setting control points when you click the left mouse button and drag to the desired value. In the textboxes at the bottom can be seen the frequency of the active control point, the level of the signal to be set by entering the numerical value or by dragging the control point, and the number of samples to act on (figure 10.15).

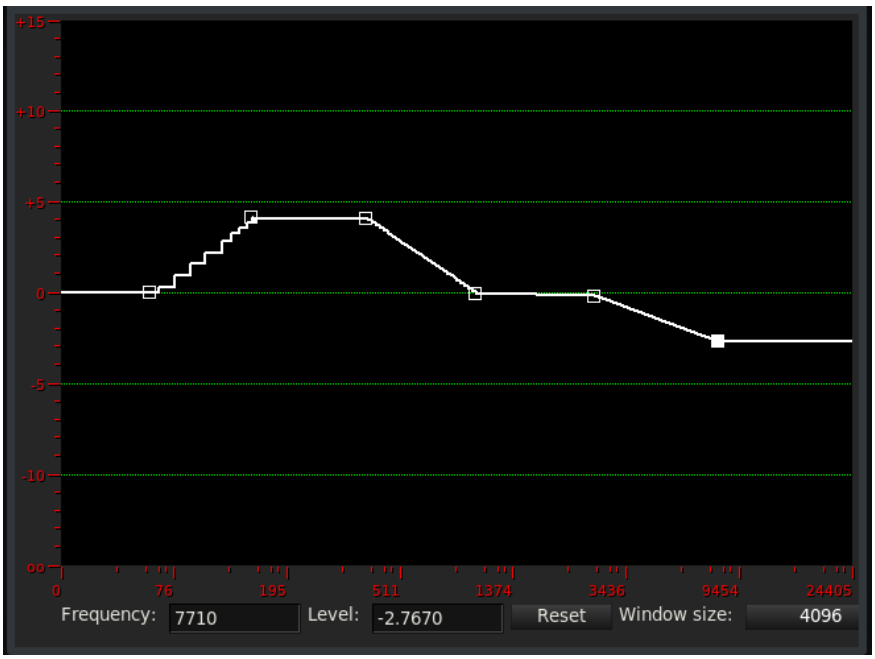


Figure 10.15: *Graphic Equalizer audio plugin*

10.7.11 EQ Parametric

Parametric equalizer shows and outputs levels for *frequency*, *quality*, *level*, *mode*, and *wetness* (figure 10.16).

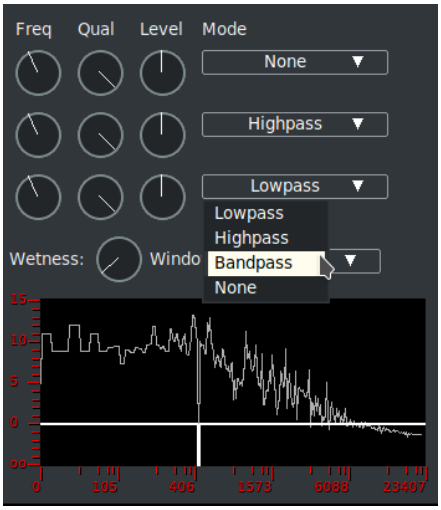


Figure 10.16: *EQ Parametric audio plugin*

10.7.12 Echo

Echo is reflection of sound. This plugin could be used to add echoing to video of your canyon hike (figure 10.17).

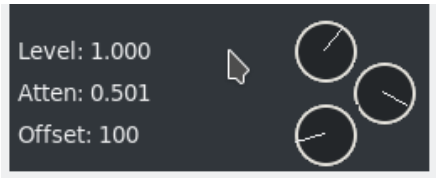


Figure 10.17: *The 3 dials of Echo plugin*

Level represents the volume adjustment.

Atten is attenuation which is a general term that refers to any reduction in the echo reflection. Sometimes called *loss*, attenuation is a natural consequence of signal transmission over long distances.

Offset is the lag in the attenuated echo signal. Offset means adding a DC level to a signal. It offsets the signal up or down in a DC sense without changing the size of the AC part of the signal. When you add an audio clip to the Timeline, the clip plays back from the beginning of the source audio file. The point in the audio file where the clip starts playing is called the offset. By default, a clip's offset is zero, the beginning of the source audio file. You can change the offset so that the clip starts playing from a later point in the source audio file.

10.7.13 EchoCancel

EchoCancel is the process of removing echos from audio in order to improve the quality. Echo cancel may be needed because an audio recording was done in a room that led to echo generation or there was some kind of unwanted feedback. There are many controls for the EchoCancel plugin which are defined here. However, the first thing you will see when you bring up the plugin, is the top portion that is black which will show a + in the middle when you mouse over it. Once you start playing audio, you will see the cepstrum spectral data inside the window. A cepstrum results from taking the inverse Fourier transform (IFT) of the logarithm of the estimated spectrum of a signal. It is used to identify the period of the echo in the audio. It is recommended to just set the *Mode* to On but the below defined parameters can be utilized by professionals (figure 10.18).

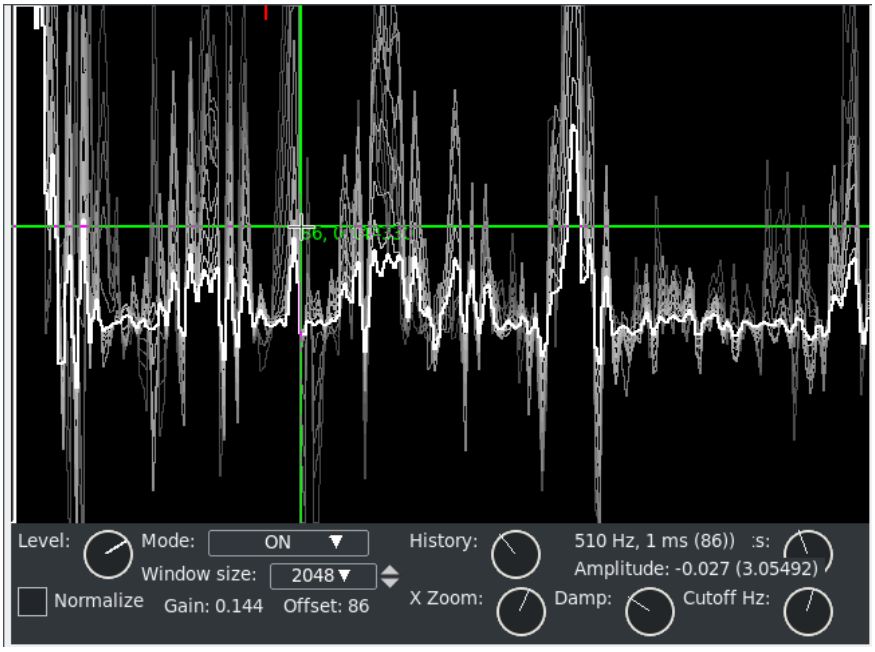


Figure 10.18: *GUI for EchoCancel with crosshair and mode set to ON*

Normalize: audio normalization adds variable amounts of gain to an audio recording to bring the average or peak amplitude to a target level (the normal amount), on an ongoing buffer by buffer basis. This is to make the cepstrum graphical data appear between 0 and 1. Checkmark appears if ON.

Level: scale factor used to draw the cepstrum output when normalize is not in effect.

History: number of previous cepstrum outputs redrawn as fading graphical data.

X Zoom: X axis scale factor to magnify low frequency cepstrum graphical output.

Damp: echo envelope decay factor used to smooth the cepstrum/correlation data.

Peaks: number of maximal envelope values used in the echo gain calculation.

Cutoff Hz: low frequency cutoff value to prevent beat frequency (*heterodyne*) echo canceling.

Mode: *MAN*, *Off*, or *On*. When Off is selected, the plugin is not active. When MAN is used, the only one peak is used for the echo gain envelope. It is set by pressing mouse button 1 (LMB) in the ceptstrum graphical output. The Gain and Offset are updated as the pointer drag operation resets the indicated gain and offset values. When On is selected, the echo gain envelope is automatically calculated by cepstrum and auto-correlation of the input audio in the last window size audio samples.

Windows size: parameter can be set to *Default*, 1024, 2048, ...*doubled values*... up to 262144.

Amplitude: the cepstrum value at the drag point during manual envelope selection.

Gain: echo gain setting determined by manual selection.

Offset: echo period setting determined by manual selection. The *Hz* (frequency), *ms* (millisecond duration), and sample offset (audio samples) as determined by manual selection.

10.7.14 Flanger

It's a single-track effect. If you apply it to multiple tracks each will work on its own track independently of the others. It consists of making a copy of the original sound wave and then playing it over the original one with a certain delay. The resulting signal (Output) will then be the sum of the two waves and will have peaks where the two values add up and gaps where the two values compensate each other. The result is a more *evanescent* and *metallic* sound. Much, however, depends on the intensity of the effect. The delay introduced consists of two distinct components: the *constant delay* and the *oscillating delay*. Their sum constitutes the total delay of the effect (figure 10.19).

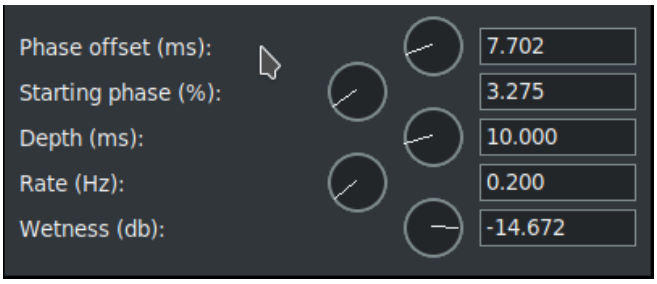


Figure 10.19: *GUI of configuration for Flanger plugin*

Phase Offset : it is the constant delay. Once set, its value does not change (unless we change it, for example by use of keyframes) for the duration of the effect.

Starting phase %: is the point of oscillation where we start the oscillating delay; basically it is the attack value at which the effect starts the calculations. Not to be confused

with the point on the timeline where we apply the effect. It only matches this for the 0% value. The position on the timeline where we want to start the flanger at a given starting phase value can be chosen using keyframes.

Depth : It is the oscillating delay. This value determines the amplitude variation of the delayed (wet) signal phase. This oscillation will be maintained for the entire duration of the effect unless we change it.

Rate : is the speed at which we apply the oscillating delay. Low values indicate a lower oscillation frequency, a high value a rapid succession of oscillations.

Wetness : indicates how much of the original (dry) signal is taken into account compared to the delayed (wet) signal.

10.7.15 Freeverb

Adds effect of multiple decaying echoes to audio signals based on a specific algorithm. Common use of reverb is to simulate music played in a closed room.

10.7.16 Gain

Add gain, input level, to increase/decrease loudness.

10.7.17 Interpolate

Generate a smooth curve based on sound creating a certain softness. There are no controls.

10.7.18 Invert Audio

Reverses the numerical sign of the digital audio. There are no controls.

10.7.19 Live Audio

The Live Audio effect reads audio directly from the sound card input in real time. It replaces any audio on the track so it is normally applied to an empty track. To use Live Audio, highlight a horizontal region of an audio track or define In and Out points. Then drop the Live Audio effect into it. Create extra tracks and attach shared copies of the first Live Audio effect to the other tracks to have extra channels recorded. Live Audio uses the sound driver selected in `Settings → Preferences → Playback A → Audio Out for recording`, but unlike recording it uses the playback buffer size as the recording buffer size and it uses the project sample rate as the sampling rate. These settings are critical since some sound drivers can not record in the same sized buffer they play back in.

Live audio has been most reliable when ALSA is the recording driver and the playback fragment size is 2048. Drop other effects after Live Audio to process sound card input in realtime. With live audio there is no read-ahead, so effects like compressor will either delay if they have read-ahead enabled or playback will under-run. A potential problem is

that sometimes the recording clock on the sound card is slightly slower than the playback clock. The recording eventually falls behind and playback sounds choppy. Live Audio does not work in reverse.

10.7.20 Loop Audio

Loop some number of samples of audio over and over.

10.7.21 Overlay (Audio)

Overlay has parameter settings of top or bottom for the track and add or multiply for the operation.

10.7.22 Pitch Shift

Like the time stretching methods, there are three pitch shifting methods: *Pitch shift*, *Resample*, and *Asset info* dialog. Pitch shift is a realtime effect which can be dragged and dropped onto recordable audio tracks. Pitch shift uses a fast Fourier transform (FFT) to try to change the pitch without changing the duration, but this introduces windowing artifacts. Because the windowing artifacts are less obtrusive in audio which is obviously pitch shifted, Pitch Shift is mainly useful for extreme pitch changes. For mild pitch changes, use Resample instead. Another way to change pitch slightly is to go to the Resources window, highlight the media folder, right click on an audio file, click on *Info*, then adjust the sample rate in the Info dialog to adjust the pitch. This method also requires left clicking on the right boundary of the audio tracks and dragging left or right to correspond to the length changes.

10.7.23 Remove Gaps

Remove silent gap (below *DB* threshold) which persist for more than the time limit.

10.7.24 ResampleRT

Allows you to convert an audio file from one sample rate to another. This effect works similarly to ReframeRT in videos.

Input / output > 1	fast rate
Input / output < 1	slow rate

10.7.25 Reverb

Reverb uses reflections of sound to add depth and fullness; the sound will seem to come from a space that can go from a small bare room to large natural valleys, cathedrals, etc. The reverb is made up of a group of echoes that occur at the same time making it feel

like a single effect. Basically simulates creation of a large number of reflections, like lots of walls, which build up and then decay. You can use the reverb plugin to mix tracks together to simulate ambiance because it is a multitrack effect. The configuration window (figure 10.20) shows a graph of the full band pass filter frequencies.

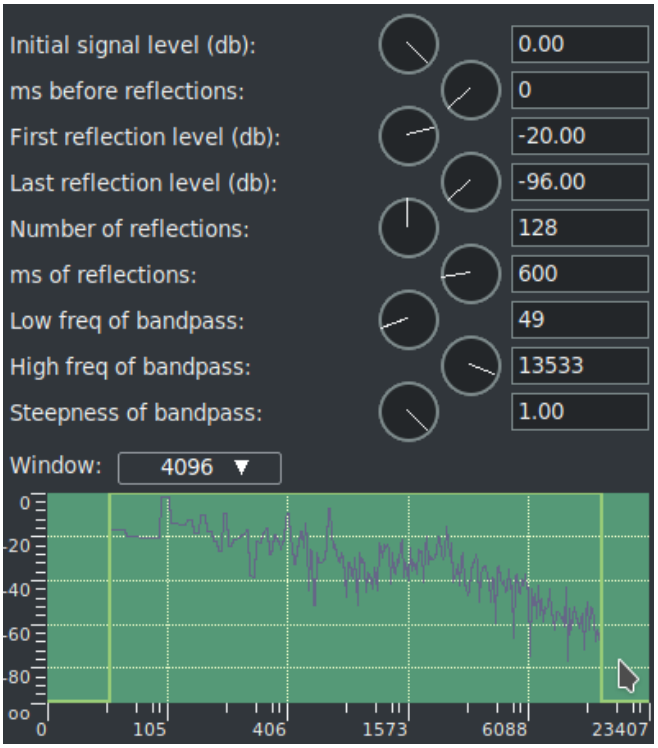


Figure 10.20: *GUI of configuration for Reverb plugin*

Initial signal level (db): Allows you to set the level of the reflected signal. At 0 you start from its maximum level. Decreasing the signal volume will give more presence to the original signal.

ms before reflections : indicates when to start the reflected sounds in relation to the original sound (delay).

First reflection level (db): the sound level of the reflection that starts first.

Last reflection leve (db): the level of the last reflection. It is weaker than the first.

numbers of refelctions : the number of reflections can be set as desired as long as there is sufficient CPU to handle it. With a few reflections you get closer to the Chorus effect. You can start from values of 100 - 150, up to a maximum of 255.

ms of refelction : Sets the action time of the effect. With high values, the sound is more clear and sharp

Low freq of bandpass and **High freq of bandpass**: allow you to set the frequency limits (range) on which the effect acts.

Steepness of bandpass : allows you to adjust the slope of the frequency range limits imposed by the previous items. With the value 1.00 we have a vertical edge that clearly separates the range of frequencies on which to act from the remaining ones (as you can see in figure 10.20). By sloping the edge we have a certain overlap and a smoother effect.

Window : determines the number of frequency samples taken into account by the effect for its calculations. The higher the number, the smoother the effect; but more CPU is used.

10.7.26 Reverse Audio

Apply reverse audio to an audio track and play it backwards. The sound plays forward. Be aware when reversing audio that the waveform on the timeline does not reflect the actual reversed output.

10.7.27 SoundLevel

Effect rewritten and improved to handle fragments. Displays the Max/RMS sound level in decibels.

10.7.28 Spectrogram

Effect rewritten and improved. Visual representation of the sound levels at specified frequencies as they vary with time.

10.7.29 Synthesizer

Generate synthesizer sounds; to set key data, turn on Generate keyframes while tweaking (figure 10.21).

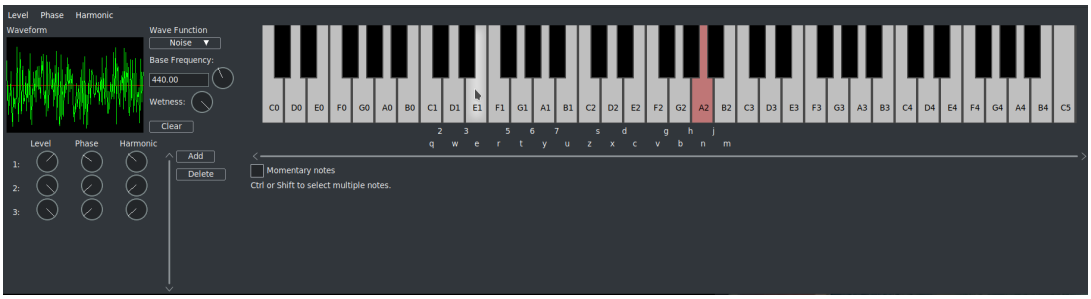


Figure 10.21: *GUI for Synthesizer*

10.7.30 Time Stretch RT

Change the speed of an audio signal without affecting its pitch.

10.7.31 Tremolo

It serves to give various vibes and vitality to the sound by modulating the amplitude of the sound signal and the delay (figure 10.22).

NOTE: There is often confusion between Tremolo and Vibrato, but the vibrato is the periodic variation of the height of a note (frequency) while the tremolo is the periodic variation of the height (amplitude) of the sound wave.

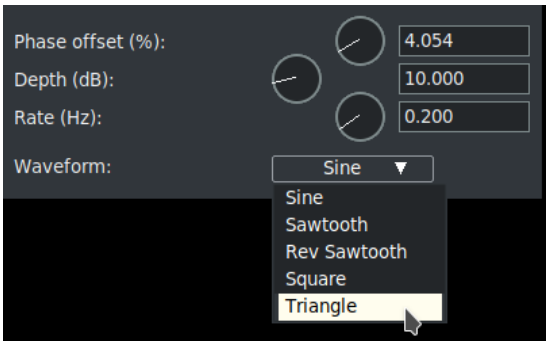


Figure 10.22: *GUI of configuration for Tremolo plugin*

Phase offset (%): is where the oscillation effect begins. At 0% we are at maximum volume; to lower the volume, increase the percentage.

Depth (dB): is the oscillation damping value. You can only decrease the level, you cannot increase it.

Rate (Hz): is the speed at which the oscillations are repeated.

Waveform : you can choose waveform algorithms to use in effect calculations. Currently available waveforms are: *Sine*; *Sawtooth*; *Rev Sawtooth*; *Square* and *Triangle*.

10.8 Audio Ladspa Effects

Ladspa effects are supported in realtime and rendered mode for audio. These audio effects are supported since CINELERRA-GG implements the LADSPA interface as accurately as possible. Besides the supplied LADSPA effects[4], additional LADSPA effects can be enabled by setting the `LADSPA_PATH` environment variable to the location of your LADSPA plugins:

```
export LADSPA_PATH=/usr/lib/ladspa
```

[4] credit Steve Harris

10.9 Audio LV2 / Calf Plugins

LV2[5] is an open standard for audio plugins using a simple interface with extensions which add functionality to support audio software. These plugins were written by external developers and provide additional audio effects to CINELERRA-GG audio without having to change CINELERRA-GG every time. Because the LV2 plugins are separate from CINELERRA-GG Infinity, if one fails or does not perform as expected, CINELERRA-GG should stay running and you will have to contact the programmers responsible for that plugin for a fix.

Typically, a user OS has specialized package groups installed. It is difficult to create one build of CINELERRA-GG to accommodate all potential LV2 plugins. Specifically for the *Calf-Studio LV2 plugins*, you should install the *Calf Plugins* package. The user's computer must have *gtk-2-runtime* installed, which seems to be automatically done already for most distros. For users doing their own builds, you can build CINELERRA-GG without LV2 support by including `--without-lv2` in the configure step. The default build is `--with-lv2=yes` and requires that *GTK-2-devel* must be installed or the build will fail and notify you. In addition for some newer distros, you will need to install *lv2-calf-plugins-gui*; for example Fedora version 32.

LV2 plugins have their own category in the *Audio Plugins Visibility* as lv2. There is a simple text interface which is available via the usual *Show controls* button when the plugin is attached to the audio track. This window has a Reset button to get back to the default settings. To change a value of one of the parameters, highlight that parameter and type in the new value in the topmost text box and then hit Apply to take effect – the reason for requiring hitting apply is so that the audio is not moving all over the place while you are still typing a value. More easily, you can just move the *pot dial* or the *slider* bar which take effect automatically.

CINELERRA-GG's buffer size setting may cause a delay in activation of the changes you make taking effect, so you can lessen the time by using a small buffer. Notice that 1024 samples at 48000 samples per sec is only $\frac{1}{50}^{th}$ a second. This is not a lot of time to shuffle a bunch of stuff. Short buffers produce low latency, but no time for complex programs or lots of stacked effects. Bigger buffers allow for more complex setups.

To set the buffer size:

`Settings → Preferences → tab Playback A → section Audio Out → variable Playback buffer samples`

However, be forewarned that due to variability in the lv2 plugin programming code, some of the plugins only work with the minimum buffer size of 1024. In these cases, what you will see is the main track canvas cursor just bounces back and forth over a very small area in the timeline. This does not crash CINELERRA-GG but you will have to remove the plugin to continue working. You can specify a certain set of LV2 plugins to use by setting

[5] Optional Feature - OS dependent

LV2_PATH as shown below before starting CINELERRA-GG – include a colon (:) separator for multiple paths. The default path for most operating systems is /usr/lib64/lv2. To list the system installed lv2 plugins key in: lv2ls.

```
export LV2_PATH=/tmp/j/balance.lv2/usr/local/lib/lv2/:/usr/local/lv2
```

If there is no default LV2_PATH set automatically, the value will be $CIN_DAT/lv2, which is a placeholder only so that no lv2 plugins will be loaded. When there is no system LV2_PATH set it is important to note, that if you do want lv2 plugins loaded, you must set the correct path in:

Settings → Preferences → Interface tab → Default LV2 → directory path name

When you change this field, cin will automatically restart and load the newly specified lv2 plugins. If when switching LV2_PATH or if the lv2 audio plugins are not displayed/usable in the Resources window, you can execute a reload via:

Settings → Preferences → Interface tab → Reload plugin index or else before you bring up CINELERRA-GG, delete $HOME/.bcast5/Cinelerra-GG_plugins so that the plugins get properly reloaded.

There are some lv2 plugins that display a *glitzy* UI (User Interface); for example the *Calf plugins*. For these LV2 plugins, if you want that to automatically come up without having to click on the UI button on the simplified UI interface, there is a flag to enable that. It is at:

Settings → Preferences → Interface tab, Operations section

then check the *Auto start lv2 gui* Flag

Below is a screencast showing the auto start gui flag and the LV2_PATH default directory path on the bottom line. Note the highlighted *Reload plugin index* which will be executed if OKed (figure 10.23).

There is also a blacklist that prevents known problematic-for-CINELERRA-GG lv2 plugins from loading to avoid crashes. If others are found to have problems, once informed about them, they will be added to this blacklist. In order to determine which lv2 plugin causes a SEGV on CINELERRA-GG startup, you can start from a terminal window and you will see each plugin that is being loaded and the last one shown before the crash is a bad plugin. However, many of the plugins causing a crash are due to not having been compiled on your current system with the current compiler so may actually work correctly on other user systems and so will not be added to the CINELERRA-GG-wide blacklist. You can either recompile the problematic plugin, or modify your own blacklist which you will have to

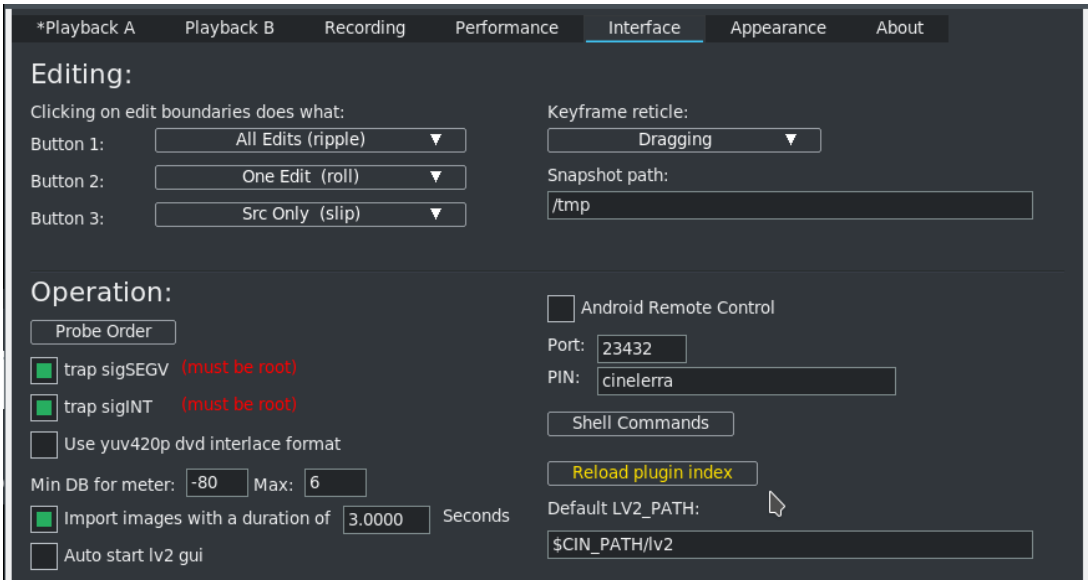


Figure 10.23: *Reload plugin index in yellow and Auto start lv2 gui unchecked*

maintain and save so as not to be written over when loading a new build.

Note the UI button in the upper right hand corner above the Reset button (figure 10.24). If you click this button, a glitzy interface window comes up (if available) for changing variable values. It is possible that a bug in the LV2 plugin causes the glitzy window to appear as blank and then die, but in that case the original simple text window might still work – in either case, if the timeline movement hangs, just detach the plugin to continue your current session. There is an environment variable that you can set, `BC_TRAP_LV2_SEGV`, to get a dump of the failure which may be helpful for debugging.

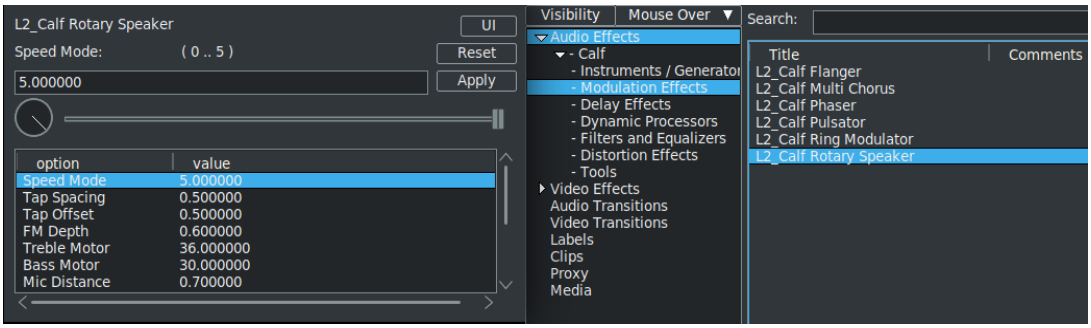


Figure 10.24: *Screencast of simple text interface in the middle of the screen for a Calf LV2 plugin*

When the glitzy ui is up, the simple text window remains up also since it is the CINELERRA-GG side and keeps track of the value changes so they remain in effect for further usage of the plugin. Changes to one or the other will occur in both with the exception of certain features in the glitzy window which are not communicated correctly back to CINELERRA-GG; for example a reset button – the simple interface Reset button must be used instead.

To change values in the glitzy window you use the mouse and move up or down unlike a knob that turns! (Figure 10.25)

Figure 10.25: *Screencast with a Calf plugin glitzy window that appears when clicking the simple interface UI button.*

In order to test a particular plugin without bringing up CINELERRA-GG, especially for ones that do not operate, it is possible to manually display an lv2ui gui with:
`/cin-path/lv2ui <lv2-uri>`
For example:

```
/tmp/cinelerra-5.1/bin/lv2ui http://calf.sourceforge.net/plugins/Flanger
```

If using the AppImage version and the Calf/LV2 plugins are not working for you, try doing the following which has been shown to work on Ubuntu 16.

1. Copy the "calf" and "lv2" folders from /usr/lib/ to /home/user/plugins_lv2-calf_for_Cin/.
2. After starting up CINELERRA-GG, change Settings-> Preferences...-> Interface tab, Default LV2_PATH to $HOME/plugins_lv2-calf_for_Cin/lv2 .
3. In the /home/user/plugins_lv2-calf_for_Cin/lv2/ directory remove any of the plugins that do not seem to work.

10.10 Video Effects – Native

10.10.1 1080 to 480

Most TV broadcasts are received with a 1920 × 1080 resolution but originate from a 720 × 480 source at the studio. It is a waste of space to compress the entire 1920 × 1080 if the only resolvable details are 720 × 480. Unfortunately resizing 1920 × 1080 video to 720 × 480 is not as simple as shrinking it.

At the TV station the original 720 × 480 footage was first converted to fields of 720 × 240. Each field was then scaled up to 1920 × 540. The two 1920 × 540 fields were

finally combined with interlacing to form the 1920 × 1080 image. This technique allows a consumer TV to display the re-sampled image without extra circuitry to handle 720 × 480 interlacing in a 1920 × 1080 image.

If you merely deinterlace the 1920 × 1080 images, you would end up with resolution of 720 × 240. The *1080 to 480* effect properly extracts two 1920 × 540 size fields from the image, resizes them separately, and combines them again to restore a 1920×480 interlaced image. The scale effect must then be applied to reduce the horizontal size to 960 or 720 depending on the original aspect ratio.

The tracks to which *1080 to 480* is applied need to be at 1920 × 1080 resolution. The project settings in `settings → format` should be at least 720 × 480 resolution. The effect does not know if the first row in the 1920 × 1080 image belongs to the first row of the 720 × 480 original. You have to specify what the first row is in the effect configuration. The output of this effect is a small image in the middle of the original 1920 × 1080 frame. Use the projector to center the output image in the playback.

Finally, once you have 720 × 480 interlaced video you can either apply *Frames to Fields* or *Inverse Telecine* to further recover original progressive frames.

10.10.2 1080 to 540

Extracts two 1920 × 540 fields from 1920 × 1080 image, resizes them separately, and combines them to 1920 × 540 interlaced image.

10.10.3 Aging TV

This effect is the one to use if you want to achieve an old movie or TV show look. It will put moving lines up and down the movie as well as putting snow on the video. Use it along with *Brightness/Contrast* and *Color Balance* to make your movie look like a really old black and white movie. This came from `https://effectv.com`.

10.10.4 Alpha

Allows you to apply an alpha value (transparency) to one or more tracks or one or more edits. You can use the slider or have maximum precision by entering numbers in the input box. You can use the *Clear* button to reset to the default value. Being also keyframable, it allows an excellent variety and possibility of use in the most disparate ways.

10.10.5 Auto Scale

Automatically scale to a specified size.

10.10.6 Blend Algebra and Blend Program

The Blend Algebra/Blend Program plugins provide the advanced feature of combining and modifying color pixels and transparencies of several tracks according to a mathematic algorithm written entirely by the user in the form of compact and simple piece of code. Such user defined algorithms (blend functions) are compiled, dynamically linked and immediately executed inside CINELERRA-GG on the fly without the need to restart the application, reload the project, reattach plugins, and so on.

The following description is based on the `/cinelerra-5.1/doc/README.blendalg` that accompanies the code by the author of both the code and the document. It is the authoriative documentation; the most detailed/original source and has additional commentary and programming information. If you encounter problems or confusion when using these plugins, always refer directly to that document here. If running CINELERRA-GG from an *AppImage*, testing has shown that it should work as well as if run from a build - just be aware of filename extensions that appear by default in the input field for the funtion name and change them accordingly.

The idea behind these plugins[6] is that the user can program sufficiently short and simple blending algorithms, called *Functions*, by following detailed instructions with very little programming skills required. Blend Algebra and Blend Program come with a library of Functions (*System Library*) that mimic the 30 overlay effects found in the Patchbay or Overlay plugins. The user can modify these Functions as needed or create new ones. To look at these example functions, there is an `Edit...` button in the respective plugin dialog. You will need a text editor to do so, which by default is *emacs* if it is installed. If not, you can set your favorite editor via the corresponding environment variable in a terminal (`export CIN_EDITOR='name editor'`) and then start CINELERRA-GG from that same terminal.

If you can not wait to get started, checkout these examples here. You will also find test projects to download to learn more about the plugins and to see the various ways they can be used to create your own functions.

Both plugins, Blend Algebra and Blend Program, are similar. They are described together along with their differences. The main difference between the two plugins is that Blend Algebra is like a function; it combines its arguments (tracks) and yields the result which is placed into the single track configured for output. However, Blend Program does not produce a function result, but instead directly modifies its arguments. If more than one track is to be modified, Blend Program is the only choice. If a single track is to be processed, Blend Program is the best choice and might run slightly faster. Any Blend Algebra function can be rewritten to become a Blend Program function. But if a function processes several tracks and modifies only one, it will run faster in Blend Algebra which has more controls in its plugin dialog.

How the plugin works is defined by the user. The word *function* is used to denote this

[6] credit to Georgy

for either of the two plugins throughout this description. These functions are for their plugins the same as the plugin shared objects in the main CINELERRA-GG binary. But these functions are simple enough that they can be easily written by users and manipulated in binary or in source form. The plugins make it possible to have a user library and a system library of functions. Thus, an experienced user can have a number of such functions to reuse in many projects.

The plugin is attached to a track as usual. Generally, it needs several tracks as arguments and it has to be attached to the additional tracks as a shared effect.

All the parameters described below are keyframable, including user functions themselves. The top area of the dialog menu deals with programmatic aspects. The middle area of the dialog is color specific. The bottom area of the dialog has controls for the argument track ordering, similar to that of the Overlay plugin.

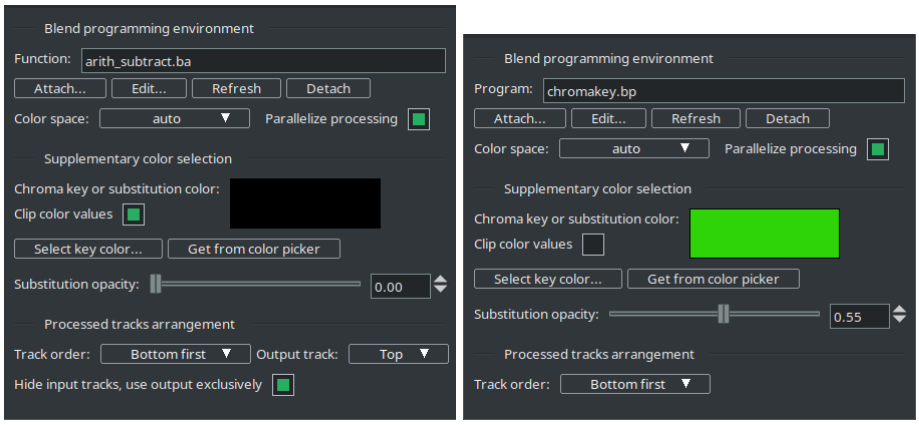


Figure 10.26: *Blend Algebra and Blend Program GUI*

Blend programming environment

Function This textbox contains the filename of the user's function source text. The most convenient method to select the function is via the `Attach...` dialog, but you can just enter its filename directly. If the function pathname does not start with '/', it is interpreted relative to the main project directory. An empty function name means that no function will be executed and the tracks will remain unchanged. If the entered filename does not exist, again nothing will be done, but in this case the user can create a source file via the `Edit...` button. An error message will be shown when a directory is selected in the file selection dialog instead of a filename. Although you can use any filename, it is highly recommended to use distinct suffixes: `.ba` for Blend Algebra, `.bp` for Blend Program or you will have to keyin specific filenames in some cases.

Attach... This button opens the file selection dialog menu as usual for Cinelerra. By default it shows only files with the recommended suffixes (`*.ba` or `*.bp`), although

this can be changed by specifying a filter. The user can select an existing file or enter a filename. You can exit the dialog with `Cancel` so that nothing is changed or with `OK` for changes to be in effect. In that case, the selected function will be set as current, and its name will appear in the *Function: input field* above. An error message will be shown if the user presses `Edit...` with an empty function name or when a directory is selected in file selection dialog instead of a filename. If the entered filename does not end with `.ba` or `.bp`, the corresponding suffix will be appended automatically. If you use a filename with another suffix or none at all, you have to change the name directly in the *Function: input field* after exiting the File Selection window.

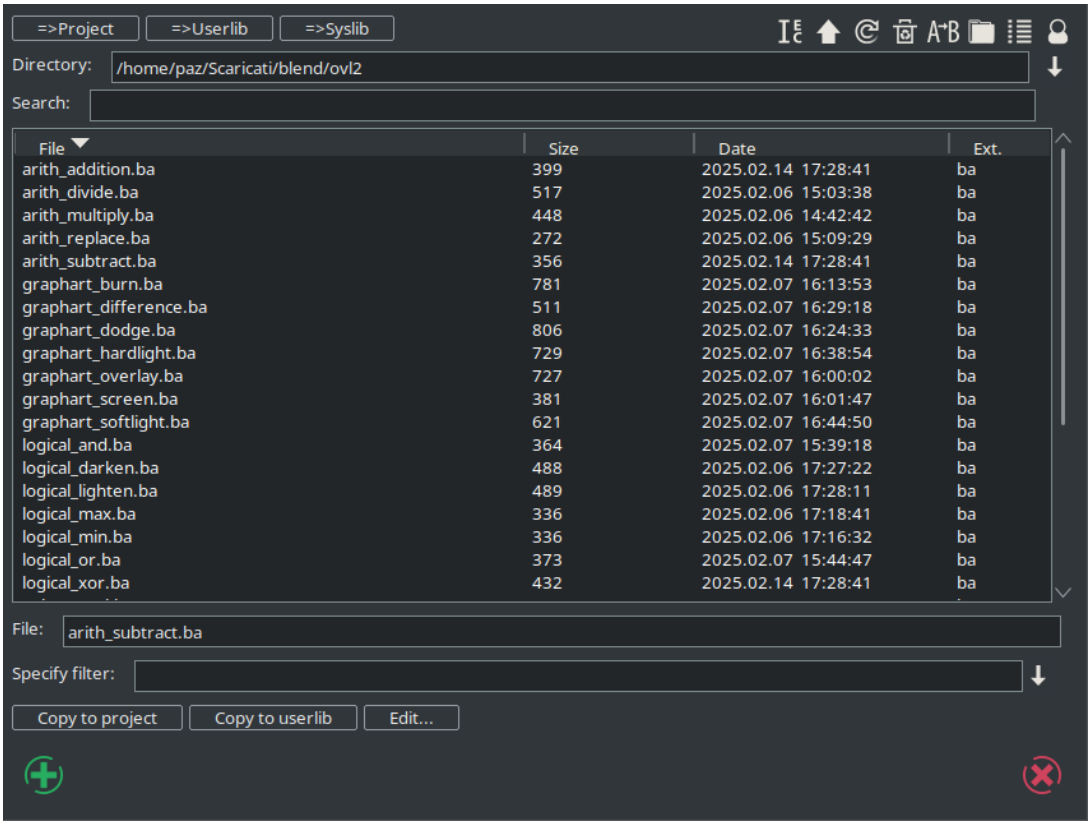


Figure 10.27: *File Selection window*

The file selection dialog menu has six additional buttons specific to the Blend Algebra/Blend Program plugins. The three buttons on top left change the current directory; the three buttons on bottom left do simple file operations.

=>Project display the files in the project directory (where the main project .xml file resides).

=>Userlib display the files in the user library.

=>Syslib display the files in the system library.

Copy to project copies the currently selected function source file into the project directory.

Copy to userlib copies the currently selected function source file into the user library. There is no button to copy a function into the system library because the system library is meant for functions distributed with Cinelerra, which can get overwritten on reinstallation or upgrade. For details on "copying to/from" project, userlib, or syslib, see "User library, System library, and Cinelerra Project directories" in a subsequent section.

Edit... This is used in the same way as the Edit... in the main plugin window as described next.

Edit... Open the currently selected function source file in the configured external editor. The editor that will be used is defined by the environment variable `$CIN_EDITOR`, with emacs being the default if not redefined. It must be a windowed editor, like *GNU Emacs* (default), or KDE's *Kate*, or *Gedit*, or ancient *Xedit*, not a bare console editor like *vi*.

```
export CIN_EDITOR=kate
export CIN_EDITOR=gedit
...

$ ~/cinelerra5/cinelerra-5.1/bin/./cin
```

If using a console editor, it must be defined to run in a terminal emulator, like *xterm*, or KDE's *Konsole*, for example:

```
export CIN_EDITOR='konsole -e vi'
$ ~/cinelerra5/cinelerra-5.1/bin/./cin
```

This is the same `Edit...` button as in the file selection window of the `Attach...` button.

Refresh The external editor is started in the background so after editing and saving a function, the user can leave the editor in that window. Alternatively the user can edit a function outside of CINELERRA-GG. In order not to check modification times of all the functions on each video frame, CINELERRA-GG has to know when a function can get changed. That is what pressing `Refresh` is for. It indicates that now is the time to check functions, and to recompile some of them. You can expect immediate recompilation of the function which is under the timeline cursor (if any), that is if the source became newer than the shared object. But any other functions not under the cursor will not be recompiled until first access.

Detach This button clears the function name in the *Function: input field* and if there was a function in effect, it will be cleared in the tracks.

Color Space: Here is defined in which color space the frame pixels are to be passed to the function. The obvious choice is the default `auto` so that you will be working in the color space which is required by the function itself. In addition, RGB, YUV, and HSV color spaces are supported as well as the choice `of project`. With this choice, you will be working in the native color space of the project (as defined in `Settings → Format`). If the required color space does not match the native one, pixel color components will be transformed to the required space as floating point numbers. After returning from the function, pixels of the result will be transformed back to the native color space of the project.

Parallelize processing This checkbox, ON by default, lets you execute user function code in a parallelized (multithreading) manner, provided that the function itself supports this. It is described further on as to how the function code is executed.

Supplementary color selection

There may be illegal floating point operations in the user's code, such as division by zero producing infinity, or square root from minus unity producing NaN (not-a-number). Even if only legal finite floating point numbers are computed, the results may go out of range. This dialog section controls what to do in such cases.

Clip color values If this checkbox is on, which is the default, first the function results will be clipped to range. Clipping takes place before back transformation to the project color space. The bounds are defined as follows:

RGB: [0.0 .. 1.0] for R, G, B
YUV: [0.0 .. 1.0] for Y, [-0.5 .. +0.5] for U, V
HSV: [0.0 .. 360.0] for H, [0.0 .. 1.0] for S, V
Transparency: [0.0 .. 1.0] for Alpha

If the project color space is `RGB(A)-8 Bit` or `YUV(A)-8 Bit`, clipping occurs unconditionally because of intrinsically restricted dynamic range of 8-bit storage per channel. For `RGB(A)-FLOAT` clipping can be switched off. What occurs with unclipped pixels later, depends on further destinations in the internal video processing pipeline. Moreover, displaying drivers handle unclipped colors differently. For example, an RGB pixel with unclipped red component `R=-1.0` and two other components `G=B=0.0` is displayed as black (that is with *R* clipped to zero) by the OpenGL driver, but as red (that is taken with opposite sign, *R=1.0*) by the X11 driver. Explicit clipping in the plugin eliminates such driver dependency.

Chroma key or substitution color: Even after optional clipping, there can still remain some non-finite results, like NaN. These results are bad and should be eliminated as soon as possible. If any of color components or alpha channel is NaN, the complete pixel (all components) is replaced with the configured color and the configured opacity. *Substitution color* is passed as additional arguments to user functions and can be used for artistic purposes, such as chroma keying.

Select key color... This button opens *color selection* dialog to select the color which will be used as substitution for NaNs or infinities in the results.

Get from color picker Use this button to copy the color selected earlier in the color picker tool of the CINELERRA-GG Compositor window.

Substitution opacity: The slider sets the opacity (*alpha-channel*) value to be used together with substitution color. Substitution color and opacity are passed as additional arguments to user functions and can also be used for artistic purposes.

Processed tracks arrangement

Track order: If the plugin is attached to N tracks, its user functions get pixels of the tracks numbered from 0 to N-1. This defines which of the tracks is to be the first (i.e. number 0), either *top* or *bottom*, much like in the *Overlay* plugin.

Output track: This defines into which track (*top* or *bottom*) the result will be placed, similar to the Overlay plugin. This control applies to *Blend Algebra* only.

Hide input tracks, use output exclusively If this checkbox is `ON` (the default), frames of the tracks the plugin is attached to are cleared and then replaced by the function result. If the checkbox is `OFF`, only the output track frame is replaced by the result; all the other tracks are left unchanged. By comparison, the Overlay plugin has no such option and its input tracks are always cleared as in the *on* state. This control applies to the *Blend Algebra* plugin only.

All the parameters are keyframable, including user functions themselves. That is, a plugin can have several functions attached at the same time and switch between them while rendering. The color and alpha channels of the substitution (key) color are interpolated between keyframes; the other parameters are switched.

User library, System library and Project directories

The user library directory is that defined by the environment variable `$CIN_USERLIB`, or if not defined, `$HOME/.bcast5lib` as a fallback default. The functions for Blend Algebra reside there in the subdirectory `dlfcn/ba` and for Blend Program in `dlfcn/bp` (*dlfcn* stands for *dynamic loaded function*). Having user functions in such a library in a single place makes it convenient to reuse the same functions in different projects. The `Copy to userlib` button in the `Attach...` file selection dialog is handy to save such functions in the user library for future use.

The reason to define user library `$HOME/.bcast5lib` by default (and not `$HOME/.bcast5`) is the following. There is a common workaround: if a newer version of CINELERRA-GG does not start, to delete `$HOME/.bcast5` completely. Doing so, all user functions would be deleted if they were saved there.

The system library is defined by the environment variable `$CIN_DAT`, the CINELERRA-GG installation path, and under the same subdirectories `dlfcn/ba` or `dlfcn/bp`. System library is meant for functions distributed with CINELERRA-GG.

When a complete project is to be exported, or fed to a render farm, it is often necessary to place all the resources under the same directory where the project `.xml` file resides. If some functions are used which reside elsewhere (for example, in a user library), they could not be found after exporting the project to another computer. The `Copy to project` button in the `Attach...` file selection dialog is used to copy such functions to the project directory in a convenient manner.

Syntax of Blend Algebra / Blend Program functions

Blend Algebra/Blend Program functions are written in C, and with usage of a set of special cpp macros hiding from the user all the Cinelerra internals. This approach makes it easy to write functions of moderate complexity with almost no background in C programming. All macros defined for user functions are written in capital letters. A typical Blend Algebra function consists of the following logical blocks.

```
<optional static variable definitions>

BLEND_ALGEBRA_INIT

<macros and/or C statements for the INIT (preparation) phase will be executed once
    ↪ before processing of each frame>

BLEND_ALGEBRA_PROC

<macros and C statements for the PROC (processing) phase will be executed
    ↪ repeatedly for each pixel in the frame>

BLEND_ALGEBRA_END
```

The macro `BLEND_ALGEBRA_STOP` terminates execution of the `INIT` or `PROC` phase immediately and returns to the calling plugin.

The structure of a Blend Program function is the same, only the word `ALGEBRA` in all macros must be changed to `PROGRAM`. The following macros can be used in the INIT phase:

```
COLORSPACE_RGB
COLORSPACE_YUV
COLORSPACE_HSV
```

Use any of these COLORSPACE statements to declare the working color space inside the user function. If there is no such declaration, the native color space of the project (RGB or YUV) will be used.

`REQUIRE_TRACKS` (<n of tracks>)
Declares the minimum required number of tracks the function uses. If the effect is attached to more tracks, it is fine. If there are less tracks, it is an error and the function is not executed. The absence of such declaration means that the function can process any number of tracks (one or more).

`PARALLEL_SAFE`
This declaration means that it is safe to execute the function in parallel. Without this statement the function will be executed sequentially, independent of the state of the *parallelization* checkbox in the plugin GUI.

The following special variables (macros) can be used for queries:

`PARALLEL_REQUEST`
1 == parallelization requested in the GUI
0 == parallelization switched off

`TOTAL_TRACKS`
The actual number of tracks passed to the function.

`HAS_ALPHA`
1 == the project color space has alpha channel (transparency)

`WIDTH, HEIGHT`
The dimensions of processed frames in numbers of pixels.

The following variables (macros) can be used in the PROC phase. In addition to the next variables, `TOTAL_TRACKS, HAS_ALPHA, WIDTH, HEIGHT` can be used as well.

Color (and alpha) components of pixel from i-th track. Track numbering is *0-based*, as standartized for C arrays. So, the first track has number 0, the last one - number TOTAL_TRACKS-1. All the color components are floating point values of the C type 'float'. Here the letters R, G, B, H, S, V, etc. are for readability. If the user has declared `COLORSPACE_RGB`, and then written something like H(i), it does not mean that he should get the H-component of color autoconverted to the HSV color space. Actually R, Y, H are mutually identical to the 1-st color component, so are G, U, S to the 2-nd, and B, V to the 3-rd one.

```
R(i), G(i), B(i)
Y(i), U(i), V(i)
H(i), S(i), V(i)
```

```
A(i)
```

Corresponding color components of the key (substitution) color in the working color space of the function and the opacity slider.

```
KEY_R, KEY_G, KEY_B
KEY_Y, KEY_U, KEY_V
KEY_H, KEY_S, KEY_V
KEY_A
```

Color components for the result, for Blend Algebra only. Blend Programs return no result so do not have these definitions.

```
R_OUT, G_OUT, B_OUT
Y_OUT, U_OUT, V_OUT
H_OUT, S_OUT, V_OUT
A_OUT
```

Integer X and Y coordinates of the current pixel in its frame, in the ranges of [0 .. WIDTH] and [0 .. HEIGHT], respectively.

```
PIX_X, PIX_Y
```

These macros clip color components of the i-th track to the bounds appropriate in the respective color space.

```
CLIP_RGB(i)
CLIP_YUV(i)
CLIP_HSV(i)
CLIP_RGBA(i)
CLIP_YUVA(i)
CLIP_HSVA(i)
CLIP_A(i)
```

Clip all color components (including alpha) for all tracks.

```
CLIP_RGB_ALL
CLIP_YUV_ALL
```

```
CLIP_HSV_ALL
```

Like clipping track color components, but for the result of a Blend Algebra function (not relevant for Blend Programs).

```
CLIP_RGB_OUT
CLIP_YUV_OUT
CLIP_HSV_OUT
CLIP_RGBA_OUT
CLIP_YUVA_OUT
CLIP_HSVA_OUT
CLIP_A_OUT
```

Clipping is done according to the inherent bounds of the respective color space as follows:

```
RGB: [0.0 .. 1.0] for R, G, B
YUV: [0.0 .. 1.0] for Y, [-0.5 .. +0.5] for U, V
HSV: [0.0 .. 360.0] for H, [0.0 .. 1.0] for S, V
Transparency: [0.0 .. 1.0] for Alpha
```

H color channel of HSV is brought into range by repeated rotation around 360 degrees; all the others by simple clipping.

`ABS(x), SQR(x), MAX(x,y), MIN(x,y)` absolute value, square, max and min values for floating point arguments. Can be used in any phase of a function.

`TO_RAD(x), TO_DEG(x)` conversion from degrees to radians and vice versa.

Both macros clip the 'x' argument to bounds between y and z. CLIP returns value leaving x unchanged. CLAMP assigns that value to x.

```
CLIP(x,y,z)
CLAMP(x,y,z)
```

All the macros are defined in the text header `$CIN_DAT/dlfcn/BlendAlgebraStart` for Blend Algebra and the analogous header `BlendProgramStart` for Blend Program in the same `dlfcn` subdirectory of the Cinelerra installation directory. These headers are

prepended to the user functions during compilation. The user with a basic C knowledge can easily understand what happens in these headers.

Of course, you can use any valid C statements inside user functions. As they are linked with `-lm`, the standard C math library and with glibc, almost any mathematical or C library function can be called. C-style comments can also be used, and are welcome.

The compilation script

Compilation of Blend Algebra and Blend Program functions is carried out by a Perl script `BlendAlgebraCompile.pl` and `BlendProgramCompile.pl`. Both scripts are distributed in the `dlfcn` subdirectory of `$CIN_DAT`, the Cinelerra installation directory. Like the famous `ContextManual.pl` script of Context Help, the system script at the first call is copied into `$CIN_CONFIG ($HOME/.bcast5)` where it can be edited by the user and adapted according to their needs.

While a regular compilation, the script works in that directory where the function source file resides. First, the `BlendAlgebraStart/BlendProgramStart` header is prepended to the source to produce an intermediate file with the `.c` suffix. This C source is compiled by the system C compiler (gcc by default) and linked with `-lm` to produce the shared object ready for dynamic loading (attaching to the plugin).

Usually the script is executed by the respective CINELERRA-GG plugin when a user function is to be compiled. But the script can also be run by the user explicitly. Execution can be controlled by options starting with the '-' character, the first argument not starting with '-' being the function name to compile. The following options are recognized:

-compile do not check modification time, compile unconditional
-cfile do not remove intermediate .c file
-opt add optimizing options to compiler command line
-debug add debugging options to compiler command line
-warn add warning options to compiler command line
-edit open function source in text editor
-verbose verbose execution
-noapi do not copy itself to `$HOME/.bcast5`
-h, -help, -? print short help and current configuration

Environment variables influencing plugin configuration

In the beginning of the script some variables can be redefined according to the user's needs: the C compiler executable, compiler options for optimization or debugging, and text editor executable. These environment variables are taken into account in various places in Blend Algebra/Blend Program plugins.

`$CIN_CC`: C compiler executable (default: gcc)

$CC: C compiler executable if $CIN_CC is not defined (default: gcc)
$CIN_EDITOR: text editor (default: emacs)
$CIN_DAT (set by Cinelerra binary): Cinelerra installation directory. The Blend Algebra / Blend Program system library with scripts, headers, and functions distributed with Cinelerra resides in the subdirectory $CIN_DAT/dlfcn.
$CIN_CONFIG (set by Cinelerra binary): user's config directory, usually $HOME/.bcast5.
$CIN_USERLIB: user's library (default: $HOME/.bcast5lib). User's functions reside in the subdirectory $CIN_USERLIB/dlfcn.

What comes in the plugins

In the distribution 30 Blend Algebra functions are provided for all the 30 CINELERRA-GG overlay modes according to the formula defined in the CinelerraGG manual. In principle, they are not needed as the built in overlayer engine is OpenGL accelerated and runs faster. Nevertheless, these functions can be handy as starting examples for users who may like to modify the formula in some way. Furthermore, their behavior differs from that of the standard overlayer in several aspects:

- Overlayer produces different results in RGB and YUV color spaces; Blend Algebra yields identical results independently of color space of the project.
- In RGB(A) FLOAT color space, overlayer results can get out of range and then be rendered differently depending on the display driver used. Blend Algebra clipping, when switched on, can prevent this undesirable effect.
- Overlayer always clears its input frames. In Blend Algebra the user has a choice, either to clear them, or to leave them intact.

There are several example functions from totally different areas of application: a user programmable plugin is a really universal tool and can be handy in unexpected cases. A programmer can sometimes make use of this plugin when creating new effects. It has the big advantage that you can more quickly test different combinations without spending time to restart CINELERRA-GG, recompile it, reload project, and reattach plugins, etc. See the examples here.

There is the ydiff.ba function, actually representing the complete *ydiff* application running inside a plugin. In addition, there is a Transition example, and 2 more function examples showing use of the Blend Programs: the chromakey.bp and background.bp functions. They too serve as excellent starting points for further user experimentation. The usage of the provided functions is either self evident or briefly described in the comments of their source files.

NOTE: make install copies the functions into Cinelerra's bin directory in source (*.ba, *.bp) and binary (*.so) forms via cp -a, to preserve modification times of the files. Oth-

erwise, if modification time of the function source accidentally becomes newer than that of its binary, the plugin will repeatedly try to recompile the function. If they are installed system-wide so that the user has no permission to modify them, such recompilations will fail.

Caveats

User's functions can have bugs. Most probable bugs in a user function can be illegal arithmetic such as division by zero or logarithm of a negative number (FPE, floating exception), and out of bounds indexation of arrays (usually leading to SEGV). Most modern processors do not generate FPE. Instead, they generate NaN as the result of an illegal arithmetic instruction. After calling user functions, the plugins test the results not to be NaN and replace them with the configured substitution color if necessary.

SEGV is more problematic. If the user has written in his function, let's say, `REQUIRE_TRACKS(2)`, and then accessed `R(1000000)`, most likely the Cinelerra binary will crash. Although theoretically it could be possible to trap SEGV via `sigaction()/setjmp()/longjmp()`, in Cinelerra it is problematic. In a multithreaded application, which Cinelerra is, only one signal handler for all its threads at the same time is allowed, and Cinelerra already has a SEGV handler which can not be overwritten. So the user is responsible for his indexation bugs, which can easily occur in off-by-one errors, like referencing elements such as `R(TOTAL_TRACKS)` (the last legal element being `R(TOTAL_TRACKS-1)`), or `R(i)` where i has been decremented to -1.

The user is allowed to call any C function from the -lm and glibc libraries. But some functions are not thread safe (which means `PARALLEL_SAFE` macro should not be used) and some others have undesired side effects. For example, should the user write inside his function something like `exit(R(0));`, Cinelerra would immediately halt. The user is left to imagine what would happen after the following expressione:

```
R_OUT = R (system ("killall -9 X"));
```

Hardware acceleration

Blend Algebra/Blend Program functions are not hardware accelerated which means that the function algorithm is not programmed as an OpenGL shader. However, functions can be parallelized using the load balancing engine of Cinelerra, and can be compiled with optimization (this is switched on by default). The default optimizing option (defined in `BlendAlgebraCompile.pl` and `BlendProgramCompile.pl`) is `-O2`. One can redefine it to `-O3` or `Ofast`. It has to be noted that using `Ofast` or `-ffast-math` options can lead to ignoring some IEEE rules for floating point math, for example some intrinsic tests on infinities or NaN which can yield unpredictable results.

Debugging

Although user's function can be compiled with debugging option such as `-g` or `-ggdb`, debugging functions can be tricky. The programmer can set a breakpoint into the function only after its code has been loaded in memory by the plugin, and set another breakpoint into plugin only after that plugin itself has been attached to a track. Moreover, a function can get detached from memory under some conditions, and then its breakpoints will be lost.

Debugging printout inside the `PROC` phase, although possible, would seem too verbose: the `PROC` phase would be called for full-HD footage 2073600 times per frame! Here perhaps one could make use of `PIX_X, PIX_Y` coordinates at some selected place like in the example:

```
if (PIX_X == 320 && PIX_Y == 200)
```

Portability

The current implementation should be portable as long as you have a working C compiler such as gcc, the default to compile CINELERRA-GG, or clang which has been known to work; but other compilers may be compatible as well. Portability to other architectures that are Unix-like and ELF based should work, but not Windows.

NOTE: For developers who want to take advantage of creating a plugin to generate an effect in the same manner as Blend Algebra/Blend Program, please reference Unique Blend plugins workflow. This methodology makes it easier for the programmer to more quickly test different combinations without spending time to restart CINELERRA-GG, recompile it, reload your project, reattach plugins, and so on.

10.10.7 Blue Banana

Blue Banana[7] is an *HSL Qualifier* (HSL= Hue, Saturation, Luminance; HSV in our plugin, with V = Value), one of the basic tools of any grading software that are based on circumscribing a zone of the frame by extracting a chromatic key and producing a *matte* in the alpha channel (Secondary Color Correction). Blue Banana differs not by creating a real matte, but by creating a *selection mask* exclusively for use within the plugin. The three H, S and V sliders are called *qualifiers*. The BlueBanana plugin has a couple of useful purposes. It can be used for color transformation or remapping – by isolating a specific color and then performing color change/correction on only that color (or color ranges). Another useful purpose is for chroma-key filtering, using multiple BlueBanana plugins on the same track. Also, it can be used in conjunction with the mask operation of the Compositor. Usage of BlueBanana may seem complicated at first, but it is necessarily so in order to get enough control to produce the desired effect simply and quickly. Just changing a single color is actually quite easy. BlueBanana is keyframable (figure 10.28).

[7] credit to Monty Montgomery programmer

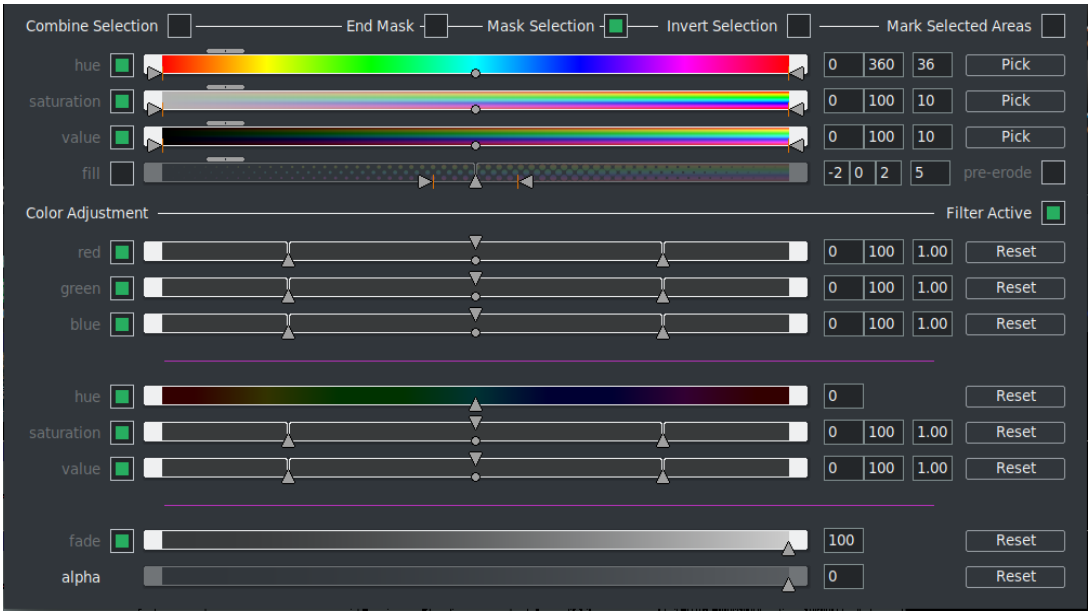


Figure 10.28: *Screencast showing the BlueBanana plugin control*

The basic strategy for BlueBanana is to:

- Select a specific target color with the eyedropper tool.
- Create a selection region by expanding color ranges around that color.
- Optionally reduce or expand the alpha plane as a regional selection mask.
- Optionally apply a color remapping or transformation to the selection.
- Optionally reset the output alpha to opaque, or pass the alpha to another BlueBanana plugin.

Just a Warning Note:

BlueBanana may use a lot of CPU and Memory because it is doing a lot of work. If you turn off the plugin on the plugin bar below the video track in the main track canvas it will stop using cpu when not in use. Or once you uncheck *Mark Selected Area*, it will no longer be using the cpu to mark the selected color area in realtime while drawing the diagonal animated pattern in the compositor window.

10.10.7.1 Example Usage[8]

If you just want to try this, follow these steps.

[8] from original message by Rebecca

First — Choose your color.

1. Load your video, add the BlueBanana plugin to the track, bring up its control window, and uncheck any checked boxes (mostly just to avoid unexpected results).
2. In the Compositor window, choose the eyedropper color picker tool on the left-hand side and click on the area of the image that shows the color you want to change/correct.
3. In the BlueBanana plugin window, to the right of hue, click Pick. And if you want to modify saturation and value, also click on the Pick button for them. To see what it does, Pick them also.
4. Next, check the Mark Selected Areas box at the top right of the BlueBanana plugin window to see the selected color-matching areas which will become marked in a diagonally striped pattern.
5. You can now manually modify your selection in the Color Selection area in the obvious ways for hue, saturation, value and fill. The arrows to each side of the small circle widen the selected area. Move the dot and you move the range. The slider on top of the horizontal color strip shifts like the amount of the strip is dedicated to that part of the color spectrum. Fill will fill more area or less area in your selected region.

Second — Adjust your color choice.

1. There are color strips under color Adjustment which will show color changes as you modify values.
2. Uncheck Mark Selected Areas and check the Filter Active box to the right of Color Adjustment.
3. As needed, you can individually check and uncheck all the various parameters using the boxes to the left of each line. Again, these are intuitive and broadly similar to the above. The arrows at the bottom widen the range, the circle at the bottom moves the range, and the top slider, which is an arrow this time, affects distribution. It provides a little histogram effect to give you an idea of what you are changing. The fade adjusts the level of color blending. The alpha is basically the opacity of your changes.

Definition of Wording/Checkboxes/Buttons/Operators are being described next. Some of the commentary was adopted from information provided by *Monty Montgomery* and from questions and answers from email by *Igor Ubuntu*, who did extensive testing.

Operational characteristics for the color-related adjusters:

left arrow slider operates the range minimum (left edge); the numerical value shows in the left-most textbox.

right arrow slider operates the range maximum (right edge); resulting numerical value is in the middle textbox.

middle circle slider below can move the current range up or down and the numerical results will show in the left and middle textbox. Move the dot and you move the range.

top pad slider operates the edge slopes (selection attack/decay) and the value will be displayed in the rightmost textbox. Sharp edges are represented by 0; 100 represents smooth edges. It is only for "Mask Selection" (Pane 1). It applies simultaneously to the left and right edges (symmetric).

top arrow affects the distribution skew. It is the equivalent of Top Pad Slider for Pane 2 ("Color Adjusting").

Left/Right Arrow determine the edges of the color range. They are precise edges independent of each other, so we can get asymmetrical ranges with respect to the center of the interval indicated by *middle circle slider below*.

Operational characteristics for Fill:

left arrow slider operates mask erosion filling. First textbox value.

center up arrow slider operates the fill skew midpoint. Second textbox value.

right arrow slider operates mask expansion filling. Third textbox value.

top pad slider operates the edge slopes. Right textbox value.

The textboxes are available so that you can directly type in numbers from the *color wheel*. This could be helpful if duplicating previous work as it would be an instantaneous exact numerical match without having to continuously fine-tune the movement of a slider.

There are two panes separated by long horizontal lines (through the middle of the screen) in the control window of the BlueBanana plugin, clearly visible in the previous screencast. The top pane is first used to create/modify a selection, and the bottom pane is used to operate a change.

Pane 1

This section is used to select the target color domain. First, a short explanation about alpha. The alpha channel used in BlueBanana is not transparency (*matte*); it is used as the *Selection mask*. Alpha plane is the alpha channel of the current image. So that:

RGBA = red/green/blue color planes, alpha data plane. YUVA = luma/Cb/Cr color values, alpha data plane.

The alpha data normally is input to the blending operations in the patchbay overlay mode. The alpha data usually creates the appearance of stacking order, and determines which color planes are visible in the rendered result. When BlueBanana is used, the meaning of

the alpha data is changed to the selection. It is useful to think of the alpha data as more solid then it is transparency in blending, and more selected when it is used in BlueBanana. In both cases, the greater the alpha value, the more the effect is expressed.

Usually, alpha is normalized to a range from 0 to 1, zero = no effect, 1 = total effect, 0.5 = partial effect. In both cases, alpha is what math people call an auxiliary variable. It is needed, but is not part of the answer. In this case, the answer is the visible rendered result. Alpha is like meta-data. Note that the "alpha" slider works with both "Filter Active" and "Mask Selection" checkboxes checked. The "Filter Active" checkbox enables the options: red, green, blue, hue, saturation, value, fade, alpha. If "Mask Selection" is not checked "alpha" will not work.

Let us now examine the instruments in **pane 1**:

Combine Selection The selection is the intersection or union of two pixel masks. Mathematically, A and B are normalized, (scaled to between 0 and 1) and used as selection mask weights.

$Intersection(\cap) = A \times B$

$Union(\cup) = A + B - A \times B$

where A is the input alpha plane as a mask, 1=selected, 0.4=partially selected, and 0=not selected; B is the color selection of trims and feathers made by varying the sliders.

The result is a new alpha plane, which will be output (if *End Mask* is not set). The $0 \ldots 1$ selection values are used to weight the color transformation filters if/when they are active and operate a change. The color adjustment filters available in Pane #2 can change red, green, blue, and remap hue, saturation, value in the pane. There is also *fade* which applies to the color channels and *alpha* which applies to the resulting alpha plane.

The basic plan is to either:

- reduce a selection area by intersection (Combine selection off) $A \times B$

- increase a selection area by union (Combine selection on) $A + B - A \times B$

Mask Selection applies the current mask to the selection, such that the mask clips/expands the selection. When mask selection is enabled, the result of the and/or will be stored to the alpha result, but when mask selection is unchecked the mask is ignored and the selection is not modified. The selection is used to weight the effect of the filtering, or to control the output alpha.

End mask only visible when *Mask Selection* is checked. End Mask causes the entire alpha plane to be set to 1. The image becomes opaque. This is usually only set in the last plugin of a stack (the stack may be just one plugin doing only color modification). In the event that a color selection mask is used with multiple, layered BlueBanana filters on the same track, the grouped BlueBanana filters may share a single mask by

all enabling Mask Selection, but with only the last BlueBanana enabling End Mask. This usage pattern gives the End Mask control its name.

End Mask as used in Color Transformation/Remapping: In many use cases where you are just remapping color, you are still interested in seeing all of the picture. If this is the case, then checking End Mask on the last BlueBanana plugin will show you the entire picture. The alpha plane may be in use as a selection mask, but it may not be wanted as part of the result.

End Mask as used in Chroma-key Filtering: In cases where the selection is for a chroma-key, you are interested in the alpha channel for blending, like *Normal* or *SrcOver*. So for this usage of the BlueBanana, don't check the End Mask.

Invert Selection reverse target color domain, which is 1 minus selection.

Mark Selected Areas when this box is checked, the chosen colors are presented in an animated diagonally striped pattern.

Hue select a hue domain; click on the Pick button to select or check the box to the left of hue or uncheck to ignore.

Saturation select a saturation domain; click on the Pick button to select or check the box to the left.

Value select a value domain; click on the Pick button to select or check the box to the left.

Fill will fill more area or less area of your selected region. This describes how it works. Fill control is an automated way of doing grow and shrink on the selected area, to fill in small holes in the middle of the selection, or to eliminate spurious pixels that are on the outer or inner edge of the selection. Be careful how much you shrink, because it can lead to edge segmentation with visible and annoying blocks of pixels. This is where Blur plugin can help. Blur should not be overdone so as not to create unsightly halos.

If none of the Hue, Saturation, or Value sliders are active – meaning that the whole frame is selected – the Fill slider will have no effect even when enabled. The word Fill will appear ghosted to indicate this.

The three lower handles in the fill slider correspond to *Shrink* (the left hand slider), *Final* (the middle slider), and *Grow* (the right hand slider). These are used in combination to alter the selection by first growing it by the amount specified by the right hand Grow slider, shrinking it to the amount specified by the left hand Shrink slider, and then growing it again to the final size specified by the middle Final slider. The top slider then feathers the resulting selection. Growing the selection it has the effect of filling small holes in the selected area. Similarly, shrinking and then growing tends to remove small flecks of unwanted selection. The Final slider specifies the overall desired shrinkage or growth of the selection when finished. To specify a pure Grow or Shrink operation, set the Final slider and the Grow/Shrink slider to the same value and leave the other slider at zero.

Pre-erode this control reverses the order of operation to Shrink, then Grow, then Final. The change is subtle on most images, but overall removes more small features because it first removes flecks before filling in holes.

Pane 2

This section is used to modify the color of your selection. Descriptive commentary for this pane.

Filter Active checkbox to indicate that the modifications will be shown.

Color Adjustment For Color Adjustment, RGB can be used as color weights while the HSV can transform color. For the following items there are three sections on the slider. The *center* section represents the nominal 0% — 100% range; the *left* section represents negative values, and the *right* section represents values greater than 100%. Values can be out-of-range within BlueBanana without clipping, but they will clip once they leave the plugin.

RGB affect the color channels individually.

- **Red** modification color; click the Reset button to revert to default. Values are reflected in numerical textboxes on the right-hand side.
- **Green** modification color; click the Reset button to revert to default. Values are reflected in numerical textboxes on the right-hand side.
- **Blue** modification color; click the Reset button to revert to default. Values are reflected in numerical textboxes on the right-hand side.

HSV reorient the color spectrum, and affect all of the color channels simultaneously.

- **Hue** a single numerical value will appear in the right-side box. Click the Reset button for default.
- **Saturation** for modifying the saturation; click the Reset button to revert to default. Values are reflected in numerical textboxes on the right-hand side.
- **Value** for modifying the value; click the Reset button to revert to default. Result is reflected in the numerical textboxes on the right-hand side.

Fade controls the entire color re-mapping, and does nothing if no color adjustment is active.

Alpha controls the output alpha (this is not available when End Mask is set); click the Reset button to revert to default. Result is reflected in the numerical textboxes on the right-hand side.

Tips

1. In Pane 1, for mask selection creation, you can also use only one or two (checked or Pick button) of the three qualifiers Hue/Saturation/Value. With Hue alone you get a precise range of hues. With Saturation you select those saturation values for all hues, and with Value you do a "Luma Key," that is, you isolate only certain gray values. It may be useful to start with only one qualifier, and then gradually activate the others; but there are important cases that require all three bars. For example, skin tones, which are always characterized by a mixture of shadows, highlights, and different levels of saturation.

2. Hue and Saturation Qualifiers can be inaccurate if the file is highly compressed and with Subsampling type 4 : 2 : 0. There are indeed blocks and macroblocks of pixels, visible especially along the edges, that disturb the creation of the selection mask. Better to use high quality files at least 4 : 2 : 2, better still 4 : 4 : 4. The brightness Y channel, on the other hand, is always at the highest quality (4). The presence of color noise can also be a problem because it creates spurious pixels. A tip may be to work on and enhance the clip before applying Blue Banana.

3. If there are unwanted (spurious) selections in the frame that are small and far from the main selection, they can be eliminated or minimized with some Blur. Larger spurious selections can be eliminated by masking them (with the `Mask selection` option enabled in Blue Banana and with the `Apply mask before plugins` option in Mask tool). This is an action analogous to *garbage matte* in Chroma Key. If there are many spurious areas, perhaps with complex motions, it is best to mask only the selection we are interested in and then bring the `Fade` slider to −100 to reverse the mask.

4. To select multiple colors in the same clip we can use multiple instances of Blue Banana.

5. Once a satisfactory selection mask has been created, scroll through the entire clip to see the presence of artifacts, defects, or spurious areas in the other frames.

6. It is known that primary Color Correction precedes secondary CC. However, if we use primary CC tools that cause the highlights and deep blacks to clip, for example the histogram, we will get clipped areas that then, in secondary CC, cannot be recovered. We can then first do a secondary CC for the areas near the white point and the black point using Blue Banana which works at 32bit Float. For example, we can turn down the highlights so as to reveal details and the same can be done in the shadows. Once we have worked in these details without causing clipping (that is, reported within the standard range) we can switch to primary CC.

Let's see two examples of HowTo:

BlueBanana Use Case #1: (Color Transform/Remapping)

- Load a video track, and add the BlueBanana plugin to your video. The alpha channel is usually all opaque. This serves as an initial full screen selection mask.
- Open the controls, and start with all boxes unchecked. Now reduce the selection using the top pane in intersection mode (that is *Combine Selection* is unchecked) to begin the effect.
- Use the *eyedropper* on the compositor window to choose a particular color.
- Click on the 3 plugin Pick boxes on the right side of each line of HSV to get the color selection.
- Check *Mark Selected Area*. The affected zones will be identified on the composer.
- Adjust the selection using the HSV and Fill sliders of the top pane. The selection mark will be updated as you operate the controls. The composer mask striping will be strongest as the mask is nearer full selection.
- Now uncheck *Mask selected area* & check *Filter Active* to begin Color Adjustment.
- Enable any needed colorspace modifiers, RGB / HSV sliders, and setup the color changes by moving the sliders. The current output may be the desired output.
- Enable *Mask Selection* and the alpha output will pass the selection mask to the image alpha channel. This can be used as a very flexible chroma-key filter. It also allows more plugins to be stacked and more selection information to be added, either by intersections or unions with other selections.
- *End Mask* simply sets the output image alpha to opaque. This is normally used to end a stack of BlueBanana plugins, and render the entire image with a complex selection.

BlueBanana Use Case #2:

This case uses stacked BlueBanana plugins working like *chroma-key* filters. It assumes you have already learned how to operate the plugin.

- Bring up 2 tracks of video media – one for foreground and one for background.
- Add 2 BlueBanana plugins on the first track. Turn off all checkboxes in both plugins.
- On the top plugin, use the top pane to create a selection mask, using *Mark Selected Areas*.
- Turn off top plugin *Mark Selected Areas*, and disable the top plugin via the plugin title bar on/off.
- Create another selection using the second plugin's mask, using *Mark Selected Areas*.

- Turn on the top plugin. Make sure both plugins *Mark Selected Areas* is off.
- Check *Mask Selection* and *Filter Active* in both.
- Check *Combine Selection* on second BlueBanana to see the final results.

You will see that there is intersection of the full plane with the first chosen regions, so the $alpha = 0$ everywhere but the area you picked and you see through. And $alpha = 1$, where the intersection selection was 1. The Normal blend shows you the track on top in these regions (the foreground track where $alpha = 1$).

If you are building an alpha selection mask by first intersection and then union, the top BlueBanana should not change the colors or the bottom plugin will need to target the remapped colors since that is the input to the lower BlueBanana.

10.10.8 Blur

This is a Gaussian type blur. Other blur plugins – *Linear*, *Motion*, *Radial*, and *Zoom* –are described later. This plugin is keyframable. If the original source has values outside the range $(0-1.0f)$, the plugin will always produce clipping. Blur is used to blur a video track via the following parameters:

Horizontal and vertical values are used to tell which one of the fields blurring affects; can be both.

Radius use this dial to define the amount of blur to apply.

Alpha determines radius use alpha to define the amount of blur to apply. (radius=gray value of alpha)

Blur alpha, red, green, blue specifies which color channels is to be blurred.

If you are getting confusing results with old projects it may be a result of the discontinued parameter *alpha determines radius*. For compatibility reasons and for possible future development it has been left in the code but hidden in the plugin's GUI. The parameter for *alpha determines radius* is `A_KEY`, which can be 0 or 1. When you press the `Reset` button in the Blur plugin window `A_KEY=0`.

Old projects may have saved that parameter (A_KEY) to 1 instead of 0 and that may present a problem so you should do one of the following workarounds to change it to 0. Then be sure to save your project with these changes applied.

1. Click on the *Reset* button in the dialogue/controls popup box and re-enter any of your other parameter desired values.

2. OR in the CINELERRA-GG program, open the project. Click on the cog icon (*Preset edit*) of the Blur effect bar and the *Keyframe parameters* window is open. There, you can see the A_KEY parameter and change it: select the `A_KEY` parameter and in the `Edit value` box, change it from 1 to 0, then press the `OK` button.

10.10.9 BoxBlur

Based on ffmpeg's boxblur, this is a very fast algorithm which can be used to blur horizontal, vertical, and at a power level. Simplest usage is to just blur the entire image but with the following parameters, you can create a specific rectangular section to blur instead.

X/Y: point coordinates of a rectangular box to be blurred. X and Y of 0 are in the upper left hand corner. Default to be blurred is the entire image.

Drag: is used when you want to create just a rectangular area to be blurred rather than the enire image. If you used keyframes to cover a moving area and had left the Drag box enabled, you can remove the drag box with *Allow keyframe spanning* whose use is described in Allow Keyframe Spanning.

W/H the values in these 2 boxes specify the width and height of the drag area box measured in pixels as shown in the compositor window. You can set these manually and if you can not see the location of your box, set them to zero because 0 sets it to the same as the width/height of the project.

Horiz: slider goes from 0 *to* 100. Increasing this *horizontal* number, increases the blurriness in the horizontal direction. The default value is 2. You can also use the mouse wheel to scroll the slider. There is a clear button on the right to set the value to 0.

Vert: slider goes from 0 *to* 100. Increasing this *vertical* number, increases the blurriness in the vertical direction. The default value is 2. You can also use the mouse wheel to scroll the slider. There is a clear button on the right to set the value to 0.

Power: the way *Power* works is like stacking up the blur multiple times. For example, a power of 3 would be like blurring once, then blurring that again, and finally blurring that a third time. The slider varies from 1 to 10 with 2 being the default value. There is a clear button on the right to set the value to 1.

Reset: button to revert to the default values and turn off Drag if it is on.

10.10.10 Brightness/Contrast

To brighten a dark shot, or add light, use this plugin. Do not overuse the effect or you risk degrading your video quality. The *Brightness* slider moves up or down the values of the entire channel and corresponds to the *Master Offset* of the various grading programs. The *Contrast* slider expands or narrows the brightness values of the entire channel; corresponds to the use of the *cursors* (small triangles) in the *Histogram* plugin. For maximum precision you can use the input box. *Clear* icons are present to reset its slider to default without

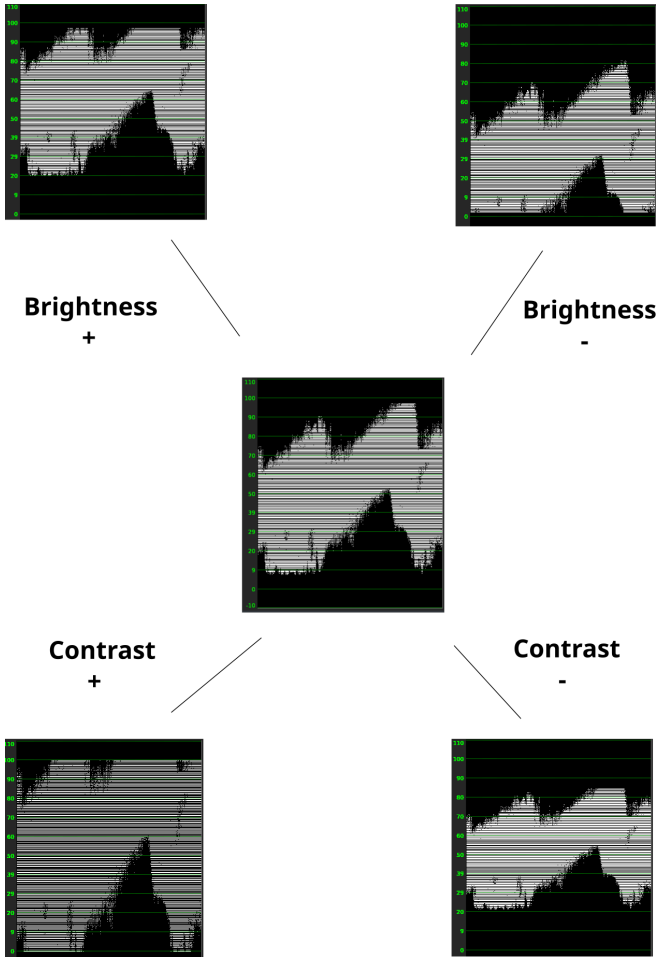


Figure 10.29: *How it works Brightness and Contrast*

affecting others. Use the effect along with keyframing to brighten a long shot that is dark at the beginning but bright at the end. Generally you will want to change the brightness and contrast about the same amount (for example – brightness 28, contrast 26) so that your original colors are kept intact. This effect is also keyframable (figure 10.29).

10.10.11 BurningTV

Makes your video burn where there are small light colored patches of video. This came from `https://effectv.com`.

10.10.12 C41

The C41[9] plugin takes a 16 $bit C41$ digital intermediate negative film as input and outputs a positive image. It became necessary because $C-41$ negatives can fade or color-shift over time which was a problem early on. It is still important today because there is a large

[9] credit Florent Delannoy, original program code author, and Edouard Chalaron

amount of documentaries, video clips, and other media out there that was shot on super 16 film. This works for RGB-float, RGB, and also YUV variations.

There are two sets of data – the scanned input values and your corrected values. Simple functionality of the plugin is to compute the data, transform to get corrected values, then apply that.

Basic usage strategy:

1. first time the controls come up, nothing is checked and everything is set to 0
2. check the box *Compute negfix values* to see the current media input values
3. check *Activate processing* and you see a $1-colored$ screen in the Compositor due to zero values
4. check the *Apply values box* to see the input values on the left side propagate to the right side
5. check *Apply default box* if you want to make sure that the borders of the image are not used
6. correct the output values as desired on the applied right side

It is important to note as you play or change the frame, the plugin re-computes the data as you move along, but it is not propagated to the applied side.

Checkboxes:

Activate processing when checked, the c41 operation is used to render the image.

Compute negfix values computes the current negative values of the image (inside the box).

Show active area draws horizontal and vertical grid lines displaying the boxed area.

Postprocess when checked, applies contrast/brightness values as defined in $coef\frac{1}{2}$.

Values:

Compute negfix values (left side) and

negfix values to apply (right side):

Min/Max R/G/B minimum and maximum values for Red, Green, and Blue.

Light value of light; a smaller number is lighter.

Gamma G/B values for gamma Green and Blue.

Contrast simple color contrast.

Brightness white brightness.

Buttons:

Apply values copies computed RGB/Light/Gamma/Contrast/Bright from negfix to applied values.

Apply default box copies default computed Box column/row from negfix to applied values.

Shading box: The boxing option allows for calculating the inversion of the digital negatives in a given area of the frame as opposed to the entire frame. The program will automatically calculate the columns and rows to shave from the frame when compute negfix values is checked. A default box area is initially calculated, called the shaving box, based on where the min/max difference in a row/column is less than the program defined tolerance. This row/column minimum and maximum difference must be greater than 0.05. The effect is to cut away the border areas with constant color. If you check the Show active area, you can see the box in the compositor window. The boundary search is constrained to a range of 0.1 to 0.9 times the frame dimensions, to create a 10 percent shaved margin to avoid over-scan and negative edge bleeding. Manual adjustment of the shaving box is controlled via the four sliders on the bottom right which move each of the left, right, top and bottom shaving margins. The slider bar new values automatically take effect as you move the box and you will see the right-hand side applied values change. When you have either the rows or the columns where the minimum slider is greater than or equal to the maximum slider, the default box will be in effect instead.

Optional postprocessing: in order to have the values of Contrast and Brightness take effect, you must check the Postprocess checkbox.

Contrast is the difference in brightness between objects or regions.

Brightness refers to the overall lightness or darkness of the image.

Figure 10.30 shows the C41 controls on the left and part of the Compositor window with grid lines showing the default shading box since the Show active area box is checked. Changes have been made to the left-hand side original computed values as seen in the right-hand side such as Gamma G which contains the hairline cursor and has a partial red outline value box.

Figure 10.30: *C41 - Control window and compositor window in action*

10.10.13 Chroma Key

This effect erases pixels which match the selected color. They are replaced with black if there is no alpha channel and transparency if there is an alpha channel. In this case, you create a matte in the alpha channel, which is not visible to us. The selection of color model is important to determine the behavior (figure 10.31).

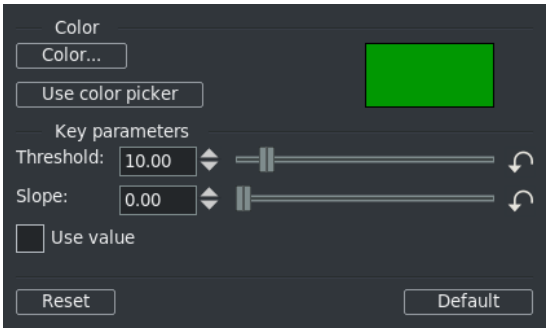


Figure 10.31: *Chroma Key control window*

Chroma key uses either the lightness or the hue to determine what is erased. *Use value* singles out only the lightness to determine transparency (Luma Key).

Color section

Select a key color to erase using the *Color* button. Alternatively a color can be picked directly from the output frame by first using the *color picker* in the compositor window and

then selecting the *Use color picker* button. This sets the chroma key color to the current color picker color.

Be aware that the output of the chroma key is fed back to the compositor, so selecting a color again from the compositor will use the output of the chroma key effect. The chroma key should be disabled when selecting colors with the color picker.

Key parameters section

If the lightness or hue is within a certain *threshold* it is erased. Increasing the threshold determines the range of colors to be erased. It is not a simple on/off switch; it's a range color. As the color approaches the edge of the threshold, it gradually gets erased if the *slope* is high or is rapidly erased if the slope is low. The slope as defined here is the number of extra values flanking the threshold required to go from opaque to transparent. The *Default* button returns Threshold to the value 10.0; Slope to the value 0.0; and Color to black (#ff000000). The *Reset* button and the *reset* icons next to the sliders always return to the value 0.0. Before the sliders are the *input boxes* so that the precise numerical values can be written.

Normally threshold is very low when using a high slope. The two parameters tend to be exclusive because slope fills in extra threshold. The slope tries to soften the edges of the chroma key but it does not work well for compressed sources. A popular softening technique is to use a maximum slope and chain a blur effect below the chroma key effect to blur just the alpha.

10.10.14 Chroma Key (Avid)

Chroma Key (Avid)[10] (figure 10.32) replaces a color with another color or transparency using HSV variables. It is frequently used to remove a color from a video to composite with another image. This process is generally referred to as green screen or blue screen process because of the background color that is keyed out. More information is at: `http://en.wikipedia.org/wiki/Chromakey`.

Requirements

The subject in the timeline video should have a good background. The lighting is crucial and good lighting during production will save you time with much less effort than in post-production. Another tip is to use a low-compressed, intraframe codec with as high a color depth as possible. In case of YUV-type source, it is better to have subsampling 4 : 4 : 4 or 4 : 2 : 2. Here we assume that we have a good video, filmed on green or blue screen that we want to use. Important: make sure you are using a color model that has an alpha channel, such as *RGBA8*, *RGBAFloat*, *YUVA8*. To change color model, go to `Settings → Format → Color Model`. Any manipulations with alpha masks in CINELERRA-GG

[10] Credit for Plugin by Jerome Cornet; credit to original creator for major upgrade and credit to SGE for adaption and improvements. Avid is a trademark of Avid Technology, Inc.

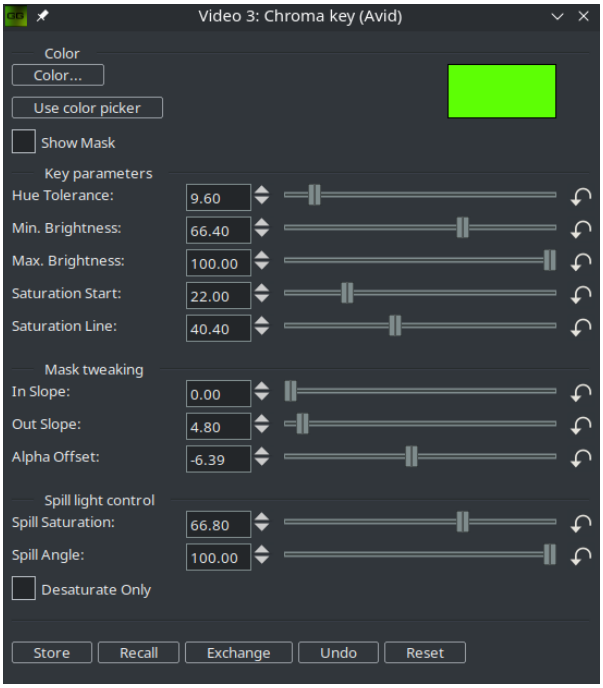


Figure 10.32: *Windows config for Chroma Key (Avid/HSV)*

require a bottom track with some opaque background, otherwise the transparent holes are displayed as if the same track without transparency is located under it.

Usage

As in any other effect, drag it to the timeline in the main window. You can tweak each parameter in order to improve the keying.

To understand how the plugin parameters work, we need to refer to the HSV color wheel (foreground color) in figure 10.33. You may want to first apply the Color Swatch plugin which is very helpful in determining which variables should be modified. Consider that this is an abstract example; in the real world we will be dealing with much more complicated masks.

The left half has Saturation from 0 in the center to 100 on the edges. An example of a green screen is shown here. The right half has brightness from 0 (center) to 100 on the edges. The example of a blue screen is given here. The key color is the radius of the wheel, and its angle determines the hue. The Hue Tolerance is the arc of a circle (a wedge) that includes the radius. The wedge is the color range (mask) that we want to eliminate by keying, its shape is precise and easy to understand. The Brightness are the inner/outer extremes of the wedge; if the inner value (Min) is 0, then it coincides with the center point, if the outer value (max) is 100 then it coincides with the arc on the circumference. With intermediate values we will have masks similar to the one shown in the figure. Min Saturation is the distance from the center, along the Hue radius, from which we want to impose saturation

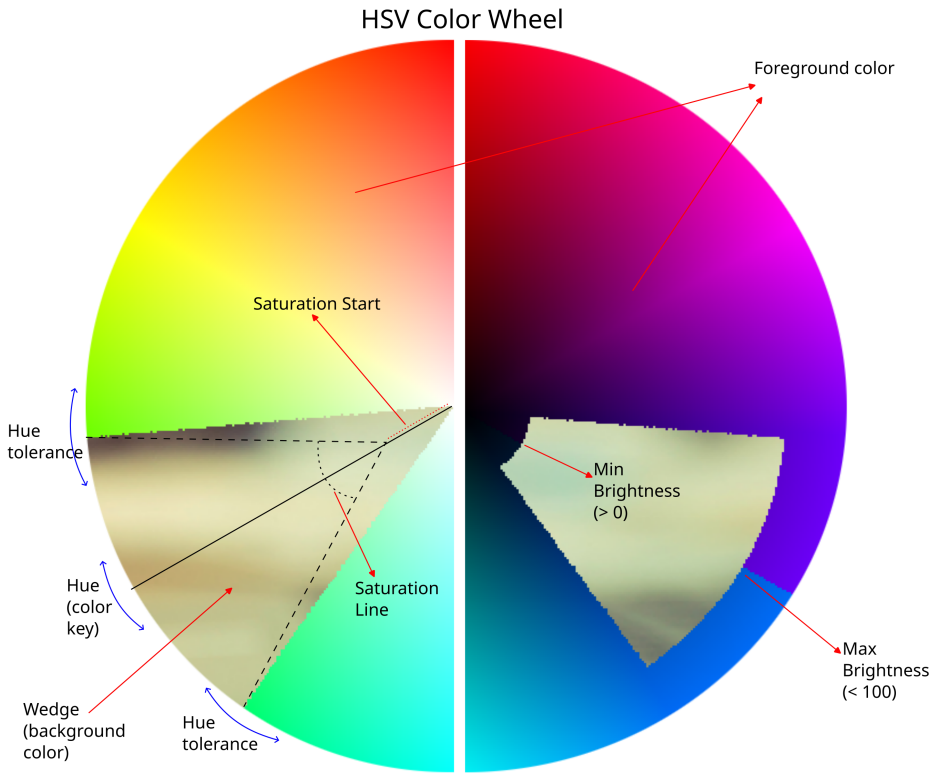


Figure 10.33: *Various parameters in the 'Color' and 'Key Parameters' sections.*

with the value 0. It will be the value from which keying starts. Saturation Offset is an additional cut we make to the inner part of the wedge from the Min Saturation value. All of these adjustments allow us to establish a range of colors (Mask) that perfectly matches the background (the wedge in the example in the figure 10.33) that we want to eliminate. This is a precision operation, and it is not rare to return to the parameter combination several times to refine the mask.

Color section

In this section of the configuration window, you choose the key color and have the ability to see the mask created.

Color... Select the key color (green, blue, or whatever) using the color wheel or the color picker. Only the Hue matters, not Saturation or Value. To use the color picker, click on the *color picker* icon in the Compositor window, then click on the color you want in the Compositor window. Finally in the Chromakey (Avid) parameters window, click on *Use Color Picker*. You will need to disable ChromaKey (Avid) plugin when using the Color Picker on the Compositor window.

Show Mask The plugin does not create a true matte in the alpha channel, but creates a

mask. Activating this option will show the foreground as a white (opaque) shape while the background in transparency. You will be able to control the extent of the mask, any jagged edges, small white areas within the mask or small transparency areas within the foreground, which are to be removed.

Key parameters section

In this section we expand the mask to a range of colors close to color key and refine the selection by also taking advantage of brightness and saturation.

Hue Tolerance: Because there are slight variations in lighting in real cases, the background will not be in a uniform key color hue. Increase or decrease the Hue tolerance to mask out the background. In determining the hue range, it is useful to use the Color Swatch plugin in *constant brightness* mode and with *Draw Source* active, which allows us to see the pixels of the figure, making us choose which ones to let into the mask (keying out) and which ones not. If there are dark spots that are keyed out that shouldn't be, it can be corrected later (With the `Mask` tool; this operation is called the *Garbage Matte*).

Brightness: It allows the color range (Mask) to be better defined by exploiting the differences in brightness between background and foreground. This could be referred to as a Luma Key within the Chroma Key. Increase *Min Brightness* so that only the background is masked out, and not parts of the foreground. You can also reduce *Max Brightness* if some clear areas are keyed out (useful for very dark backgrounds).

Saturation: Saturation allows the color range (Mask) to be better defined by exploiting the differences in saturation between background and foreground. Since the best results are obtained by keying pure colors, it may come in handy to eliminate the less saturated colors (proper to the foreground) while leaving the more saturated colors (proper to the background, i.e. Green/Blue Screen) to the keying action. Increase *Saturation Start* so that only the background is masked out, and not parts of the foreground. *Saturation Line* can be used to change this, because it acts similarly to Min Brightness. It could be said that Saturation Start concerns only the key color (wedge's vertex), while Saturation Line concerns the range of Hue tolerance. But it is best to start by leaving it at 0.

Check what it looks like at this stage, your mask should be pretty clean. Toggle *Show Mask* to check what it looks like, it should be OK. If not, repeat steps in the Key parameters section to get a better key. The rest of the controls are useful to smear the mask to help compositing later on. They will help you to make your key look much cleaner.

Mask tweaking section

In this section you intervene on the already created mask with edge smoothing and transparencies.

Slope: For now, the mask is a full on/full off mask that can be really harsh on the edges and not necessarily what you are looking for. *In Slope* and *Out Slope* will help you to smooth that key. In Slope acts on the inner side of the edges by grading them with keying (background) colors. It leaves more colors in the mask. Out Slope acts on the outer side of the edges by grading them with the foreground colors. It takes more colors out of the mask. Basically, the colors that are borderline in the mask will see their alpha channel reduced by half instead of being completely on or off.

Alpha Offset: This control offsets the whole alpha channel by some amount, from -100=full opacity to +100=full transparency. Be sure to know what you are doing if you change it from the default value of 0.

Spill light control section

In this section we try to make halos, reflections and parasitic lights present on the foreground less noticeable via spill suppression.

Spill saturation: Similar to Saturation Start, indicates the starting point from which spill suppression begins. It works if Saturation Start is > 0 because it acts only from that value, with the effect of retracting the starting point moving away from the Saturation Start value toward the center of the color wheel (white). This has the consequence of creating a small area beyond the edge of the wedge where spill suppression acts. Generally you set it with a small random value > 0 and then return to it after setting Spill Angle.

Spill Angle: Spill Angle is the main parameter of spill suppression because it causes a very noticeable effect. Basically, it is an area that extends beyond the edge of the mask, increasing its angle and thus its size. In this area (which is a gradient) pixels of the reflections of the key color (green, for example) are mixed with pixels of the adjacent color (cyan on one side and yellow on the other, because we always refer to the HSV color wheel). The suppression effect replaces green pixels with adjacent ones, taking into account the gradient. This parameter, in addition to the spill saturation, softens the edges of the mask without resorting to desaturation to gray or white of the green pixels, which create a more visible detachment. It should not be overdone so as not to compromise the color balance of the entire figure.

Desature Only: It takes the parameters of Spill Saturation and Spill Angle but instead of copying the neighboring colors over the green pixels, it applies a simple desaturation of the green pixels that become gray and then totally white. The effect is more noticeable and noisy because it creates a halo around the figure.

In the lower part we find various buttons:

Store: Stores the complete current parameter set in memory.

Recall: Sets all the parameters to the values, memorized previously by Store.

Exchange: Swaps current values and Store'd values of the parameters.

Undo: Restores all the parameters to the undo'ed values.

Reset: Reset to default values.

All buttons work globally on the whole parameter set. Each time the ChromaKey dialog is opened, the *Store* values are cleared and reset to default. Therefore, if you press *Recall* having not pressed Store beforehand, it will do the same as *Reset*. Each time the dialog is closed, the Store values are forgotten (reset to defaults). As long as the dialog remains opened, Store values remain intact, even if the current timeline position changes. The operations on distinct parameters (turning sliders etc.) do not update the *Undo* values. The following operations update values for subsequent Undo: Global Recall, Exchange, Reset buttons (but not the buttons which reset individual parameters), opening the dialog, and moving current position in the timeline.

Now the mask is probably still very harsh, so just below the Chromakey (Avid) plugin, add a *Blur* effect, and select only the *Alpha channel*, with a radius of 2 or 3 (more if you really want to soften the edges). This will significantly help the keying.

10.10.15 Chroma Key (HSV)

Chroma Key (HSV)[11] (figure 10.34) replaces a color with another color or transparency using HSV variables. This plugin is replaced by the better Chroma Key (Avid) and is maintained only for reasons of compatibility with old projects. Refer to Chroma Key (Avid) for theory and parameter descriptions, which are quite similar, except the description of spill suppression, which differs considerably.

Spill light control section

In this section we try to make halos, reflections and parasitic lights present on the foreground less noticeable.

Spill light control: This step helps you remove the green or blue halo around the edges of the mask. It does so by removing the saturation of pixels of the foreground that have a similar hue to the key color, turning them into gray instead of green or blue. The hues to be desaturated are set with *Spill Threshold*, the degree of desaturation is set with *Spill Compensation*. In order for Spill compensation to work, Spill Threshold must be > 0. If you start with Spill Compensation at 100%, slowly increase the *Spill Threshold* until the remaining green or blue areas turn gray. Then reduce Spill Compensation until the image looks good.

[11] Credit for Plugin by Jerome Cornet http://jcornet.free.fr/linux/chromakey.html

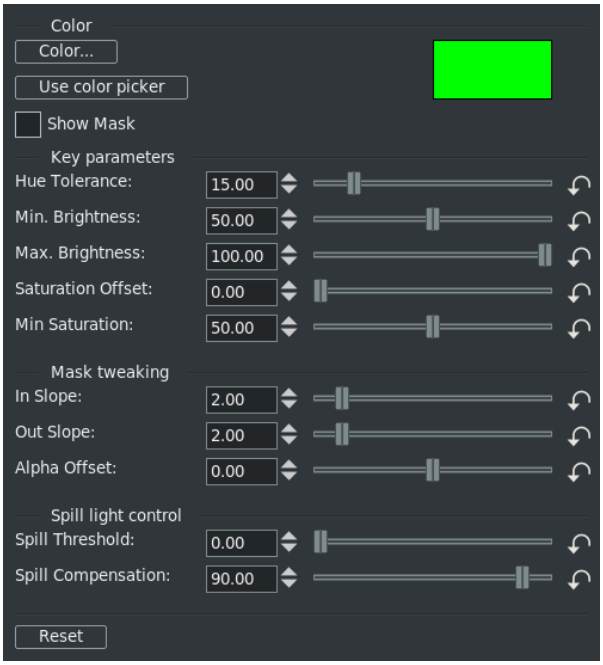


Figure 10.34: *Windows config for Chroma Key (HSV)*

10.10.16 Color 3 Way

Together with *Histogram Bezier / Curves* Color 3 Way is the main tool of Color Grading because you can modify the colors of *Shadows*, *Midtones*, and *Highlights* as desired. Color 3 Way is keyframable (figure 10.35).

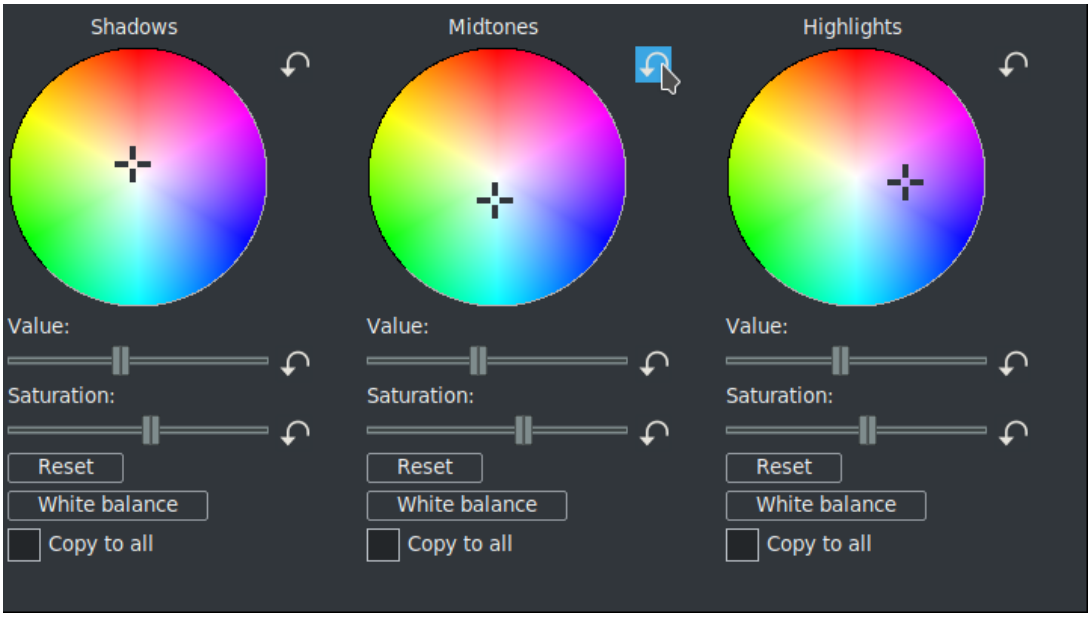


Figure 10.35: *Color 3 Way control window*

- It allows you to vary the *contrast* of the image using the slider Value, always acting separately on shadows, midtones, and highlights and thus resulting in very precise application.
- Allows you to automate the *white balance* by simply choosing a neutral color in the output of the Compositing window using the Color Picker and pressing the corresponding button in the plugin.
- Allows you to vary the *Saturation* with sliders in the same manner as contrast was varied by the Value slider. For istance, to decrease the incidence of color dominants present in the shadows or in the highlights, vary the Saturation.
- With the *color wheels* you can make very sophisticated adjustments to the shades of the images, in each of the three main areas of shadows, midtones and highlights.
- Allows you to copy exactly the setting of one zone to the other two zones using *Copy to all* button. With this option, it is like using a fourth wheel called *OffSet* in other programs.
- In addition to the three reset buttons, each slider and each wheel has its own Clear button, to return it to the default value without affecting the others.

This plugin allows maximum control over the result and maximum precision of adjustments when used simultaneously with the control monitors, i.e. *Waveform*, *RGB Parade* and *Vectorscope*. It is important to keep in mind that the three zones are not clearly separated, but slightly overlapping. This results in less precision but looks better for more smooth shades. By varying the values on the color wheels all RGB channels are affected simultaneously, which can result in unwanted color dominance. Saturation is also affected and must therefore be monitored. To use more precisely, drag the *crosshair* with the mouse in the desired area and then adjust in steps of 0.001 using the up/down and right/left arrows on the keyboard. The most common use cases (but can be adapted to virtually any situation) of the plugin are:

- White balancing.
- Expand/compress contrast.
- Mitigate under and over exposure.
- Balance colors, i.e. eliminate color dominance.
- Color matching Shot to Shot.
- Create a Stylized look.

When using the X11 graphics driver and RGBA-FLOAT color model, this plugin allows for greater than (0 - 1.0f) values (*illegal* values). This does not work when using X11-OpenGL because it is an 8-bit limited driver. Illegal values will only work for the color wheel, while the Value and Saturation sliders are always clipped to 1.0f.

10.10.17 Color Balance

Video Color Balance is a great effect to use along with Brightness/Contrast and Hue/saturation to try to compensate for possible errors in filming (low lighting, for example). It can do so much without greatly lowering the quality of the video. With it you can change the colors being sent to output *CMY* (Cyan, Magenta, Yellow) or *RGB* (Red, Green, Blue). Color Balance is also keyframable.

Since *complementary colors* are neutralized, to eliminate a *color cast*, the pertinent slider is moved in the direction of the complementary color. Clear buttons are present to reset its slider to default without affecting others. If you *Lock parameters* you get the same *Color Offset*, that is the fourth color wheel in the grading programs. The parameters of the plugin are:

CMY/RGB sliders: allows you to adjust the colors.

Preserve Luminosity: Adjusts colors while keeping the overall brightness constant.

Lock Parameters: works as a Color Offset.

White Balance used in conjunction with the *color picker* on a neutral color of the output, it will automatically balance the white.

10.10.18 ColorSpace

This plugin is a tool that can be used to convert your input media, such as a recording from your camera, from one color space/range to another. It works for both RGB and YUV as set by your project format. Options are BT601, BT709, or BT2020 for Color Space input and output and JPEG or MPEG for Color Range input and output. The Inverse option checkbox is available in case your media was rendered in the wrong color space or range so that you can fix it. By its nature, this plugin will always cause clipping on values above 1.0f.

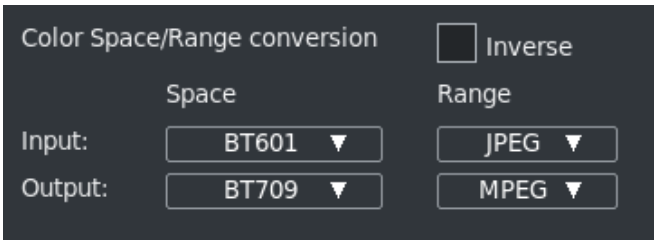


Figure 10.36: *ColorSpace control window*

Algorithm for conversion – where equations is a 3 × 3 matrix multiply

$$output = (input - input_zero) \times equations + output_zero$$

Algorithm for inverse – where equations is a 3 × 3 matrix multiply

Description: invert(equations); swap(input,output); swap(input_zero,output_zero)

$$input = (output - output_zero) \times inverse_equations + input_zero$$

10.10.19 Color Swatch

Color Swatch is mainly used together with the two chroma key plugins. It should be applied first so that its output becomes the input for the chroma key.

Constant Brightness: creates the HSV color wheel in which the brightness is constant and you can only move the Brightness slider.

Costant Saturation: creates the HSV color wheel in which saturation is constant and you can only move the Saturation slider.

Angle: turns the HSV color wheel by a certain angle.

Draw Source: is analogous to vectorscope; shows the pixels of the upper track (figure + background = source) superimposed on the HSV color wheel. It basically displays the source colors that are also present in the color wheel, but not the others. You can easily see which colors fall within the wedge (mask) and are therefore extracted, while those outside make up the unextracted figure. This makes it easier to adjust the key color and the hue range according to our needs.

10.10.20 CriKey

The Chroma Interpolation Key plugin, CriKey, is a regionally based chroma key with interpolation (figure 10.37). This is useful when you only want 1 or some specific zones to be defined by the chroma key as opposed to the entire image. Its most significant feature is that you can select several regions of interests and of different colors as opposed to only 1.

To start, if not already checked, turn on drag. In the composer window select an area of a certain color by clicking on that point with the *right mouse button* and check to see that it is enabled with an ∗ in the "E" field. The color of the area is used to define the region of interest and then you can use the *threshold* slider to designate the tolerance variation. This creates a region that is the chroma key selection and a fill will be performed in that area, but only within that region. So, say for example, a red colored area was chosen, only the red color inside the region is selected – not that color red in the entire image. The drag capability makes it easy to check a point before right clicking it to see the effect. You will want to turn off drag when you are finished with CriKey so that it does not interfere with other compositor functions.

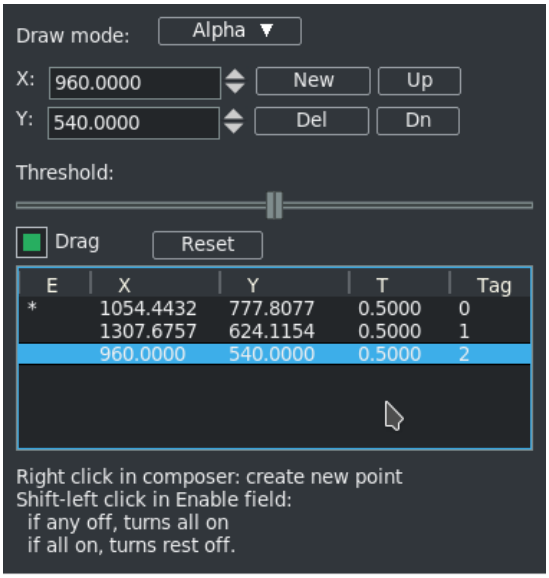


Figure 10.37: *three active point created in CriKey*

Draw mode: options let you use *Alpha* for see-thru, *Edge* to just outline the edges of the region, or *Mask* to block. The pixels which match the selected color are replaced by black if Mask is chosen or see-thru/transparent if Alpha.

X, Y: points coordinate.

Buttons: *New* to create a new point, *Up/Dn* to move highlighted point up or down *Del* to delete the highlighted point.

Threshold: slider goes from $0\,to\,1$. Increasing the threshold, increases the area to be filled or masked. You can also use the mouse wheel to scroll the slider.

Drag: for ease of use. If you need to clear the Drag enabled, you can easily do this with *Allow keyframe spanning* whose use is described in Allow Keyframe Spanning.

Reset: button to revert to only the default middle point with all others being deleted.

ListBox: "E" for Enabled with $*$ marking that; "X" is the point's x coordinate; "Y" is the point's y coordinate; "T" is the threshold value of X, Y point; *Tag* represents the # of the selected point.

Hints: for usage shortcuts.

Some notable caveats

1. When choosing an area that has variations of the same color within a region, for less work and for the best results, choose an average color in that region instead of an extreme end of that color.

2. If the threshold is set appropriately you can see the edges which is helpful.

3. The mask is computed and shows the fill region.

4. Use the Gradient plugin to substitute a different color for the selected area.

Figure 10.38 and figure 10.39 shows how moving the Threshold slider with the Point selected blacks out the single region which has the darker brown hills in it. Because the edge was located, any of the same color in the rest of the video would not be blacked out.

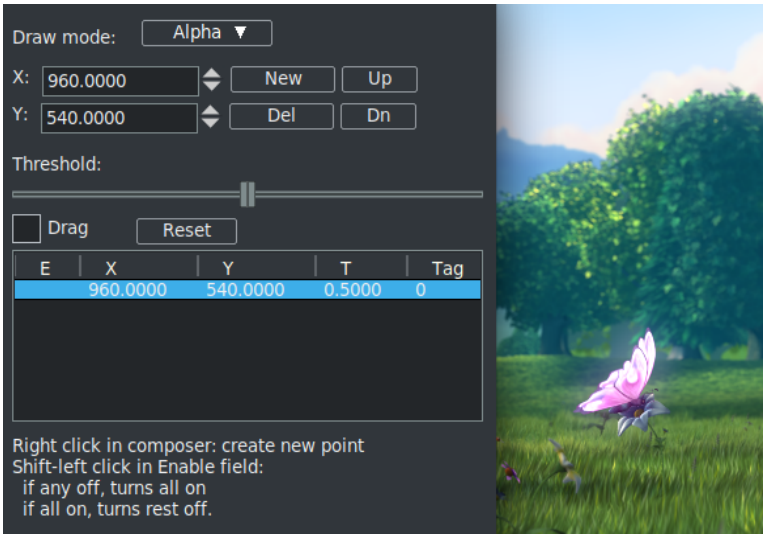


Figure 10.38: *The screenshot shows the compositor with some default settings in the controls window.*

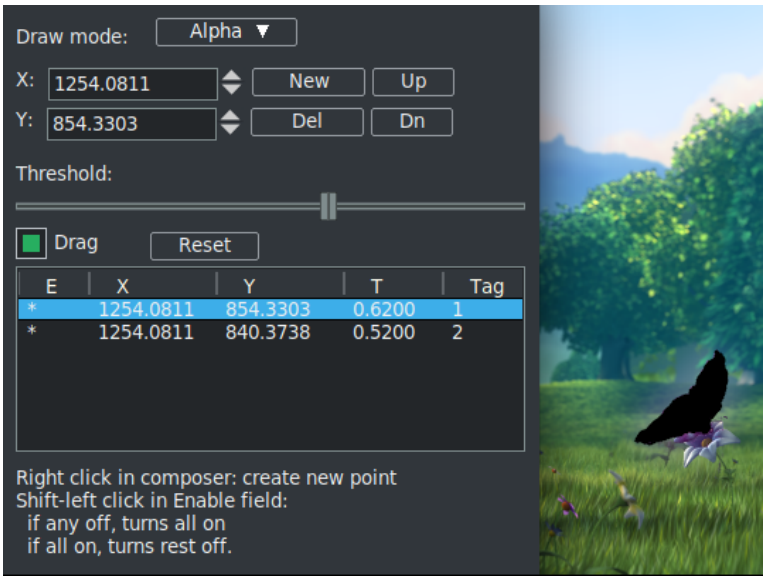


Figure 10.39: *same screenshot with moving Threshold*

Usage steps

1. Click *Reset* (there will be a single X, Y coordinate point that is in the middle and not enabled)
2. Check to make sure *Drag* is on.
3. In the Compositor, right click on area of interest and an X, Y coordinate will appear in the listbox.
4. Click on the *E* Enabled field next to this latest point and an $*$ asterisk will show.
5. Now you will see an area turn black so use the *Threshold* slider to only black out the area of interest.
6. Repeat steps $3-5$ until you have selected all of the desired areas.
7. Finally, turn off *drag* so as not to interfere with other compositor functions

10.10.21 Crop & Position

It allows you to obtain a rectangle from the frame, whose dimensions are fully adjustable by four sliders for the four sides of the frame. You can also place this rectangle in the canvas using two other sliders for right/left and up/down scrolling. With the Clear buttons we can bring the slider to default values without affecting the other parameters. Unlike the *Crop* tool, the original frame size is not altered and being keyframable allows a wide variety of uses. In figure 10.40 the Crop & Position plugin is compared with the *Crop* tool.

Figure 10.40: *Crop tool and Crop & Position plugin compared*

10.10.22 DeScratch

The descratch video plugin can be used to remove vertical scratches from film. It can also be used, after image rotation, to remove horizontal noise lines that may appear on analog *VHS* captures. For best results *YUV* should be the video format; however if your format is *RGB*, it will first be converted to YUV. There are many tuneable parameters necessary to get good results for your specific film.

Figure 10.41 shows a list of the parameter descriptions:

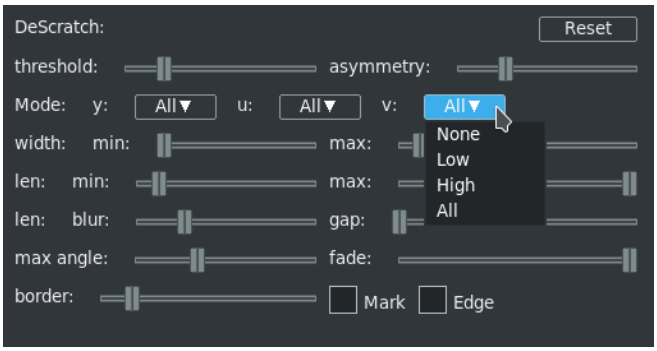


Figure 10.41: *DeScratch control window*

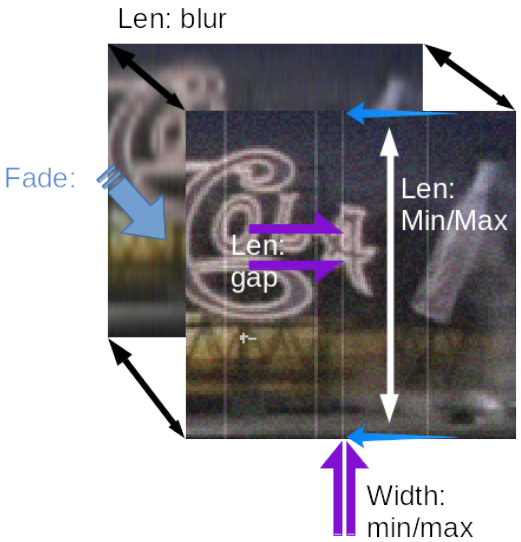


Figure 10.42: *Various parameters of DeScratch*

threshold instantaneous slope value; chroma difference in numerical pixels.

asymmetry maximum asymmetry of surrounding pixels.

Mode *None*; *Low*=black; *High*=white; *All*=both; *y* – processing mode for *luma* plane; *u*– processing mode for *chroma u* plane; *v* – processing mode for *chroma v* plane.

width min/max minimal scratch width in pixels and maximum scratch width in pixels.

len min/max percent minimal scratch length and percent maximum scratch length.

len blur scaled radius of vertical blur for frame analysis.

len gap number of pixels for maximum vertical gap to be closed.

max angle maximal angle to vertical in degrees.

fade percent of how much it fades to and how much it. Uses between before image and blurry image.

border pixel thickness of border near scratch for partial restoration.

Mark shows the potential scratch lines for ease of viewing and for debugging. It shows chosen pixels in the color green, close but still rejected in yellow, and extreme pixels in the color red. This makes it easy to vary some parameters to choose more or fewer scratch lines.

Reset activating this button returns all of the parameters to their default values.

Figure 10.43 shows a before and after DeScratch scenario. With *Mark* set, you can see the black lines which indicate what the program was looking at to determine the scratches to remove.

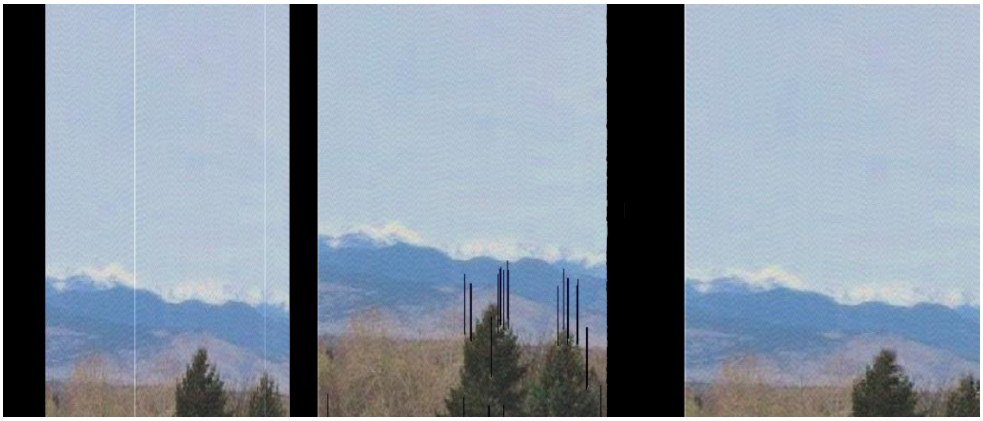


Figure 10.43: *Original video with scratch; Option Mark selected and Final video*

10.10.23 Decimate

This is used to decrease the frame rate of a video. Changing the frame rate means eliminating a frame for any given number of frames ($1 in N$); but if frames that are important for visual continuity are deleted, temporal artifacts arise: flickering, slowdowns, accelerations, etc. The Decimate filter maintains a higher quality because it first eliminates duplicate frames or frames that are most similar, thus limiting the appearance of artifacts. It is often used after the *Invert Telecine* plugin to make the video more smooth.

One use of the decimate effect can be applied to a DVD to convert the 29.97 *fps* video to the 23.97 *fps* film rate, but the effect can take any input rate and convert it to any lower output rate. The output rate of decimate is the project frame rate. The input rate is set in the decimate user interface. To convert 29.97 *fps* progressive video to 23.97 *fps* film, apply a decimate effect to the track. Set the decimate input rate to 29.97 and the project rate to 23.97.

Keep in mind that every effect layered before decimate, processes video at the decimate input rate and every effect layered after decimate, processes video at the project frame rate. Computationally intensive effects should come below decimate.

10.10.24 Deinterlace

The deinterlace effect has evolved over the years to deinterlacing and a whole lot more. In fact two of the deinterlacing methods, *Inverse Telecine* and *Frames to Fields*, are separate effects. The deinterlace effect offers several variations of line replication to eliminate comb artifacts in interlaced video. It also has some line swapping tools to fix improperly captured video or make the result of a reverse effect display fields in the right order.

10.10.25 Deinterlace-CV

Selection of deinterlacing mode for your video to eliminate comb artifacts (figure 10.44).

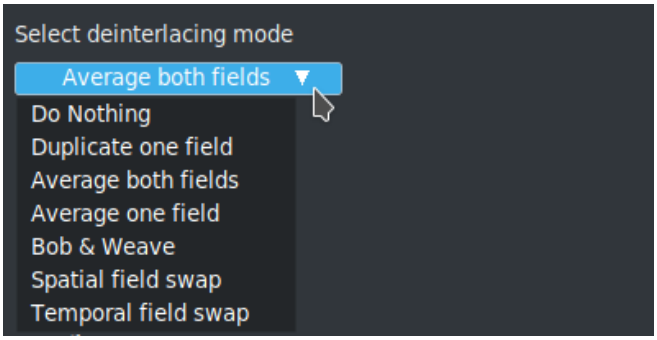


Figure 10.44: *Pulldown menu*

10.10.26 Delay Video

Delay the video by some number of seconds.

10.10.27 Denoise Video

Denoise video (figure 10.45).

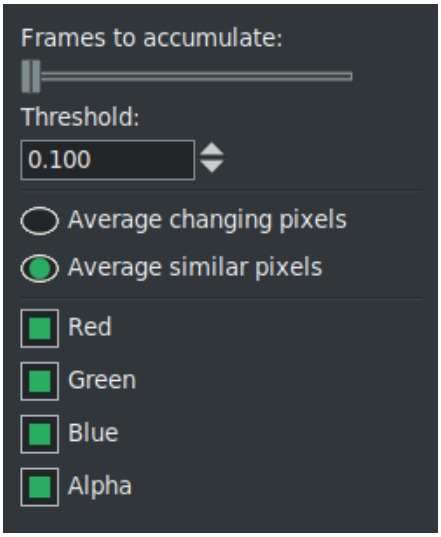


Figure 10.45: *Control window of the DeNoise plugin*

10.10.28 Difference key

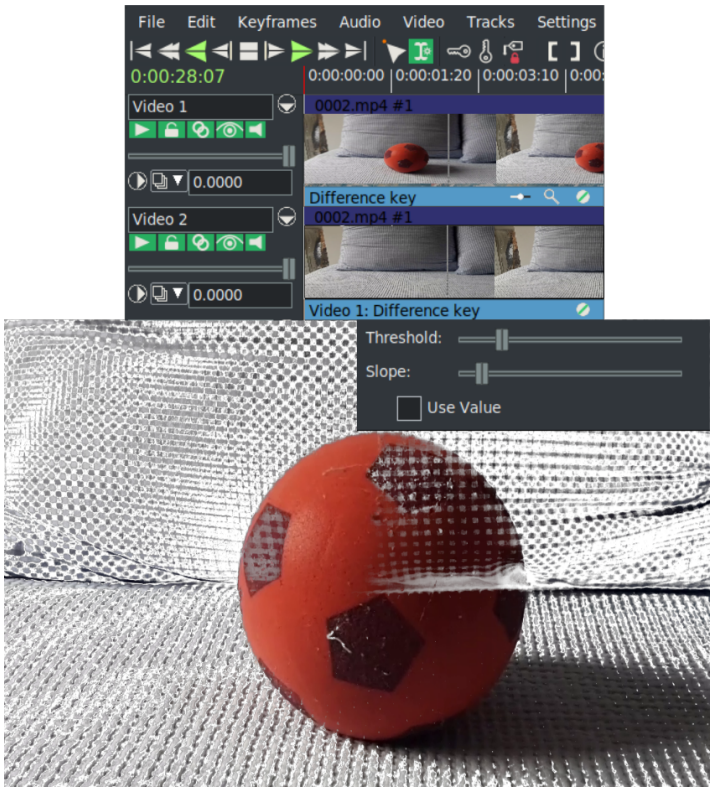


Figure 10.46: *Difference key and its problematic output*

The difference key creates transparency in areas which are similar between two frames. The Difference key effect must be applied to two tracks. One track contains the action in

front of a constant background and another track contains the background with nothing in front of it. Apply the difference key to the track with the action and apply a *shared effect* of it to the track with the background. The track with the background should be muted and underneath the track with the action and the color model should have an alpha channel. It's hard to get good results.

Pixels which are different between the background and action track are treated as opaque. Pixels which are similar are treated as transparent. Change *threshold* in the difference key window to make more pixels which are not the same color transparent. Change *slope* to change the rate at which the transparency tapers off as pixels get more different. The slope as defined here is the number of extra values flanking the threshold required to go from opaque to transparent. A high slope is more useful with a low threshold because slope fills in extra threshold. *Use value* causes the intensity of pixels (*luma*) to be compared instead of the color. Applying a *blur* to the top track with just the alpha channel blurred can soften the transparency border (figure 10.46).

10.10.29 DotTV

Puts various size dots over the picture to simulate TV effect. This came from: `https://effectv.com`.

10.10.30 Downsample

Downsampling is the process of reducing the size of an image by throwing out data, reducing sampling rate.

10.10.31 Edge

Display only the edges of the video throughout the image.

10.10.32 Fields to frames

See the theory description in the *Frames to Fields* plugin. This effect reads frames at twice the project framerate, combining two input frames into a single interlaced output frame. Effects preceding fields to frames process frames at twice the project frame rate. Each input frame is called a field.

Fields to frames needs to know what field corresponds to what lines in the output frame. The easiest way to figure it out is to try both options in the window. If the input fields are the result of a line doubling process like frames to fields, the wrong setting results in blurrier output. If the input fields are the result of a standards conversion process like *1080 to 480*, the wrong setting will not make any difference.

10.10.33 Flip

This effect flips a video track either vertically or horizontally.

10.10.34 Foreground

Whatever the visual content of the frame, the Foreground plugin application applies a uniform color that can be chosen by color picker, color wheel, color presets, or by entering the hexadecimal value. In addition there are HSV, RGB, YUV, and alpha sliders to vary the color accordingly. If the original source has values outside the range $(0-1.0f)$, the plugin will always produce clipping.

10.10.35 Frames to fields

Theory behind the Frames to Fields and Fields to Frames plugins

Historically, CRT-type TVs used interlaced signals to save bandwidth. An interlaced video consists of two *fields* that are read and drawn on the screen one after the other. Each field must be played at a framerate double that of the resulting video. In two steps the complete frame will be reconstructed.

Frame 1 $\Longrightarrow$ F1-field1 (*Top* or *Odd*), F1-field2 (*Bottom* or *Even*)

Frame 2 $\Longrightarrow$ F2-field1, F2-field2

Interlaced video reading: $F1-f1$ then $F1-f2$ then $F2-f1$ then $F2-f2\ldots$ There may be visual problems if the Top type interlacing is read according to a Bottom scheme. So it's important to know if a video is Top or Bottom. Generally an *HD* video is Top; a *DV* video (both PAL and NTSC) is Bottom; *SD* (PAL) is Top; *SD* (NTSC) is Bottom (but not always). Instead, high-definition videos need to be more compressed and this contrasts with the interlacing that is little and badly compressible, so modern videos are mostly *progressive*.

In CINELERRA-GG

1. upload an interlaced video to the Timeline and Resources and play it for viewing.
2. The video presents visual artifacts because PC monitors are progressive.
3. In the Resources window, open the media *Info* with the right mouse button. Below you can see that the *asset's interlacing* is active. It has four options for settings interlacing type: *Unknown*, *Top Fields first*, *Bottom Fields first*, and *Not interlaced*. If the file is (H)DV type, recognition and configuration is done automatically. All other media types will be set *unknown*. We need to set the type of interlacing, so we have to manually set the interlacing.
4. Now we can to use the *Frames to Fields* plugin, but we have to configure it and act manually.

Now for the practical use of this plugin which applies the operation reverse to the *Fields to Frames* plugin. It extracts the two interlaced fields stored in alternating lines of interlaced source footage and outputs them as separate full frames. The alternating lines missing on each output frame are interpolated.

This plugin is only useful if its output is pulled with doubled framerate with respect to the source footage. One typical usage scenario is to do *masking*, *scaling* and *translating* on interlaced footage without the need to destroy the additional temporal information contained in such source material. This is helpful if your intended target format is interlaced. If on the other hand, you just want to target a progressive display (for example, you create video for display on a computer monitor solely) then it is much more convenient to de-interlace the source material prior to any further processing.

Processing interlaced footage without deinterlacing

1. Create a new project with doubled frame rate. That is, make it 50 *f ps* if your source footage is 25*i*. In `Resources` → `Media` → `Info` set the interlaced type (or unknown)
2. Insert your source footage onto a video track in the timeline. Now, CINELERRA-GG will playback each frame of your footage twice. There will be visual artifacts because the video is interlaced and the monitor is progressive.
3. Apply the *Frames to Fields* effect. Be sure to choose the correct field order. If we know or believe that the original video is *Top First* let's try it first, but it doesn't have to be the right solution. The only way is to playback and look for visual artifacts.
4. Then apply any further effects afterwards, including translations, scaling, slow motion, precise frame-wise masking or use of the motion tracker plugin.
5. Render your project to an intermediate clip. Be sure to choose a rather lossless video codec, for example *Motion-JPEG-A* or even *uncompressed YUV* if you have plenty of storage.
6. Insert the intermediate clip into your original project. Make sure the doubled framerate has been detected correctly by CINELERRA-GG (by looking in the clip's media *info* in the media resources folder).
7. Apply the *Fields to frames* effect to the intermediate clip. This will combine two adjacent fields into one interlaced field with the original frame rate.
8. Do the final render on your original project. Now there will be no visual artifacts on the monitor.

10.10.36 Freeze Frame

In its simplest form, highlight a region of the track to freeze, drop the `freeze frame` effect on the highlighted region, and the lowest numbered frame in the affected area will play throughout the entire region. Freeze Frame has an enabled option which can be keyframed. Regions of a freeze frame effect which are enabled, repeat the lowest numbered frame since the last keyframe. This has unique possibilities.

- If a freeze frame effect has a keyframe in the middle of it set to enabled, the frame in the middle is repeated in the entire effect.

- If a freeze frame effect has several keyframes, each set to enabled, every time a keyframe is encountered the frame under it becomes the frozen one.
- If a freeze frame effect alternates between enabled and disabled, each time an enabled keyframe is encountered the frame under it is replicated until the next disabled keyframe. The disabled regions play through.

10.10.37 Gamma

Log camera images store colors in a $logarithmic$ scale. The blacks in these images are nearly 0 and the whites are supposed to be infinity. The graphics card and most video codecs store colors in a $linear$ scale but CINELERRA-GG keeps log camera images in their original logarithmic scale when it renders them. This is necessary because the raw image parser can not always decode the proper gamma (γ) values for the images. It also does its processing in 16 bit integers, which takes away a lot of information.

Mathematically, the gamma function is exponential ($output = input^{\gamma}$) and therefore the inverse of the logarithmic function [$\gamma = \log_{input}(output)$]. Actually the formula used by the CINELERRA-GG plugin is: $output = input^{\frac{1}{\gamma}}$ which allows for a range of values $0 \div 1.0$. The gamma effect converts the logarithmic colors to linear colors through a *gamma value* and a *maximum value*. The gamma value determines how steep the output curve is (i.e. the value of the gamma parameter; for color space Rec709 is 2.4 ($\frac{1}{\gamma} = 0.41\ldots$), for sRGB is 2.2 ($\frac{1}{\gamma} = 0.45\ldots$), etc.). We use $\frac{1}{\gamma}$ because with $\gamma < 1$ there is gamma compression of the curve, increasing the output values relative to the linear (see figure 10.48.). With $\gamma > 1$ we have gamma expansion, typically used to linearize a compressed gamma curve (*Log*). The maximum value is where 1.0 in the output corresponds to maximum brightness in the input. It serves to avoid clipped values because it allows you to set the maximum value of the output, 1.0, whenever range adjustment is done (see figure 10.47). It is important to adjust the two parameters accurately in order to avoid undesired and unexpected effects, such as excessive values, unbalanced image, incorrect linearization, etc.

The gamma effect has two more parameters to simplify gamma correction. The automatic option causes it to calculate max from the histogram of the image. Use this when making a preview of a long list of images since it changes for every image. The use color picker option uses the value currently in the color picker to set the maximum value. Note that every time you pick a color from the compositor window, you need to click on use color picker to apply the new value.

The best use of the gamma is manually monitoring the waveform as shown in figure 10.48.

- Look at the highest peak on the *waveform* and measure it with the crosshair observing the numerical value at the top left.
- Set this value with the *maximum* slider or *textbox*.

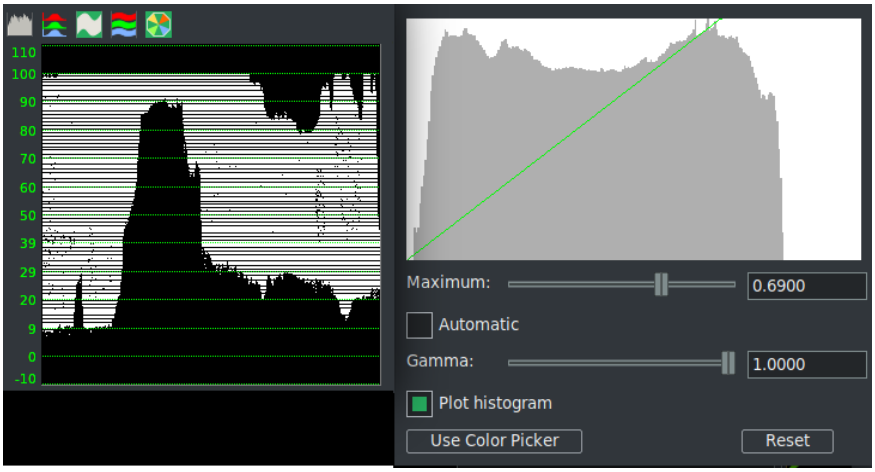


Figure 10.47: *settting Maximun to* 0.6900

- Then adjust the slider of the *gamma* to our liking, always checking the result on the waveform so to be sure never to exceed the values of clipping, $0 \div 1.0$.

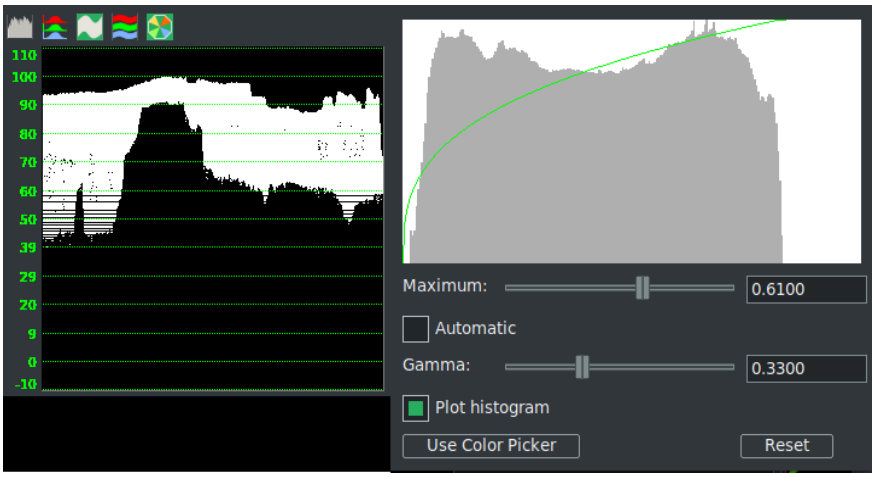


Figure 10.48: *Setting Maximun to* 0.6100 *and Gamma to* 0.3300

Care must be taken when using gamma correction: if the image carries a specific gamma value, or if it has already been corrected previously (for example automatically in the camera), etc.; then a second application of the gamma leads to excessive and artificial results. Gamma is keyframable.

10.10.38 Gradient

The *gradient* effect overlays a smooth color gradient on top of every video frame. It is useful for all sorts of background fills, for partially filtering, adding depth to the image, or for adding moving highlights. The Gradient effect can generate linear or circular color fills / shape. For linear fills, you can choose the *angle*, for circular fills the *center* (X, Y) of

the created gradient pattern. You can control the *slope* of the color transition by selecting a transition function (*linear*, *logarithmic*, *squared*) and by changing the start (*inner*) and stop (*outer*) radius. Note that both colors used in this color transition can contain an arbitrary *Alpha* value (transparency). All parameters can be keyed and will be interpolated between keyframes.

The first time you use the plugin it may seem complicated, but if you understand that we have to adjust the gradient from an inner spot we choose to an outer spot we also choose, the work will become easy and fast.

Use case (Vignette)

1. Setting the shape to radial
2. Setting the rate to Linear (or Log or Square)
3. Position X and Y to center the main figure.
4. Choose inner color=black
5. Adjust inner radius
6. Bring the inner color alpha slider to 0
7. Choose outer color=black
8. Check that the outer color alpha slider is 1.
9. Adjust outer radius

Note: The inner and outer colors are visibly mixed in the gradient area. If you want to make a vignetting of only black, you must set the two colors to black and then make the inner one transparent so that it does not cover the figure.

10.10.39 HistEQ

Remap colors using blended histogram weights. Figure 10.49 shows the GUI and the results in a split screen.

If the histogram is as wide as the entire range ($0-1.0$), the image is high contrast. If the range is small (i.e., the values are close together) the contrast is small. In this case HistEQ is used to distribute the few clustered values across the entire range ($0-1.0$), increasing the contrast of the frame. In order to do this HistEQ equalizes the colorspace through use of a *histogram equalization algorithm*. Parameters are:

Gain: when set to 1, the colorspace is best effort. If the gain is set to 0, the result is the entire regression line of the color map.

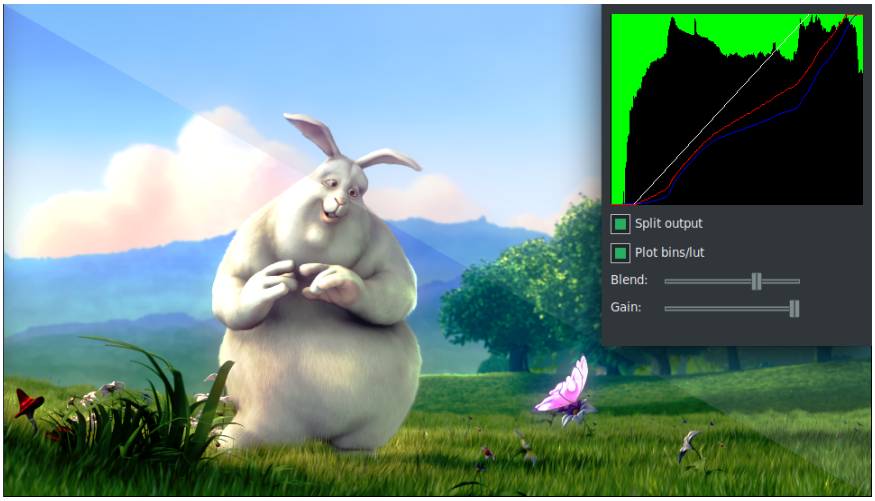


Figure 10.49: *Control window and split screen*

Blend: goes between a straight and a twisted line.

Split output: diagonally shows in the compositor, the new results on the left and old on the right.

Plot bins/lut: displays a plot of the result.

10.10.40 Histogram

The histogram allows an immediate view of the contrast amplitude of an image with its distribution of *luma* and *colors* values. If the columns of values occupy the whole range 0—1.0 (also expressed as a percentage above the graph) then we have maximum contrast; if the range is smaller the contrast is smaller. In fact, contrast is mathematically given by the highest value of the abscissa minus the lowest value. If the values are similar and close together, the difference is smaller and the lower the contrast. If most of the values are on the right of the histogram you have an image with highlights at the limit with values clamped to 1.0. This is called *overexposure*. However, if most of the values are moved to the left, with the limit of the values clamped to 0, we have a lowlight image and we talk about *underexposure*. This plugin is a tool for Primary Color Correction, that is, affecting the entire frame. Histogram is keyframable (figure 10.50).

The Histogram is always performed in floating point RGB regardless of the project color space, but with clipping at 1.0. When using the X11 graphics driver and RGBA-FLOAT color model, Histogram allows you to display greater than (0 - 1.0f) values to accomodate HDR. This does not work when using X11-OpenGL because it is an 8-bit limited driver. The display will stop at +110%, but there is no clipping. By lowering the brightness all out-of-range values become visible, even those initially above 110%.

The histogram has two sets of transfer parameters: the *input transfer* and the *output transfer*. The input transfer has value on the horizontal axis of x; it is a normalized scale of

Figure 10.50: *Master Histogram and RGB Histogram*

values ranging from 0 - 1.0 (which for a depth color of 8 bits corresponds to the range 0 - 255, for 10 bits corresponds to 0 - 65536, etc). The output transfer (the $yaxis$) represents the height of the column where a given value x appears. A higher column (y greater) indicates that many pixels have the corresponding value x; a lower column indicates that fewer pixels have that value. On the left we have the minimum value 0, which is the black point. On the right we have the maximum value 1.0 which is the white point. The intermediate values pass smoothly from one extreme to the other. The three important points (including the midtones) are indicated by cursors (small triangles) at the base of the histogram. You can adjust them to change the values of the three points if you want. Acting on the white or black point involves horizontal shifts only; acting on the midtones triangle also involves vertical movements leading to a "gamma" correction of the curve (Linear to Log or Reverse Log or vice versa). Further down is an additional bar with related cursors and textboxes. It is used to adjust input and output values (on the vertical).

There are 4 possible histograms in the histogram viewer. The red, green, blue histograms show the input histograms for red, green, blue and multiply them by an input transfer to get the output red, green, blue. Then the output red, green, blue is scaled by an output transfer. The scaled red, green, blue is converted into a value and plotted on the value histogram. The value histogram thus changes depending on the settings for red, green, blue. The value transfers are applied uniformly to R, G, B after their color transfers are applied. Mathematically, it is said that the values of x are linked to the values of y by a transfer function. This function, by default, is represented by a line that leaves the values of x and y unchanged, but we can intervene by modifying this line with the cursors or the textboxes.

You need to select which transfer to view by selecting one of the channels on the top of the histogram. You can also choose whether to display the master, i.e. only the values of the *luma*, or show the *Parade*, i.e. the three RGB channels. You can switch from one to

the other with the two buttons in the upper right corner. The input transfer is defined by a graph overlaid on the histogram; this is a straight line. Video entering the histogram is first plotted on the histogram plot, then it is translated so output values now equal the output values for each input value on the input graph.

After the input transfer, the image is processed by the output transfer. The output transfer is simply a minimum and maximum to scale the input colors to. Input values of 1.0 are scaled down to the output's maximum. Input values of 0 are scaled up to the output minimum. Input values below 0 are always clamped to 0 and input values above 1.0 are always clamped to 1.0. Click and drag on the output gradient's triangles to change it. It also has textboxes to enter values into.

Enable the **Automatic** toggle to have the histogram calculate an automatic input transfer for the red, green, and blue but not the value. It does this by scaling the middle 99% of the pixels to take 100% of the histogram width. The number of pixels permitted to pass through is set by the Threshold **textbox**. A threshold of 0.99 scales the input so 99% of the pixels pass through. Smaller thresholds permit fewer pixels to pass through and make the output look more contrasty. *Plot histogram* is a checkbox that enables plotting the histogram. It can be useful because having a histogram that changes in real-time can use a lot of system resources for computation. In this case it is enabled only at times when it is needed. *Split output* is a checkbox that enables a diagonal split showing in the compositor, so we can see the effects of the changes from the original frame. *Reset* returns the four curves to their initial state (neutral) as well as the Value/RGB histogram buttons.

Linear to Log slider: frequency in Linear range to Log range; default is 50%, but if you want it to be the same as the Videoscope or Histogram Bezier, which are of type Log, set the slider all the way to Log. The variation is by 10% steps. Linear generally represents what the eye sees in the real world. Log is an exponential look where the small numbers are represented more - that is, the leading digit.

The histogram plugin ordinarily applies a value for a single frame. A new feature modifies histogram to show values for a selection of frames. You use several frames in a video that you want to use the *average* color to determine the color you want the rest of the video to be. When there is a large light or color variation within a range of a few frames, you spend a lot of time correcting each frame when it would be better to just get an average value to use. If you want, then you can make a LUT for that set of frames instead of each frame.
It works by selecting the area on the timeline for which you would like to see the Histogram averaged and then click on the **sum frames** button in the Histogram plugin. If a Selection is set on the timeline, the number of frames will be calculated and equalized to the mean value.

A side note: by using a number of frames, you can get a *dissolve-like transition effect*.

Theory: For more on the mathematical aspect see here:

https://thirdspacelearning.com/gcse-maths/statistics/histogram/

For our discussion, it is enough to understand a few concepts. Each vertical line we see in the histogram is not a simple line, but a rectangle having a certain base. This base is given by the values of x_i present at the edges of the rectangle i (pixel range, $x_{max_i} - x_{min_i}$). Rectangles are called *Bins* or Accumulators. The bin's number is of fixed and known size because it depends on the color depth. The bin height is our output y and the bin area ($A_i = f(x_i)$) is known because it represents the *number of occurrences* that are read in bin, also called *frequency (f_x)*. The plugin scans the entire range of x_i, from 0 to 1.0, and records all the occurrences within each bin. The value of f_x for each bin is the *max* value. At this point knowing base and area, we can obtain the value of y axis that is reported in the histogram.

$$width = b_i = x_{max_i} - x_{min_i}$$

Having established the depth color, the bin width is always the same (b) for every bin_i:

$$b_i = b = \frac{range(1.0 - 0)}{\#bins}$$

$$A_i = f(x_i) = max_i = Base \times High = b \times y_i$$

Hence:

$$y_i = \frac{f(x_i)}{b}$$

Dim bins are on the left, bright bins on the right. You can have discordance of results, looking in the scopes, either by switching from Histogram to Histogram Bezier or after a conversion between color spaces (with associated change in depth color). The number of bins used depends on the color model bit depth:

Histogram: 256 for rgb8 and 65536 for all others

Bezier: 256 for rgb8/yuv8 and 65536 for all others

Scopes: always uses 65536

All of the bins are scanned when the graph is plotted. What is shown on the graph, in the Compositor window, and finally on the scopes, depends on which plugin is used:

Histogram: was max of the bins in the pixel range, now is the sum (to make it congruent with Bezier and the Scopes)

Bezier: is the max of the bins in the pixel range

Scopes: is the max of the bins in the pixel range

Another difference in behavior is regarding the type of curve, whether Linear or Log:

Histogram: the curve is Linear, but it is editable with the Linear/Log slider

Bezier: the curve is Log

Scopes: the curve is Log

This diversity also leads to different visual results from Histogram Bezier or Videoscope.

When the color space and the bin size are the same, all of the values increment the indexed bins. But if we start from YUV type edits, the plugin will automatically do the conversion to RGB. When the color is the result of yuv → rgb float conversion, we go from 256 bins of YUV to 65536 bins of RGB Float and the results *spread* if there are more bins than colors. This is the same effect you see when you turn on *smoothing* in the vectorscope histogram.

The *total* pixels for each value is approximately the same, but the *max* value depends on the color quantization. More colors increment more bins. Fewer colors increment fewer bins. In both cases, the image size has the same number of pixels. So the pixels will distribute into more bins if you go to a higher depth color; those bins will have a lower count. The fewer color case increments the used bins, and skips the unused bins. This sums all of the pixels into fewer bins, and the bins have higher values. That is the *rgb* vs *yuv* case, fewer vs more bins are used. To get more consistent visual feedback (and on scopes), the concept of *sum* was used instead of the maximum number of occurrences (max).

To report something more consistent, the reported value has been changed from the original code to be the *sum* of the accumulated counts for the bins reporting a pixel bar on the

1				
1			1	
000100	3 pixels	vs	0011000	3 pixels

On the left, the course color model piles all 3 pixels into one bin, max value 3. On the right, the fine color model puts the counts into 2 bins, max 2, sum 3.

So, by reporting the sum the shape of the results are more similar to Bezier.

10.10.41 Histogram Bezier / Curves

Histogram Bézier allows an immediate view of the contrast amplitude of an image with its distribution of luma and colors values using a piecewise linear method (bins). In addition it uses a Bézier curve (parametric) on the histogram plot. When mapping color spaces, it has a variety of presentations to get smoother transitions and more pleasing output. It uses more general remapping, not just straight lines but more contour lines. Curves are perhaps the most powerful and sophisticated tool for color correction. For some repetitive details, see the previous description of the Histogram plugin. Histogram Bézier is keyframable. The scale is given in percent ($0-100\% \pm 10\%$). The default curve is Log type and can use either RGB or YUV sources without implementing conversions. If the original source has values outside the range ($0-1.0f$), the plugin will always produce clipping.

NOTE: Histogram Bezier may give results that are not congruent with Histogram plugin. To understand the difference in behavior see the Theory section in Histogram plugin.

The input graph is edited by adding and removing any number of points. Click and drag anywhere in the input graph to create a point and move it. Click on an existing point to make it active and move it. The active point is always indicated by being filled in. The active point's input X and output Y values are given in textboxes on top of the window. The input and output color of the point can be changed through these textboxes. Points can be deleted by first selecting a point and then dragging it to the other side of an adjacent point. They can also be deleted by selecting them and hitting delete (figure 10.51).

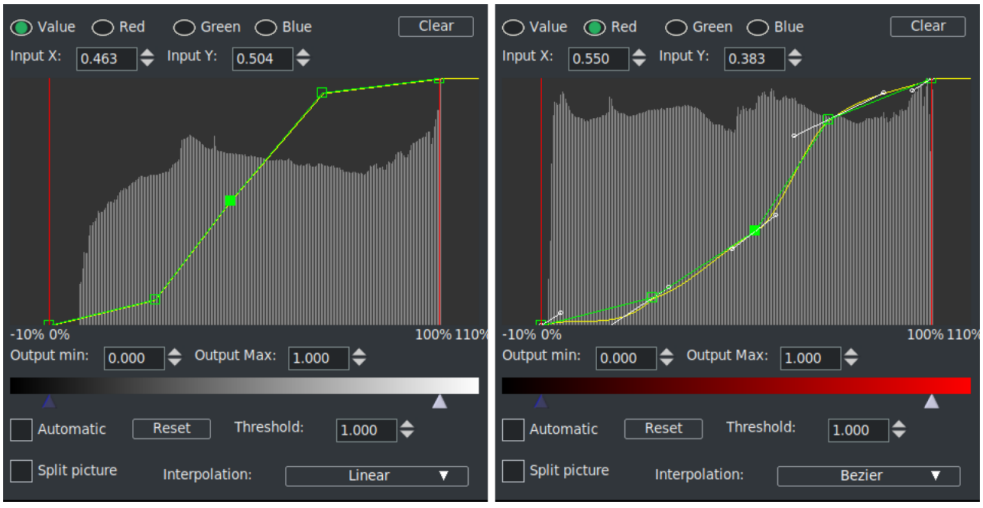


Figure 10.51: *Histogram Bezier / Curves*

- *Master* (value) and *R*, *G*, *B* histograms.
- Textbox for input *x* (*input*) and input *y* (*output*).
- *Output min* and *output max*: sets black or white points. If you use both points it works as Master/Color Offset. Values can also be less than 0 and greater than 1.0.

- Scale for *contrast range*: sets with cursors shown as little triangles.
- *Automatic* and *Threshold*: enable the Automatic toggle to have the histogram calculate an automatic input transfer for the red, green, and blue but not the value. It does this by scaling the middle 99% of the pixels to take 100% of the histogram width. The number of pixels permitted to pass through is set by the Threshold textbox. A threshold of 0.99 scales the input so 99% of the pixels pass through. Smaller thresholds permit fewer pixels to pass through and make the output look more contrasty.
- *Reset:* returns the four curves to their initial state (neutral).
- *Split picture:* a checkbox that enables a diagonal split showing in the compositor.
- *Interpolation:* type of algorithm for the parametric curves; linear, polynomial and Bezier.

Curves are used by introducing *control points* simply with the left mouse button and adjusting the value by dragging and dropping. If you drag along the horizontal line only, you change the value of *x* and you can read this value in the input *x* textbox. If you drag along the vertical line only, you change the value of *y* and you can read the value in the input *y* textbox. This is the output value. The newly clicked control point becomes active and is full green in color. To delete a point we have to make it active and then press the Del key, or we can drag the point beyond the position of another control point to its right or left or, finally, pressing RMB. The control points corresponding to the black point and the white point are automatically created from the beginning, to fix their values and prevent clipping.

Curves are generally adjusted by introducing several control points, some to be kept fixed (as anchors) to prevent curve modification beyond them, and others to be dragged to make the desired correction. The power of the curves lies in being able to circumscribe a small interval at will and intervene only on this without involving the remaining parts of the frame. The precision with which you can work is such that you can almost arrive at a secondary color correction.

The most used type of modification is to create a *S curve*. There can be a lot of shapes that use the S curve; the simplest is to create a control point in the shadows, one in the midtones (anchors) and one in the highlights. Moving the highlight point upwards and the shadow point downwards increases the contrast, making the image sharper and improving the color rendering. With the type of *linear* curve you can make hard adjustments, similar to the result of the use of *Color 3 Way*, even if this acts on the color wheel (Hue) while the curves act on individual RGB channels.

The *Polynomial* and *Bézier* types introduce *control handles* that allow for more sophisticated and smoother adjustments. The quality of the result is much better, but they require more experience for their optimal use. By varying the angle of the handles we change the *tangent* and thus the slope of the curve below. Unlike true Bezier curves, there is no

effect in extending the handles: there is no change in the radius of the curve. The difference between Polynomial and Bézier lies in the underlying mathematics, but for practical purposes the use is similar. By default Bézier starts with a slight S-curve, as opposed to Polynomial.

Some examples of the use of curves to demonstrate the variety of possible interventions (figure 10.52):

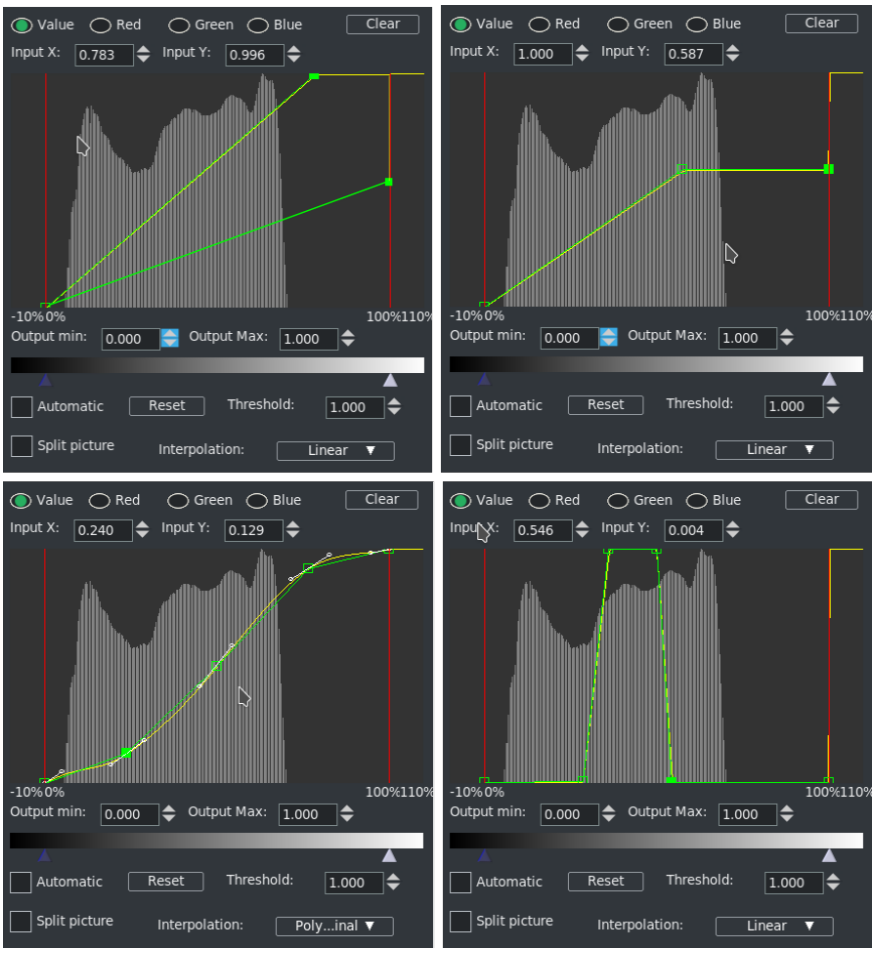


Figure 10.52: *Gain Up/Down; clamp; S-Shaped curve and Luma Key*

- Scale the image values by increasing the white point or decreasing the white point (gain up and gain down). You can decide the scaling value with the formula: $(Input \div Output) = ScaleFactor$
- Limit a value beyond a certain point of brightness (clamp to the value 0.587 in the figure).
- S-shaped curve to increase contrast without changing the black and white point (i.e. without *clipping*).

- Make a real *Luma Key* by bringing a certain value of gray to 100% (white) and lowering everything else to 0% (black). The slope of the two sides indicates how much we want to fade the edges of the matte obtained.

10.10.42 HolographicTV

Figure 10.53: *Holographic messages in CinGG!*

Incoming objects are projected like holovision seen in the movie Stars Wars as in R2-D2's video message projector of the Princess Leia (figure 10.53). You need a movie or background image and above it a track containing the figure on which to apply the effect. This must have a transparent background. There are no configuration parameters; it only has to be applied to the upper track.

This effect originated from `https://effectv.com`.

10.10.43 Hue saturation

With this effect you can change hue, saturation and value. The parameters are modified using 3 simple sliders. For maximum precision you can use the input box. The *hue* control shifts the colors circularly in the color plane, normally resulting in false colors. The *saturation* control can be used to reduce color footage to black and white. The *value* control makes any given colors more bright or more subdued. *Clear* buttons are present to reset its slider to default without affecting others.

10.10.44 Interpolate Bayer

Uses a Bayer filter algorithm to interpolate (estimate) missing color information. This is needed for some cameras where each pixel location only has an R or G or B value instead of all R, G, and B values for each location. The algorithm creates values for each of the three colors at every location by smearing (interpolating) each set of partial R, G and B values to create values at every pixel location.

10.10.45 Interpolate Video

Theory

Each video has its own framerate. If we want to change it (for *timelapse* or *slow motion*) the best thing is to shoot the scene with suitable framerate. But even in post production we can do something. The simplest method is to remove some frames to speed up the movie or add some to slow it down (from now on, for simplicity we will consider only the timelapse). Needless to say, the result is not smooth and the viewer will notice it immediately. A better method is to use the interpolation, mediating the pairs of frames that alternate. For example, if we have a sequence of frames $1, 2, 3, 4, 5, 6, 7, 8 \ldots$ we can make a timelapse mixing frames 1 and 2, 3 and 4, 5 and 6, 7 and 8 and so on. So we will have a new sequence of 4 frames instead of the initial 8: $\underline{12}, \underline{34}, \underline{56}, \underline{78} \ldots$ We will get 50% acceleration but it will always be of bad quality because of the too rough blending between the pairs of frames. Blending can be improved by weighing it differently by $50\% frame1 + 50\% frame2$, but the result is still unsatisfactory. Further improvements can be achieved by using $logarithmic$ or $exponential$ interpolation instead of $linear$ interpolation. But the most sophisticated methods that lead to better results are based on *optical flow analysis*. These analyses the movement of circumscribed areas over a given period of time. With this method the intermediate frames do not derive from an approximate blending, but from the calculation of the *vector* of the motion between two frames that determines the displacement (*warping*) of the moving figure in the new intermediate frame. *Interpolate Video* works this way.

Practice

The practical use of *Interpolate Video* is a little different than the theory. The interpolate effect tries to create the illusion of a higher frame rate from source footage of very low framerates by averaging frames over time. It averages two input frames for each output frame. You choose a zone to be evaluated (*macroblock size*) and a radius (*search radius*) where you can search for this macroblock in the following frames. The *Use optic flow* button is activated and playback starts. The plugin will calculate the motion vector (which can be made visible by the *draw motion vectors* button) and apply it to intermediate frames. This operation is CPU intensive. Once the analysis is done, we can scroll the video by unchecking the two buttons and obtaining the desired result. There are two ways of specifying the input frames. You can specify an input frame rate which is lower than the project frame rate (*imput frames per seconds*). This causes input frames to be taken at even intervals. You can also specify keyframe locations as the positions of the input frames (*use keyframes as input*). In this mode the output frame rate is used as the input frame rate and you just create keyframes wherever you want to specify an input frame.

10.10.46 Inverse Telecine

This is the most effective deinterlacing tool when the footage is a video transfer of a film. This can be used to solve the problem, i.e., undo the damage caused by making film into a TV broadcast. That process came about because film is at 24 *fps* while TV is at 29.97 *fps* and fields are at 60. So the film was converted from 24 *fps* to 60 *fps*. Roughly speaking, converting every 4 frames into 5 frames plus a slight slow down in speed. Then the 60 *fps* was down-sampled to 30 *fps* by extracting odd and even lines and interlacing the lines.

This process is referred to as *three-two pull down* (3 : 2 pull down) in filmmaking and television production for the post production process of transferring film to video. The three-two pull down is where the telecine adds a third video field (a half frame) to every second video frame, but the untrained eye cannot see the addition of this extra video field. The `IVTC` effect is primarily a way to convert *interlaced* video to *progressive* video. It reverses the effect of three patterns of interlacing. In the next lines `A`, `B`, and `C` represent fields.

```
A AB BC CD D

AB CD CD DE EF

Automatic
```

The first two options are fixed patterns and affected by the pattern *offset* and *odd field first* parameters. The last option creates several combinations of lines for each frame and picks the most progressive combination. It is a brute force algorithm that is trying to resample the lines. This technique does not rely on a pattern like other techniques and is less destructive but the timing is going to be jittery because of the lack of a frame rate reduction. In order to smooth out the timing, you need to follow *inverse telecine* with a *decimate* effect.

10.10.47 Invert Video

Invert video is a method of reversing the colors of a video track. The four parameters refer to channels – *Red*, *Blue*, *Green*, *Alpha*. A very common use is to invert the alpha channel to change transparency.

10.10.48 Lens

Create the effect of looking through a lens.

R, G, B, A Field of View: quantity of deformation of the relative fields. Often used with `Lock` to simultaneously affect the 4 fields.

Aspect Ratio: determines the aspect ratio that you intentionally set.

Radius: radius of curvature of the distortion. At minimum, it is a sphere (*fish eye*) and at maximum, it is a rectangle (no distortion).

Center X, Y: determines the coordinates of the center of the sphere. It can be made visible with *Draw Center*.

Mode: determines the type of distortion. The choice is between *sphere shrink*, *sphere stretch*, *rectilinear shrink* and *rectlinear stretch*.

Interpolation determines the interpolation algorithm; from the fastest and least precise *Nearest*, passing through *BiLinear* to the better *BiCubic*.

10.10.49 Linear Blur

This effect acts only in one direction which can vary up to an angle of 180° with these parameters:

Figure 10.54: *For clarity of presentation only 2 fields are shown*

Length: distance between original image and final blur step; corresponds to the distance of the fields.

Angle: angle of motion in one direction for linear blur

Steps: number of blur steps to be used in the calculation. Increasing the number takes more CPU.

Channels: R,G,B,A.

Clear With the Clear buttons we can bring the slider to default values without affecting the other parameters.

For maximum precision you can use the input box. Figure 10.54 shown here has the parameters: $Length = 19$, $Angle = 25$, and $Steps = 2$.

10.10.50 Live Video

This effect reads video directly from the capture card input. It replaces any video on the track so it is normally applied to an empty track. Only one *Live Video* effect can exist at any time on the timeline. It can not be shared by more than one track. The configuration for the capture card is taken from the recording preferences. Go to `Settings` → `Preferences` → `Recording` to set up the capture card. There is a limitation when both Audio and Video use DV, in that you can not use this plugin at the same time as *Live Audio*.

In the *Video In* section where it says *Record driver*, it should be set to either *Video4Linux2* or *IEC 61883*. Other video drivers have not been tested with Live Video and probably will

not work. For live video, the selection for *File* Format and *Video* needs to be set to a format the timeline can use. The file format must be Quicktime for Linux and video recording must be enabled for it. Click on the wrench to set the video compression.

The video compression depends on the recording driver. For the *Video4Linux2* recording driver, the compression must be *Motion JPEG A*. For the *IEC 61883* driver, the compression must be *DV*. This gets the driver to generate output in a color model that the timeline can use. Some cards provide color and channel settings. Live video takes the color settings from the values set in the Video In window. Go to `File` → `Record` to bring up the recording interface and the Video In window. Values set in the Video in window are used by Live Video. Any channels the capture card supports need to be configured in the Video In interface since the same channels are used by the Live Video effect.

With the video recording configured, highlight a horizontal region of a video track or define in and out points. Then drop the Live Video effect into it. Drop other effects after Live Video to process the live video in realtime. For best results, you should use OpenGL and a video card which supports GL shading language. Go to `Settings` → `Preferences` → `Playback` → `Video Out` to enable the OpenGL driver which will make it possible to process video input in real time.

10.10.51 Loop video

Sections of video can be looped by dropping a loop effect on them. Contrary to the `settings` → `loop playback` option, the loop effects can be rendered where the `settings` → `loop playback` option can not be. The loop effects are also convenient for short regions.

The loop effects have one option: the *number of frames* or *samples* to loop. This specifies the length of the region to loop starting from either the beginning of the effect or the latest keyframe. The region is replicated for the entire effect.

Every time a keyframe is set in a loop effect, the keyframe becomes the beginning of the region to loop. Setting several keyframes in succession causes several regions to loop. Setting a single keyframe causes the region after the keyframe to be looped throughout the effect, no matter where the keyframe is. The end of an effect can be looped from the beginning by setting the keyframe near the end.

10.10.52 Mirror

The Mirror plugin splits the image along a line, an axis, and reflects one side onto the other. When the `Enable Reflection Center` option is checked you can move the position of this axis, using X% and/or Y% slider and/or textbox: X% for the horizontal position and Y% for the vertical position. Mainly in the Mirror effect there are 2 cases: with or without the *Enable Reflection Center* checked.

When the *Enable Reflection Center* option is UNCHECKED, the used reflection is on the

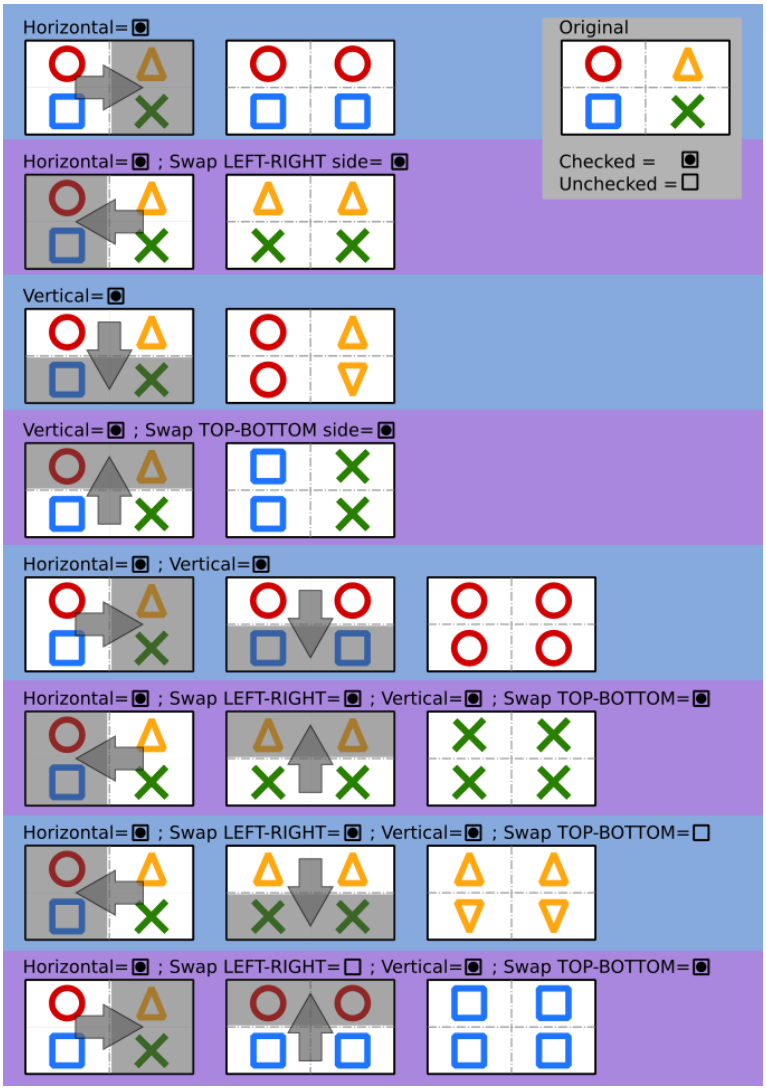


Figure 10.55: *Possible combinations with both Horizontal and Vertical checked*

center of the image (i.e. $X = 50\%$, $Y = 50\%$) and the options about *Reflection Center*, $X\%$ and $Y\%$, are disabled (grayed). The Horizontal option reflects the left side of the image on the right. If the `Swap LEFT-RIGHT side` option is checked then the right side to the left will be reflected. The Vertical option reflects the top on the bottom. If the `Swap TOP-BOTTOM side` option is checked then the bottom to the top will be reflected. There are special combinations when both Horizontal and Vertical checkboxes are enabled (and *Enable Reflection Center* is always UNchecked, of course). You can see those combinations in the figure 10.55.

When the *Enable Reflection Center* option is CHECKED the options about *Reflection Center*, i.e. $X\%$ and $Y\%$, are enabled, and the two "Swap ... side" options are disabled and hidden. These two "Swap ... side" are disabled because the $X\%$ and $Y\%$ allow the swap side feature automatically. The $X\%$ and $Y\%$ positions move the center of the reflection to fit to the

user's need.

10.10.53 Motion51

Note: Motion and Motion51 plugins are complex and slow to use for video stabilization. If you are stabilizing an entire video, you may want to preprocess by using ffmpeg's *vidstabdetect* and *vidstabtransform* plugins before importing the video into CINELERRA-GG.

This plugin compensates for unwanted motion and stabilizes the picture. The *Motion51* Plugin simplifies motion stabilization so that without a lot of tweaking you can easily achieve reasonable results, either by using the defaults or varying a single parameter. Since the motion in every clip is specific, there are some additional parameters useful to adjust the settings accordingly. The Motion51 plugin uses different methods for tracking than the other motion plugins. Motion Stabilization is very useful if you have jittery video, for example when taken from a car window, or while walking.

The better results require more samples. Setting the *sample set size* is probably the most important setup change. Also, when computing motion compensation, the entire history of the image motion is important, and so it is desirable to enable the playback setting *play every frame* in order to get good results. When every frame has to be processed, it can be time-consuming. Reasonable results are possible with small sample sets. After setup, the sample size can be increased to produce a high quality rendered result.

Description of what the program is doing

The motion is detected by *sampling* the video image in a circular field. This size and placement of the sample region defaults to most of the image area. When the *draw vectors* feature is enabled, the outer line trace encloses the searched region. The dotted circles define the target pixel set as the reference sample. The image is sampled using the circle pattern in a grid search. The best match is used to find the center and amount of rotation to transform the current image so that the reference area motion is canceled.

The amount of sampling does not significantly change for smaller or larger search areas. This means that a wide area can be searched just as easily as smaller areas. The main parameter which determines how hard it looks at the image is the *samples* parameter. It represents the number of possible rotations, as well as the search precision. More samples mean more precision, and less jitter, but the program will run more slowly (figure 10.56).

The Samples box at the top is most often the only parameter that you may want to vary.

Program Parameters Description follows

X11-OpenGL: setting can speed up the computation significantly in some cases when hardware OpenGL is available.

Samples: is the number of pixels which the software will examine to stabilize the picture. The sample set is arranged in 4 equal concentric circular sets. Each sample dot

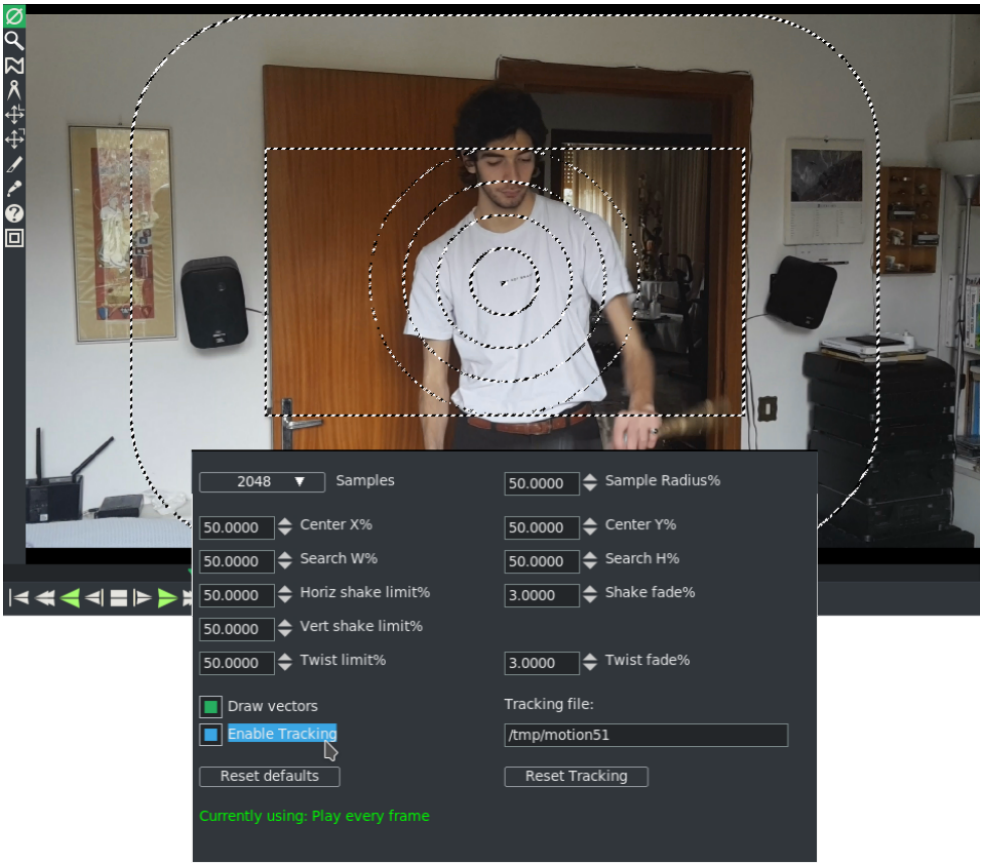


Figure 10.56: *Motion51 plugin window with its default options set.*

represents content and position for a pattern matching test. Setting the samples to larger values improves the match by adding lots of placement possibilities. The samples/pixels that will be utilized are distributed throughout the selected area – this is seen within the circles drawn when Draw Vectors is enabled. See figure 10.56.

Draw vectors: demonstrates the search operation of motion stabilization. When enabled, the outer search boundary (oval), the search grid area (rectangle), and the reference sample (circles) are visible. 4 concentric circles show the reference sample set (target). You will also see an arrow in the center of the circle which shows each image displacement from frame to frame. When you render the video using the motion plugin, these dots/lines/circles are drawn into the rendered output. Draw vectors helps to visualize the meaning of the parameters to aid in setup. You should disable Draw Vectors before the final rendering.

Sample Radius: is the radius of a circle that denotes the area of the sample locations. It is expressed as a percentage of the smallest image edge. For example, if it is set to 50%, then the circle will overlap about $\frac{1}{2}$ of the image. This does not change the number of samples. It does change the area from where the samples are gathered. If you have Draw vectors on, you can see the faint outline of a circle used for the radius.

Center X/Y: is the center position of the sample circle, as a percentage of image width and height. This is useful to reset the reference focal point in cases where the important feature target is off-center. When both X and Y are set to 50%, the samples will be used from around the center of the video.

Search W/H: determines the width and height of the rectangular area used for the grid pattern search. Samples are taken by moving the center of the circles in a grid pattern.

Horiz/Vert shake limit: (Shaking refers to image translation) determines translation constraints. If the motion determined by the search exceeds the limit, it is clipped to a value that is at the limit boundary. For example, if the match indicates that the motion is 60% off the reference target, but the limit is 50%, then the actual translation used will be limited to only 50%.

Shake fade: determines how fast the translation cancellation fades away, and the image resettles to its actual appearance. Every frame that is processed accumulates the motion of the past. The amount of the past motion which is applied is reduced by the fading factor. The current match is then added to the fading past motion history. Fading works fast. It is applied every frame. So if fade is 10%, and no new motion occurs in the input, the history will be $90\%, 81\%, 73\%, 66\%\ldots$ and in 30 frames only 4.2% of the past motion will be present in the effect. Fading insures that the image will eventually settle re-centered when the image motion ends.

Twist limit: (Twisting refers to image rotation) determines the rotation constraints. If the rotation determined by the search exceeds the limit, it is clipped to the limit boundary. Its operation is similar to the shake limits.

Twist fade: determines how fast the rotation cancellation fades away, and the image resettles to its actual appearance. Its operation is similar to the shake fade.

Enable Tracking: caches the search results in a file so that subsequent playback does not have to be recalculated. When tracking is enabled, before a frame is processed the frame number is used to look for cached results. If cache data is available, it is used. If no data is available, the frame is processed by the motion tracking search, and the results are added to the cache file. If tracking is not enabled, the data is always sourced from the motion tracking search and the tracking file is not updated.

Tracking file: is the name of the file which will contain the calculated values to be saved. Note that the default is `/tmp/motion51` which can be hazardous, since a system crash or a reboot can delete `/tmp` files.

Reset Defaults: button is used to revert to the initial defaults built into the program.

Reset Tracking: will delete the current Tracking file and disables tracking so that any previously calculated values are no longer available. However, because motion stabilization can often be cpu intensive, if the default file already exists, it will create a file name from the loaded asset.

Play Every Frame: shows if you are Currently using: Play every frame. For best results, set play every frame. This can be set in `Settings` → `Preferences` → `Playback A` Tab in the *Video out* section.

10.10.54 Motion

Note: Motion and Motion51 plugins are complex and slow to use for video stabilization. If you are stabilizing an entire video, you may want to preprocess by using ffmpeg's *vidstabdetect* and *vidstabtransform* plugins before importing the video into CINELERRA-GG.

The *motion tracker* is almost a complete application in itself. The motion tracker tracks two types of motion: *translation* and *rotation*. It can track both simultaneously or one only. It can do $\frac{1}{4}$ pixel tracking or single pixel tracking. It can stabilize motion or cause one track to follow the motion of another track. Although the motion tracker is applied as a realtime effect, it usually must be rendered to see useful results. The effect takes a long time to precisely detect motion so it is very slow.

Motion tracker works by using one region of the frame as the region to track (Match Box). It compares this region between 2 frames to calculate the motion. This region can be defined anywhere on the screen. Once the motion between 2 frames has been calculated, a number of things can be done with that *motion vector*. It can be scaled by a user value and clamped to a maximum range. It can be thrown away or accumulated with all the motion vectors leading up to the current position.

To save time the motion result can be saved in a file for later reuse, recalled from a previous calculation, or discarded. The motion tracker has a notion of 2 tracks, the *master* layer and the *target* layer. The *master* layer is where the comparison between 2 frames takes place. The *target* layer is where motion is applied either to track or compensate for the motion in the *master* layer (figure 10.57).

Motion tracking parameters[12]:

Track translation Enables translation operations. The motion tracker tracks X and Y motion in the *master* layer and adjusts X and Y motion in the *target* layer.

Translation block size For the translation operations, a block is compared to a number of neighboring blocks to find the one with the least difference. The size of the Match Box to search for is given by this parameter.

Translation search radius The size of the area to scan for the translation block.

Translation search steps Ideally the search operation would compare the translation block with every other pixel in the *Translation search radius*. To speed this operation up, a subset of the total positions is searched. Then the search area is narrowed and

[12] credit Georgy (sge)

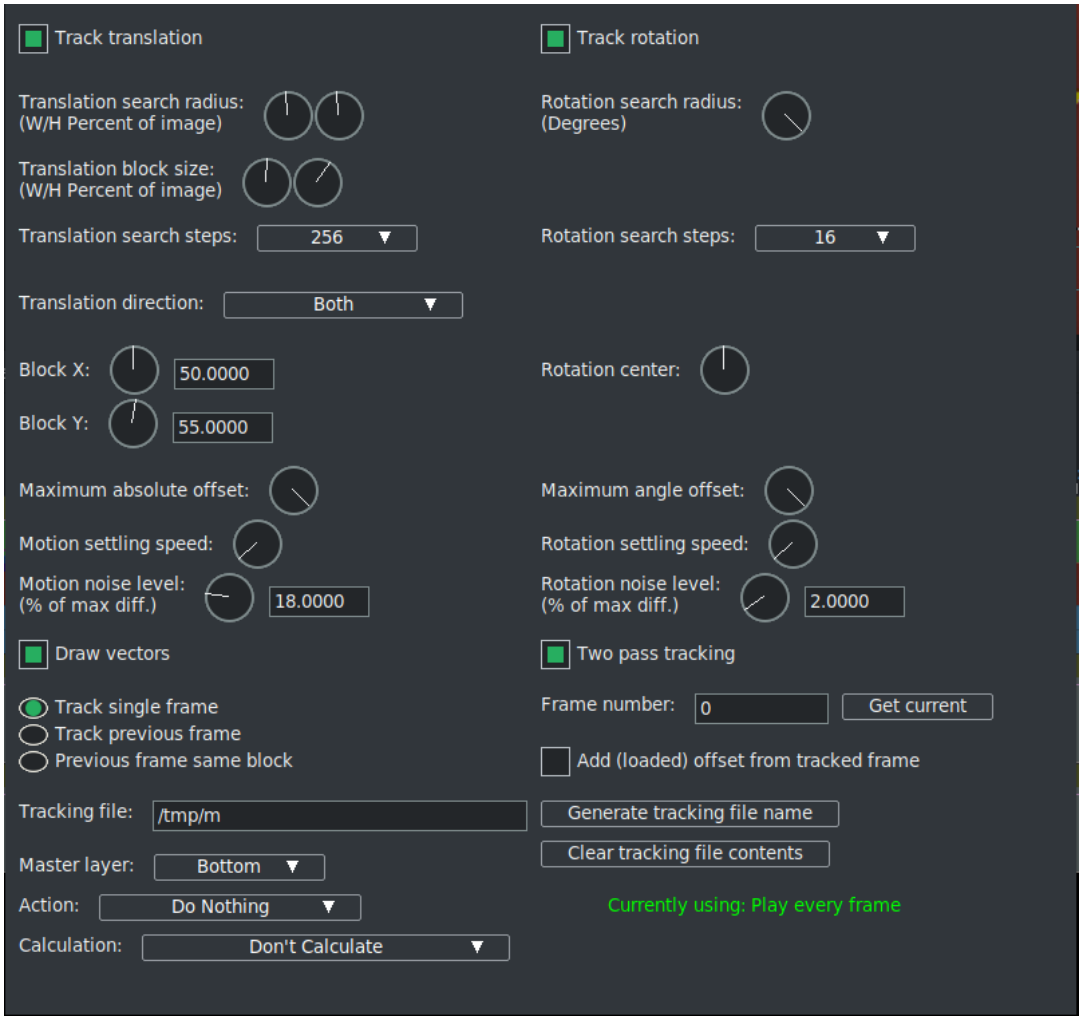


Figure 10.57: *Motion plugin GUI*

re-scanned by the same number of search steps until the motion is known to $\frac{1}{4}$ pixel accuracy.

Translation direction Tracking translation is possible in any direction, or in horizontal or vertical direction only, depending on this selection.

Block X, Y These coordinates determine the center of the translation block based on percentages of the width and height of the image. The center of the block should be part of the image which is visible at all times.

Maximum absolute offset The amount of motion detected by the motion tracker is unlimited if this is 100. If it is under 100 the amount of motion is limited by that percentage of the image size.

Motion settling speed The motion detected between every frame can be accumulated to form an absolute motion vector. Settling speed determines how fast the accumulated translation fades away, and the image resettles to its actual appearance. If the

settling speed is 0% the total absolute vector is added to the next frame. So every frame that is processed accumulates the whole motion of the past. If the settling speed is 100% the absolute vector is cancelled completely, adding no past translation to the next frame. If the settling speed is intermediate between 0% and 100% the absolute vector is downscaled by (100 − settling amount) before being added to the next frame.

Track rotation Enables rotation operations. The motion tracker tracks rotation in the *master* layer and adjusts rotation in the *target* layer.

Rotation search radius This is the maximum angle of rotation from the starting frame the rotation scanner can detect. The rotation scan is from this angle counterclockwise to this angle clockwise. Thus the *Rotation search radius* is half the total range scanned.

Rotation search steps Ideally every possible angle would be tested to get the rotation. To speed up the rotation search, the *Rotation search radius* is divided into a finite number of angles and only those angles compared to the starting frame. Then the search radius is narrowed and an equal number of angles is compared in the smaller radius until the highest possible accuracy is achieved. Normally you need one search step for every degree scanned. Since the rotation scanner scans the *Rotation search radius* in two directions, you need two steps for every degree in the search radius to search the complete range.

Rotation center Usually this parameter is zero, and the rotation search range is +/− *Rotation search radius*. Otherwise, it shifts the rotation search range to the area of angles between (*Rotation center* − *Rotation search radius*) and (*Rotation center* + *Rotation search radius*).

Maximum angle offset The total accumulated amount of rotation is unlimited if this is 90 degrees. If it is under 90, the total amount of rotation is limited by the given maximum angle.

Rotation settling speed This parameter determines how fast the accumulated rotation fades away. Analogously to the *Motion settling speed*, if it is 0%, the total accumulated rotation is added to the next frame, and every frame accumulates the whole rotation of the past. If the settling speed is 100%, no past rotation is added to the next frame. If the settling speed is intermediate between 0% and 100%, the amount of accumulated rotation is downscaled by (100 − settling amount) before being added to the next frame.

Motion noise level This parameter allows to compensate such undesirable behavior when motion tracker wildly springs away because some outlying area of a noisy picture becomes accidentally more "similar" to the reference picture than the right one. If noise level is set to 0% (its normal value), then it is not taken into account at all. When it is set to 100%, then similarity of the match block contents is totally ignored, and the current motion vector will be zero. An intermediate value between 0% and

100% defines the proportional noise level between minimum and maximum difference of all examined match block positions. From all such positions whose difference is found below noise level, that one is chosen which has the shortest translation relative to the reference frame.

Rotation noise level This parameter has similar meaning to the *Motion noise level*. It influences rotation tracking. If set to 0% (its normal value), it is not taken into account. The 100% setting completely suppresses rotation detection. An intermediate value between 0% and 100% defines the proportional noise level between minimum and maximum difference of all examined rotated match blocks from that of the reference picture. From all rotation angles whose match block differences are found below noise level, that one is chosen which has the minimum rotation angle relative to the reference frame.

Draw vectors When translation is enabled, 2 boxes are drawn on the frame. One box represents the translation block. Another box outside the translation block represents the extent of the *Translation search radius*. In the center of these boxes is an arrow showing the translation between the 2 master frames. When rotation is enabled, a single box the size of the rotation block is drawn rotated by the amount of rotation detected.

Two pass tracking Although this parameter can be enabled at any time, it is useful only while simultaneous tracking both translation and rotation, when the rotation amount is large. The normal tracking algorithm firstly examines translation, then rotation. However, comparison of a strongly tilted picture with the reference one can lead to rather inaccurate determination of the translation vector. The subsequent determination of rotation will be also inexact because of the preceding translation error. The second tracking pass, if enabled, refines translation and rotation.

If motion and/or rotation noise level is set above 0% and *Two pass tracking* is enabled, then both noise levels affect the first tracking pass only. During the second tracking pass of two pass tracking both noise levels are ignored.

It should be noted that two pass tracking is not twice slower than the usual single pass tracking. Actually it is only about 40% slower. As the second pass serves only for motion refinement, it starts from the reduced search range for block positions and rotation angles. And, as the first pass only yields an approximate position, which will be later refined during the second pass, it skips subpixel position search and examines rotation angles with the accuracy reduced to 1 degree.

Track single frame When this option is used the motion between a single starting frame and the frame currently under the insertion point is calculated. The starting frame is specified in the *Frame number* box. The motion calculated this way is taken as the absolute motion vector. The absolute motion vector for each frame replaces the absolute motion vector for the previous frame. *Settling speed* has no effect on it since it does not contain any previous motion vectors. Playback can start anywhere on the timeline since there is no dependence on previous results. We talk about *Keep shape* and it is the most precise way to calculate the motion vector; but it only

works well when the object to be traced does not change along the clip, remaining identical in shape, size and luminance.

Strong rotation of the picture can also have a bad impact on the tracking accuracy. If it is the case, accuracy can be significantly improved by enabling *Two pass tracking*.

Frame number The number of the reference frame used for motion tracking in the *Track single frame* mode is specified in this input field. Frame number is set as the 0-based absolute number starting from the very beginning of the whole timeline. Instead of manual calculating, it is possible to get *Frame number* directly from the current cursor position by pressing the *Get current* button.

Get current Pressing this handy button provides a convenient way to set the *Frame number* parameter to the value of the current cursor position.

Add (loaded) offset from tracked frame This toggle can help when the tracked object changes shape, leaves the screen area or becomes obscured by other objects somewhere in the middle of the *Motion* effect. Sometimes it can help also if a strong rotation causes the motion tracker to miss the right tracked object and wildly spring away.

To overcome this problem, one has to define two keyframes: the first one at the start of the *Motion* effect, the second shortly before the problematic location in the clip. At the first keyframe one defines the *Motion* effect parameters as usual, the *Add offset* toggle being switched off. At the second keyframe the match block can be redefined to cover another region of the picture which remains visible during the following part of the effect. The *Frame number* parameter should be set several frames before position of this second keyframe, and the *Add offset* toggle switched on.

After defining the two keyframes (even more can be added in the similar way) one switches at all the defined keyframes *Calculation* mode to *Save coords to tracking file*, *Action* mode to *Do Nothing*, rewinds the timeline cursor to the beginning of the *Motion* effect, and starts playback. The determined motion vectors will be written to the tracking file, the *Add offset* parameter having no effect during the *save* stage.

After playback finishes and all the relevant motion vectors are saved in the tracking file, one switches at all the defined keyframes *Calculation* mode to *Load coords from tracking file*, *Action* mode to whatever one needs, rewinds again to the beginning of the *Motion* effect and starts playback or rendering. After reaching the second keyframe, where *Add offset* was switched on, the program continues to follow the second object adding the total motion and rotation, accumulated so far.

Of course, the *Add offset* toggle can be used in the *Track single frame* mode only. In the other tracking modi such a technics is unnecessary.

Track previous frame Causes only the motion between the previous frame and the current frame to be calculated (*Follow shape*). This is added to an absolute motion vector to get the new motion from the start of the sequence to the current position. After every frame processed this way, the block position is shifted to always cover the same region of the image. Playback must be started from the start of the *Motion*

effect in order to accumulate all the necessary motion vectors. This method is less precise because you have error propagation between frames. However, it is essential when the object changes shape, size or luminance. Possible inaccuracies caused by rotation of the picture can partly be reduced by enabling *Two pass tracking*

Previous frame same block This is useful for stabilizing jerky camcorder footage. In this mode the motion between the previous frame and the current frame is calculated. Instead of adjusting the block position to reflect the new location of the image, like *Track Previous Frame* does, the block position is unchanged between each frame. Thus a new region is compared for each frame.

Master layer This determines the track which supplies the frames for the motion calculation. If it is *Bottom* the bottom track of all the tracks sharing this effect is the *master* layer. Then the top track of all the tracks is the *target* layer. And vice versa, if the *master* layer is *Top*, then the *target* layer is bottom.

Calculation This determines whether to calculate the motion at all and whether to save it to disk. If it is *Don't Calculate* the motion calculation is skipped. If it is *Recalculate* the motion calculation is performed every time each frame is rendered. If it is *Load coords from tracking file*, the motion vectors are loaded from a previous *save* calculation. If the previous *save* calculation does not provide the data for the currently processed frames, a new motion calculation is performed for that frames. If *Calculation* mode is *Save coords to tracking file* the motion vectors can be still loaded from disk if the previously calculated data exist. Otherwise, a new motion calculation is performed, and the calculated motion vectors are saved.

Tracking file This parameter determines the name of the file where the motion data are saved and/or loaded from. By default it is `/tmp/motion` but it can have any name and reside anywhere on the disk. *Tracking file* is a text file where each line contains four numbers: the absolute frame number (0-based integer), the X and Y translation displacements of subpixel precision (a subpixel is $\frac{1}{4}$ pixel in size, and to give integer values of X and Y these subpixel coords are multiplied by 4), and the rotation angle in degrees (as a floating point value). The user may have different tracking files for different timeline regions, it is possible to do any manipulations on them if necessary and even manually edit in a text editor.

Generate tracking file name This handy pushbutton suggests a tracking file name based on the name of the current asset. It helps to maintain more different tracking files with unique but predictable names used in the same project. When tracking file name is changed (either via the pushbutton or after entering it in the corresponding input field), the contents of that files (both the old and the new ones) remains unchanged.

Clear tracking file contents The values and meaning of the saved motion data depend on such a bunch of parameters, they can be even edited manually, so one can hardly predict whether the outdated motion vectors are to be reset or left untouched. Therefore the existing motion data are never altered implicitly (only newly calculated data may be added). And, should the user need to reset them here and right

now, it is possible to clear the given tracking file contents explicitly by pressing this pushbutton.

As protection against accident clearing tracking file contents, right before clearing the program ensures closing the current tracking file and renames it to the same name with the suffix `.old`. This way, the user can undo an erroneous clearing action by manual restoring the saved file.

Action Once the motion vector is known this determines whether to move the *target* layer opposing the motion vector or following the motion vector. If it is *Do Nothing* the *target* layer is untouched. If it is *Track...* the *target* layer is moved by the same amount as the *master* layer. This is useful for matching titles to objects in the frame. If it is *Stabilize...* the *target* layer is moved opposite to the motion vector. This is useful for stabilizing an object in the frame. The *Motion* operations can be accurate to single pixels or subpixels by changing the *Action* setting.

As motion tracking works consistently only in the *Play every frame* mode, the corresponding info on the current playback mode is shown in the bottom right edge of the effect dialog. The needed playback mode is defined in the Preferences dialog of CINELERRA-GG main window.

Secrets of motion tracking

Since it is a very slow effect, there is a method to applying the motion tracker to get the most out of it. First disable playback for the track to do motion tracking on. Then drop the effect on a region of video with some motion to track. Then rewind the insertion point to the start of the region. Set *Action* → *Do Nothing*; set *Calculation* → *Don't calculate*; enable *Draw vectors*. Then enable playback of the track to see the motion tracking areas.

Enable which of translation motion or rotation motion vectors you want to track. By watching the compositor window and adjusting the *Block X, Y* settings, center the block on the part of the image you want to track. It is advisable to choose elements that have evident edges in the x and y directions because the calculations are made on these coordinates. Then set *search radius*, *block size* and *block coordinates* for translation and rotation.

Once this is configured, set *Calculation* to *Save coords* and do test runs through the sequence to see if the motion tracker works and to save the motion vectors. Next, disable *Draw vectors*, set the motion *Action* to perform on the *target* layer and change *Calculation* to *Load coords*.

When using a single starting frame to calculate the motion of a sequence (*Keep Shape*), the starting frame should be a single frame with the least motion to any of the other frames. This is rarely frame 0. Usually it is a frame near the middle of the sequence. This way the search radius need only reach halfway to the full extent of the motion in the sequence. The suitable reference frame number can be defined conveniently via the *Get current* pushbutton.

Summarizing:

1. In the Master Layer create and save the tracking file:
    - *Draw vectors*: checked
    - *Action*: Do Nothing
    - *Calculation*: Save coords to Tracking File
2. Clone the track, which becomes the Target Layer. The *Motion* plugin will also be copied.
    - *Draw vectors*: unchecked
    - *Action*: Do Nothing
    - *Calculation*: Load coords from Tracking File
3. Add a third track to place the video of the object to be superimposed. Add the shared effects of the *Motion* plugin taken from the Master Layer.
    - *Draw vectors*: unchecked
    - *Action*: Track Subpixel
    - *Calculation*: Load coords from Tracking File

If the motion tracker is used on a render farm, *Save coords* and *Previous frame* mode will not work. The results of the *Save coords* operation are saved to the hard drives on the render nodes, not the master node. Future rendering operations on these nodes will process different frames and read the wrong coordinates from the node filesystems. The fact that render nodes only visualize a portion of the timeline also prevents *Previous frame* from working since it depends on calculating an absolute motion vector starting on frame 0.

2 step motion tracking

The method described above is *two-step motion tracking*. One step is used just to calculate the motion vectors. A second step is used to apply the motion vectors to the footage. This is faster than a single pass because errors in the motion vector calculation can be discovered quickly. This also allows the motion tracking to use a less demanding colormodel like *RGB888* in the scanning step and a more demanding colormodel like *RGB Float* in the action step. The scanning step takes much longer than action. This has the disadvantage of not being practical for extremely long sequences where some error is acceptable and the picture quality is low to begin with, like stabilizing camcorder footage.

The slower method is to calculate the motion vectors and apply them simultaneously. This method can use one track as the motion vector calculation track and another track as the target track for motion vector actions. This is useful for long sequences where some error is acceptable.

Pre-processing the shot

1. The *Motion* plugin uses *luminance* to do its own calculations, so we can edit the clip to enhance contrast and make it easier to calculate motion vectors. You can even create a copy of the monochrome clip and optimize it for the plugin. It lengthens the time but minimizes errors. The saved file can then be used for the original clip.

2. Correct lens distortion, especially if the object to be tracked moves to the edges of the frame.

3. Study the entire shot well: if necessary, divide it into many edits, each with its own *Motion* plugin. For example, if the object to be tracked leaves the frame or is covered by some other element or changes in shape, size or rotation. You can try to use the *Offset Tracking* technique described below.

Using blur to improve motion tracking

With extremely noisy or interlaced footage, applying a blur effect before the motion tracking can improve accuracy. Either save the motion vectors in a tracking pass and disable the blur for the action pass or apply the blur just to the *master* layer. You can even use a copy of the track formed only by the channels of Red + Green, because the channel of Blue is the noisiest. Another trick is to enlarge the Match Box to minimize the effect of noise.

Using histogram to improve motion tracking

A histogram is almost always applied before motion tracking to clamp out noise in the darker pixels. Either save the motion vectors in a tracking pass and disable the histogram for the action pass or apply the histogram just to the *master* layer. Finally, you can use the histogram to increase contrast.

Using two pass tracking to improve accuracy and stability

A common source of inaccuracy when tracking single frame is rotation of the picture. Ideally, the motion vectors should be determined by checking all possible rotation angles at every possible match block location. But such an algorithm would be unacceptably slow. Therefore the program finds for each frame the translation vector first, and then the rotation angle. A strong rotation can prevent motion tracker from finding accurate translation displacements. Then the subsequent rotation search will be also inaccurate because translation has been estimated not sufficiently well. In the *Two pass tracking* algorithm the first, coarse estimation of translation and rotation is followed by the second, refinement pass yielding more exact motion vectors.

Sometimes, particularly if the amount of rotation is large, the translation tracker can lose its right object and wildly spring away. Then the following strategy may help.

Firstly switch *Two pass tracking* off, *Track rotation* off, set *Motion noise level* to zero. Switch *Draw vectors* on, *Action: Do Nothing*, *Calculation: Save coords to tracking file*. Start play-

back and watch if the motion tracker has problems finding the right displacements (you can also examine the tracking file afterwards on the existence of discontinuities in coordinates). If there are such, try to increase *Motion noise level*, let's say, to 10% or 20%, *clear tracking file contents*, repeat playback and look if tracker does not spring away any more. A single pass translation tracking can be relatively fast, while tracking rotation is expensive.

When a sufficient motion noise level is found, switch *Track rotation* on, *Two pass tracking* on, *clear tracking file contents* and perform the complete translation + rotation tracking pass. As the refinement pass uses restricted search area and ignores the noise level parameters, the result should be more stable and accurate.

Possible sources of errors

Search radius too small: the traced object moves too fast with respect to the size of the search box set.

Search radius too large: The search box is so large that it also collects other similar items in the frame.

Occlusions: the traced object is temporarily hidden by some other element. *Offset tracking* or splitting of the video into several homogeneous clips is required.

Focus change: you may get errors if the object changes its focus. The video must be divided into several homogeneous clips.

Motion Blur: blurs the object making the calculation of the motion vector less precise. Very little can be done.

Shape change: you can use *Track previous frame* or the subdivision of the video in more homogeneous clips.

Lighting change: Contrast change can produce errors. *Track previous frame* or color correction can be used to return to the initial illumination.

Tracking stabilization in action

This is an explanation of how to stabilize a video as in the case of a video taken from a vehicle.

First select on the timeline the part of the footage you want to stabilize, using the in and out points. Then apply the *Motion* effect on that part of the video. Select the *Previous frame same block* option. That option is recommended for stabilizing jerky camcorder footage. Its goal is not to *follow* an object. The block stays exactly at the same place during all the effect length.

Enlarge the block and select almost half the size of the video. Select the *Stabilize subpixel* option as it will give a finer stabilization. Reduce the *Maximum absolute offset* value to limit the stabilization amplitude. You probably prefer to get a non-perfect stabilization

on some places on the video rather than having a very large black border on one side of the picture during big shakes. Set the *Translation search steps* value to 128. Increasing that value will not give a better result and only considerably increases the rendering time. Make sure the *Draw vectors* option is selected, and render the part of the video where the *Motion* effect is applied.

If the result is good, deselect the *Draw vectors* option so that the block and vectors are not drawn anymore on the video. Then, render your video to a `.dv` file, and import it into your project. You will notice the video is stabilized but there are black borders which appear on sides of the frame. You have to *zoom in* and define projector keyframes to move the projector around the screen, in order to remove those black borders. The more your footage is jerky, the more you have to zoom in to discard the black borders. That is why the result is better with HDV footage than with DV footage.

An interesting side note about *Add offset* usage is explained next[13]

To stabilize video, the *Motion* plugin uses a *tracking frame* to which to track to and a region within that frame to track (generally an object in the background) in the current frame. When the region is obscured, often by something in the foreground or by leaving the screen, then the motion compensation would fail, and the video jumps all over the place.

You set a second region to track, and then *add offset*.

This shows how it is used. It works very well:

```
------k-------k-----------------
^       ^^
A       BC

A - object1 is visible in the background up until C
B - (the frame before C) has both object1 and object2 visible
C - has only object2 visible
```

1. Make a keyframe and set to track object1.
2. Make a keyframe at C and track frame at B, set to track object2.
3. Set keyframe at C to add offsets that were calculated at B.

Tips

1. Motion plugin[14] algorithm in CINELERRA-GG is rather difficult to understand (the most efficient way to understand it is to look inside its C++ code:). Moreover, it has

[13] credit Pierre Marc Dumuid
[14] Thanks to Georgy

some tricks that, although can be derived knowing the details of the algorithm, are absolutely non-intuitive by themselves. Moreover, it is rather slow, and the results cannot be visible immediately while altering the settings.

2. The Motion plugin *cache* (or *motion vector* or *tracking file*) sometimes allows to play with very sophisticated techniques. For example, by editing the cache, perhaps with some custom script, you can induce some kind of motion which would be difficult to make in other way. But in the same time the existence of the cache file can trigger user's mistakes. So you must always pay attention, which cache file may be used in the moment. // Motion's cache data are absolute-frame-number based. If you insert some small piece of video before that where the plugin was attached, the frame numbers get displaced relative to the cached numbers, and the result may become incorrectly stabilized.

3. The motion vector is a text file located in `/tmp`. We can open it with a plain editor and modify the wrong $X\ Y$ coordinates, i.e. those that deviate from the linearity, to correct the errors that always happen when we perform a motion tracking (jumps). It can be a long and tedious job, but it leads to good results.

4. It is not possible to switch Motion cache off. While working, *Motion* always looks in its cache. If it contains some data assigned to the current frame numbers, that cached data will be used unconditionally. If not, that data will be calculated, stored in the cache, and later reused, also unconditionally.

5. In the Motion plugin dialog there is a button `Clear tracking file contents`. When in doubt, you can press this button, the cached data will be erased and fresh recalculated on the next pass.

6. It is quite usual to wish independent caches for different parts of video. It is possible to set such distinct cache file names (either manually, or let software generate them with `Generate tracking file name`). When you generate new names, the plugin automatically switches Calculation off, just for security. You switch it on again when needed.

7. If `play track` is switched off for the track where the Motion plugin was attached, the plugin will calculate nothing. Also, if `Play every frame` was not active in `Preferences`, some video frames can get skipped from processing.

8. You can try tracking using reverse playback of the track. Sometimes it may lead to a better calculation.

10.10.55 Motion 2 Point

Motion plugin using 2 *Match box* for tracking. For theory and explanations refer to the Motion plugin.

10.10.56 Motion Blur

Uses X/Y camera automation vectors to apply a linear blur trailing camera direction due to movement.

Length distance between original image and final blur step; corresponds to the distance of the fields.

Steps number of blur steps to be used in the calculation. Increasing the number takes more CPU.

10.10.57 Oil painting

This effect makes video tracks appears as a painting. It can be controlled by *Radius* slider and *intensity of colors* can be chosen as an option.

10.10.58 Overlay (Video)

This effect can combine several tracks by using the Overlayer. This is a basic internal device normally used by CINELERRA-GG INFINITY to create the dissolve transitions and for compositing the final output of every track onto the output bitmap. The Overlayer has the ability to combine one or several image layers on top of a "bottom layer". It can do this combining of images in several different (and switchable) output modes such as *Normal*, *Addition*, *Subtract*, *Multiply* (Filter), *Divide*, *Max* and *Replace*. For a detailed list and some differences in output expectations, refer to the Overlays chapter – PorterDuff.

The *overlay* plugin enables the use of this Overlayer device in the middle of any plugin stack, opening endless filtering and processing possibilities. It is only useful as a *shared plugin* (i.e. a multitrack plugin). To use the overlay plugin:

1. Add the effect to Track A.
2. Choose *attach effect* from the context menu of another track (Track B).
3. Choose Track A: Overlay as a shared plugin.
4. Manipulate the plugin parameters in Track A.

In the Overlay plugin's parameter window you can choose the overlay "Layer order" to select which track plays the role of the "bottom layer" and which plays the role of the "top layer". For some overlay modes this can make quite a difference, for example the bottom layer is subtracted from the top layer for "Subtract" mode. Also in the parameter window, you can choose on which of the tracks to overlay the combined output by changing the "Output layer" and it too can make quite a difference. (Hint: in most cases, you will want to mute the other track so you only see the combined output).

10.10.59 Perspective

The *perspective* plugin (aka Corner Pinning) allows you to change the perspective of an object and is used to make objects appear as if they are fading into the distance. Basically, you can get a different view. A transformation is used which preserves points, lines, and planes as well as ratios of distances between points lying on a straight line.

In (figure 10.58) you can see that there are four options for the endpoints used for the edges.

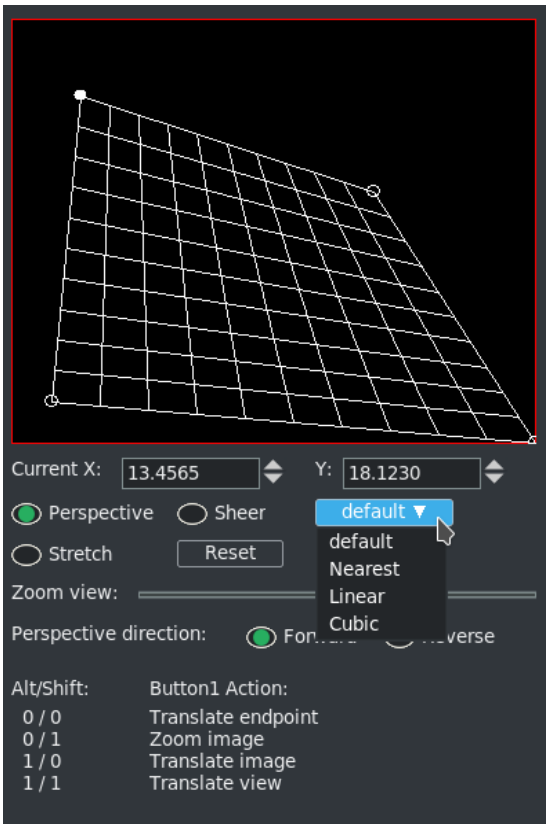


Figure 10.58: *perspective control window*

Default if OpenGL is being used with your graphics card, this will be the option in effect. If no OpenGL, then it will be Cubic.

Nearest using software, nearest neighbor can look step-py.

Linear software implementation of a linear algorithm.

Cubic smoothest looking on the edges and considered the best.

Key Presses for using the Perspective plugin:

Left mouse button	drags the corner that is closest to current location
Alt key + left mouse	translates the perspective; drags the whole image
Shift key + left mouse	zooms the perspective
Alt+Shift + left mouse	translates view but does not change output

Note that the red color lines in the box show the composer boundary.

In order to see endpoints that go off the screen, you can use the zoom slider which changes only the zoom view and does nothing else. The slider uses a logarithmic scale ranging from $\frac{1}{100} to 100$. Although not shown in the image here, each endpoint is labeled in yellow with 1-4 and the Current X endpoint is shown in the menu to make it easier to tell which point is in play.

Figure 10.59 show the results of the 4 different smoothing options.

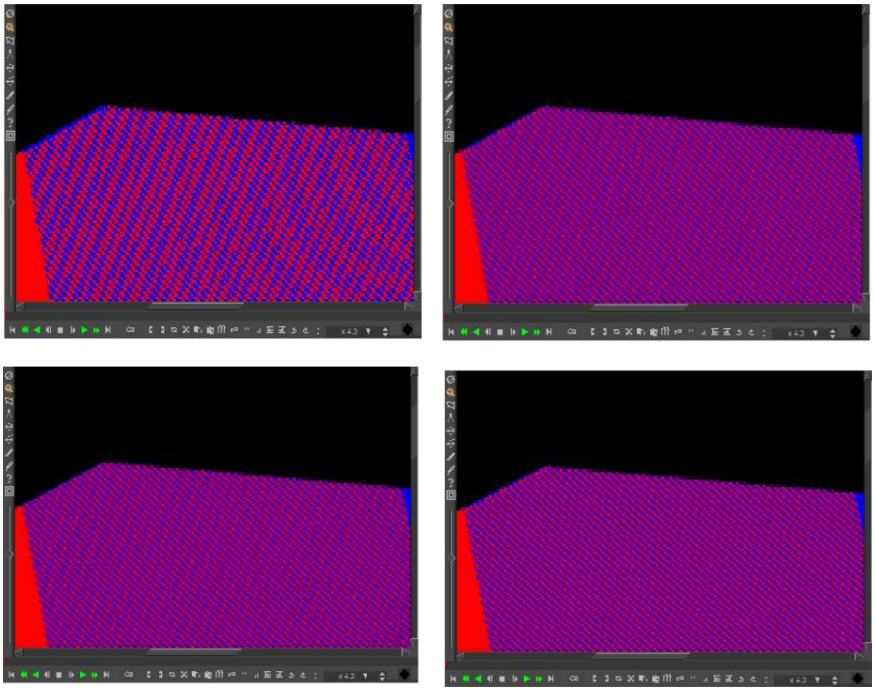


Figure 10.59: *Clockwise: Nearest; Linear; OpenGL and Cubic*

Just a side note about the possibility of ending up with a divide by 0 case when manipulating the endpoints. Sometimes this results in a single frame anomaly but you can workaround this by setting the X value for the middle keyframe of a series to just a little more than 0, such as .01 so that dividing by 0 is avoided. The actual cause of the problem is that the interpolated X1..X4 auto coordinates used by perspective eventually have x1==x2, x3==x4, so that all points x are scaled by zero. Another solution is to tweak the frame count to an even number, so that the center is not over the point where x1==x2, x3==x4, or perturb the midpoint position a little so that the answers are not exactly zero.

10.10.60 Polar

The *Polar* effect bends and warps your video in weird ways. Mathematically, it converts your video from either *polar* coordinates to *rectangular* coordinates, or the reverse. For maximum precision you can use the input box. With the *Clear* buttons we can bring the slider to default values without affecting the other parameters.

10.10.61 Posterize

Reduces the color depth so as to decrease the displayed color gradients, up to the bands typical of posterization. It is used by setting the number of *steps* from 0 to 255.

10.10.62 RGB-601

For analog video or MPEG (including DVD) output, the maximum range for R,G,B is $[16, 235]$ $(8\,bit)$. For YUV, the maximum range for intensity (Y) is $[16, 235]$ $(8\,bit)$. This range corresponds to gray levels from 6% to 92%. When rendering, values outside of these ranges will be clipped to these limits.

To render to MPEG, add the *RGB-601* effect to all video tracks where material uses the full intensity scale (0—100%), and enable *RGB→601* compression. Consider adding the *Videoscope* effect after RGB-601 to see how RGB-601 affects your dynamic range. To preview how your rendered MPEG would look without *RGB→601* compression, instead enable *601→RGB* expansion and you will observe a noticeable contrast increase. Although RGB-601 will reduce contrast in your video tracks, the contrast will be restored during MPEG playback.

10.10.63 RGBShift

Most cameras take the light coming into the lens, and convert that into 3 sets of numbers, one for Red (R), one for Green (G), and one for Blue (B). Figure 10.60 showing RGB shift before and after.

Some of the older cameras were composed of 3 sensors and originally the RGB sensors were on 3 separate planes and had to be aligned. If they were misaligned in the video, you can use *RGBShift* to get them realigned. To move a specific color up/down, modify the *dy* value using the slider bar in the RGBShift window. To move a color left/right, modify the corresponding *dx* value. For maximum precision you can use the input box. *Clear* buttons are present to reset its slider to default without affecting others. Note that the current values of the RGBShift are maintained in the `.bcast5` defaults file and will be retained across sessions. If using the YUV color space, you will want to use *YUVShift* instead.

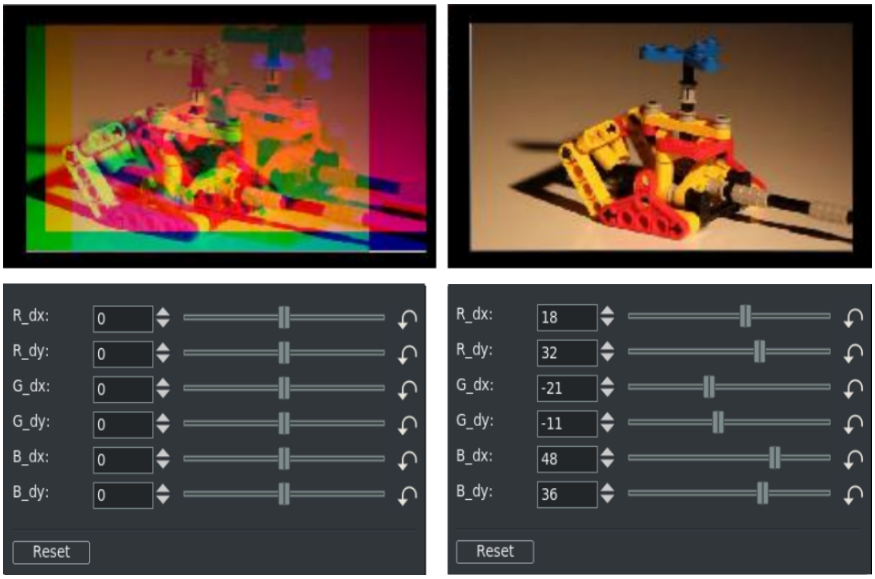


Figure 10.60: *Bad Misaligned color and after color aligned*

10.10.64 Radial Blur

Radial blur is a *Bokeh* effect that creates a whirlpool which simulates a swirling camera. You can vary the location, type, and quality of the blur.

Figure 10.61: *For clarity of presentation only 2 fields are shown*

X,Y center of the circle of movement.

Angle angle of motion in one direction.

Steps number of blur steps to be used in the calculation; increasing this number uses more CPU.

Clear to reset its slider to default without affecting others.

For maximum precision you can use the input box. Figure 10.61 has the parameters: $Angle = -35$ and $Steps = 2$.

10.10.65 ReframeRT

ReframeRT changes the number of frames in a sequence of video directly from the timeline. The faster method for getting the same results as this plugin is to use the *speed curve* which was a later addition. But if you need very precise results, *ReframeRT* is most useful. There are two ways to do this, which can be selected from the checkboxes in the configuration GUI. The first *Stretch* mode changes the number of frames in the sequence, and therefore its length, but not the framerate. The *Downsample* mode instead keeps the length of the movie by varying the framerate. It is important to understand that the plugin works by varying the frames, the possible change of *fps* is only a side effect of the creation of new frames due to interpolation. The interpolation algorithm is simply the slope of a linear curve. This plugin is keyframable and the *Interpolate* option works between keyframes. A simpler and more handy version is the *Speed PerCent* plugin.

Stretch

Stretch mode multiplies the current frame number of its output by the *scale factor* to arrive at the frame to read from its input. The scaling factor is not entered directly but using a number of *input* frames to be divided by the number of *output* frames.

```
Scale factor (SF) = Input frames / Output frames
```

$$\frac{1}{8} \Rightarrow scale\,factor = 0.125 \qquad (slow\,motion)$$

For slow motion we leave 1 for the frames of the input and we increase the number of frames of the output (for example putting 8 for the output we have slow motion 8×, with $SF = \frac{1}{8} = 0.125$). For fast motion we leave 1 for the output and we increase the number for the input (for example 8 to have 8×, with $SF = \frac{8}{1} = 8$). Another possibility is to put the frame rate of the media (e.g 120 fps) in the input and the project frame rate in the output (e.g 30 fps) or the opposite.

The stretch mode has the effect of changing the length of output video by the inverse of the scale factor. If the scale factor is greater than 1, the output will end before the end of the sequence on the timeline. If it is less than 1, the output will end after the end of the sequence on the timeline. The ReframeRT effect must be lengthened to the necessary length to accommodate the scale factor. Change the length of the effect by clicking on the endpoint of the effect and dragging.

Although stretch mode changes the number of the frames read from its input, it does not change the framerate of the input. Effects before ReframeRT assume the same frame rate as ReframeRT. In stretch mode to create a fast play effect enter a value greater than 1 to

get accelerated playback. For a slow motion effect, use ReframeRT in stretch mode with a value less than 1.

Example: you have a clip that you want to put in slow motion. The clip starts at $33.792 seconds$ and ends at 39.765. The clip is $5.973 seconds$ long. You want to play it at $\frac{4}{10}^{ths}$ normal speed. You divide the clip length by the playback speed ($5.973 \div 0.4$) to get a final clip length of $14.9325 seconds$. You create an in point at the start of your clip: $33.792 seconds$. You put an out point $14.9325 seconds$ later, at $48.7245 seconds$ ($33.792 + 14.9325$). You attach a `ReframeRT` effect, set it to 0.4 and stretch. You change the out point at 48.7245 to an in point. You start your next clip after the slow motion effect at the 48.7245 out point. You can do this without making any calculations by first applying the effect and then lengthening or shortening the bar to where the stretched movie ends.

Now in the timeline we have the affected part of the plugin where we see the slow/fast effect, and the continuation of the timeline from where the plugin ends. We then have to select the interval on which the plugin acts and render it or transform it into a nested clip or nested asset. In this way we can replace or overlap it with the part of the timeline that originally included all of the part we wanted to slow down/speed up. See also the *Reframe* render effect for direct rendering.

Downsample

Downsample mode does not change the length of the output sequence. It multiplies the frame rate of the output by the scale factor to arrive at a frame rate to read the input. This has the effect of replicating the input frames so that they only change at the scaled frame rate when sent to the output. It does not change the length of the sequence. If the scale factor is 0.5 and the output frame rate is $30 fps$, only 15 frames will be shown per second and the input will be read at $15 fps$. Downsample is only useful for scalefactors below 1, hence the name downsample.

Downsample mode changes the frame rate of the input as well as the number of the frame to read, so effects before ReframeRT see the $framerate \times scalefactor$ as their frame rate. If the scale factor is 2 and the output frame rate is 30, the input frame rate will be 60 and the input frame number will by doubled. This will not normally do anything, but some input effects may behave differently at the higher frame rate.

Other important points

- ReframeRT uses the fps indicated in `Settings` → `Format` → `fps` project and not the `fps` of the assets.
- It can be associated with Nested Clips.
- As an alternative to ReframeRT you can use the *speed curve*, or change the framerate in `Resources` → `info` and in the `Project`; or using *Reframe* render effect, or using *Speed PerCent* plugin.
- It is keyframmable.

- ResampleRT with the same settings is used to act on audio tracks.

10.10.66 Reroute

The *Reroute* plugin[15] enables you to selectively transfer the Alpha channel or the Components (RGB or YUV) or both from a *source* track to a *target* track, partially overwriting the target's contents.

Target Track: you can choose between *Top* and *Bottom*.

Operation: you can choose between *Replace Target*, *Components Only* and *Alpha Replace*.

- *replace Target* (component + alpha channels), this allows you to have the ability to overwrite everything on the target track with the source track where the the plugins are inserted.
- *Components only* (color channels only), replace the RGB or YUV components of the target track by the RGB or YUV components of the source track. The Alpha of the target track does not change.
- *Alpha replace* (alpha channel only), only the Alpha of the target track is changed but the components are not. This makes it easy to create an alphs channel on a track and then use that to apply it easily to another track.

It works as a shared plugin. The typical usage scenario is to build up a possibly animated Mask in one track and then to transfer the Alpha channel to another target track.

10.10.67 Reverse video

Media can be reversed on the timeline in realtime. This is not to be confused with using the reverse playback on the transport panel. The `reverse` effects reverse the region covered by the effect regardless of the transport direction. The region to be reversed is first determined by what part of the track the effect is under and second by the locations of keyframes in the effect. The reverse effects have an enabled option which allows you to set keyframes. This allows many possibilities.

Every enabled keyframe is treated as the start of a new reversed region and the end of a previous reversed region. Several enabled keyframes in succession yield several regions reversed independent of each other. An enabled keyframe followed by a disabled keyframe yields one reversed region followed by a forward region.

10.10.68 Rotate

The *Rotate* filter can rotate the video in 90 degree increments or by any number of degrees through use of the *wheel* and about any *pivot point*. It can also reverse and flip the video.

[15] By Hermann Vosseler

10.10.69 Rumbler

The *Rumbler* plugin can be used to create dream-like or earthquake-like noise in the video. It creates noise by jiggling the corners through use of perspective transformation at the corners. The algorithm used is:

$$Rumbler\,(value) = (value\,at\,time) + amplitude \times (random\,generator)$$

The random generator varies from $-0.5\,to\,0.5$. The rumble perturbs the normal values at time points which occur rate times a second. The values used between the rumble points are interpolated, so that the value jiggles rate times a second, by as much as the rumble amplitude. The time unit is frames per second. The corners are in units of percent width/height (figure 10.62).

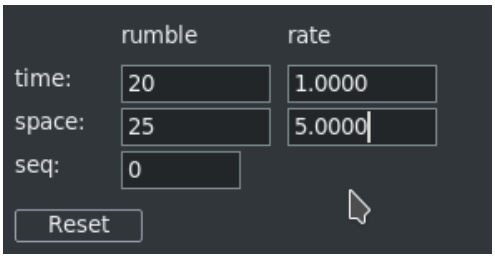


Figure 10.62: *Rumbler control window*

Screencast shows:

Time jittering - 20 *f ps* 1 time a second. *Corners jittering* - 25% 5 times a second. Using *random seed* 0 for a rumble sequence.

Plugin variables:

Rumbler: gain applied to random rumbler.

Rate: number of times per second.

Time: number of frames per rate times a second; if 0 has no effect on results.

Space: corners jiggling in percentage per rate times a second; if 0 has no effect on results.

Seq: is a random seed number; any reasonable positive or negative number you choose.

10.10.70 SVG via Inkscape

This plugin allows the user to manipulate an SVG (scalable vector graphics) image with *Inkscape* without having to leave the program. The associated CINELERRA-GG window provides the ability to change the DPI, the Out x/y coordinates, and the Out w/h values.

For more information on use of inkscape, refer to: https://inkscape.org/develop/about-svg/

DPI is Dots per inch and it represents the resolution of the SVG image. Since the image is scaled with interpolation mode linear, the edges will look blurry when the input resolution is lower than the output resolution. You can either set the desired *DPI value* in the window or use the tumbler on the integer text box, then use the *update dpi* button to have the change take effect. Changing DPI causes the entire image to be re-exported via inkscape. DPI changes cause adjustments in the resolution, speed of re-import, and storage needed for the image data.

Out_x/Out_y allow for changing the location of the SVG via the *x* or *y* coordinates.

Out_w/Out_h The scaling is controlled by width and height as they are normal parameters to overlay.

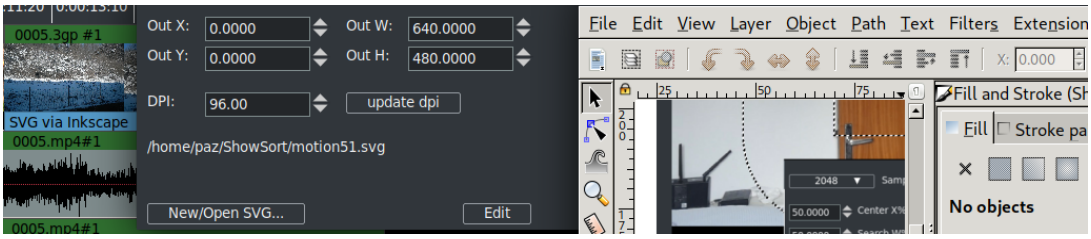


Figure 10.63: *Control window and Inkscape*

Figure 10.63 shows the menu options plugin window and the SVG image in the Inkscape window.

10.10.71 Scale

Reduce or expand the image size depending on the ratio.

Size Height and Width in pixel; plus pulldown menu with preset.

Scale Aspect ratio

Constrain ratio Lock height to width rate

10.10.72 Scale Ratio

With the Scale Ratio plugin you can manipulate your video to maintain the pixel aspect ratio (proportional geometry), possibly for a different output Display device (figure 10.64).

In R and *Out R* representing the current input and output aspect ratios. Use the arrows to change to your desired values. Next you have the *In W/H* and the *Out W/H* for Width

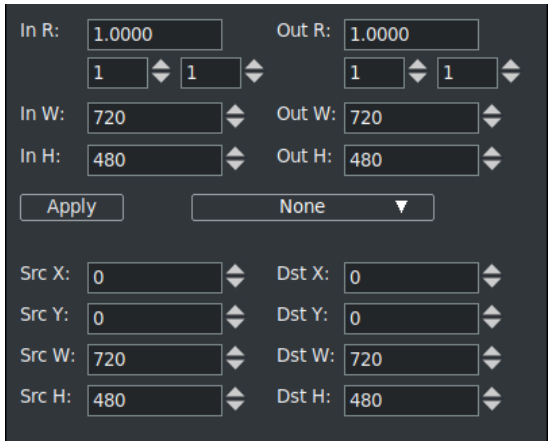


Figure 10.64: *Many parameters of scale ratio*

and Height. In the middle of the plugin on the right-hand side, you can set the Scale type of *None*, *Scaled*, *Cropped*, *Filled*, *Horiz edge* and *Vert edge*. The top part (aspect ratio data) is used to compute the bottom part when the Apply button is pressed. The bottom part allows you to reposition the image input or output to customize the results. When the in/out aspect ratios are different, the output must be cropped or filled to fit the output and maintain pixel square appearance. Left and right sides of the bottom portion show the *Source* and the *Destination X, Y, W, H* values. As you change the values on the left side, you can see how this will affect the output as you observe the results in the compositor window. For example, as you change the values for *SrcY* in a *cropped* Scale scenario, you see up/down movement.

10.10.73 Selective Temporal Averaging

This plugin is designed to smooth out non-moving areas of a video clip (figure 10.65).

Denoise is generally done on a spatial basis, mediating the values of a group of adjacent pixels to achieve greater uniformity. The effectiveness of Denoise can be increased by also introducing a *time average* between a group of successive frames. The union of these two phases is the basis of the plugin. In fact the smoothing is performed by averaging the color component for each pixel across a number of frames. The smoothed value is used if both the standard deviation and the difference between the current component value and the average component value are below a threshold. The standard deviation is a mathematical index used to estimate the variance of a group of pixels: at high values corresponds more variation of the pixels and therefore more noise.

The *Selective Temporal Averaging* plugin plays on the homogenization of the values of a group of pixels in a group of frames, based on a threshold below which the original values are left and above which the average is performed and then the noise reduction.

The *average* and *standard deviation* are calculated for each of the components of the video.

The type of components averaged is determined by the color model of the entire project. The average and standard deviation of the frames can be examined by selecting the specific radio button in the plugin options window. The region over which the frames are averaged is determined by either a *fixed offset* or a *restart marker system*. In a restart marker system, certain keyframes are marked as beginning of sections. Then for each section, the frames surrounding the current frame are used as the frames to average over, except when approaching the beginning and end of a section, where the averaging is performed over the *N* beginning or ending frames respectively.

An example of common usage is to select the number of frames you wish to average.

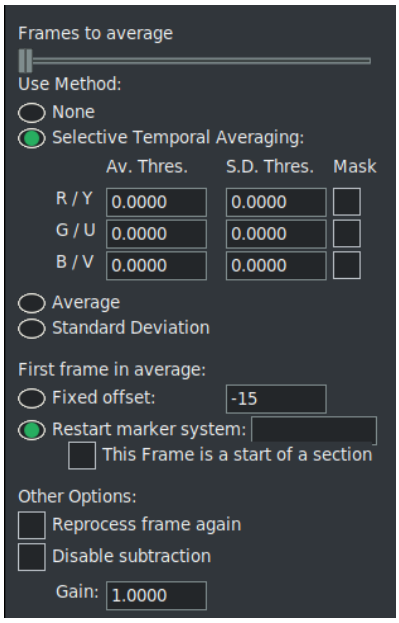


Figure 10.65: *STA control window*

1. Enter a reasonable number of frames to average (for example, 10).
2. Select the *Selective Temporal Averaging* method and enter 1 and 10 for all the *Av. Thres.* and *S.D. Thres.* respectively. This basically causes all pixels to use the average value.
3. Turn the *mask* for the first component on. This should make the whole frame have a solid color of that specific component.
4. Slowly reduce the *S.D. Thres.* value. As you do so, you will notice that the regions vastly different from the average will have a flipped mask state. Continue to reduce the threshold until you reach the point at which non-moving regions of the video have a flipped masked state. This value is known as the *noise-floor* and is the level of natural noise generated by the CCD in the camera.
5. Repeat the same procedure for the *Av. Thres.*
6. Turn off the *mask*.
7. Repeat this for all channels.

10.10.74 Sharpen

Sharpen the video, either the *luminance*, *horizontal*, or *interlace*. You can use sliders or, for maximum precision, you can use the input box. With the *Clear* buttons we can bring the slider to default values without affecting the other parameters.

10.10.75 Shift Interlace

Shift the interlace lines using *odd* or *even*. With the Clear buttons we can bring the slider to default values without affecting the other parameters.

10.10.76 Sketcher

Now you can sketch *lines*, *curves* or *shapes*, on your video in different colors using various pen widths and pen type with the `sketcher` plugin. You can even fill them. You can do rotations and scaling and apply two anti-aliasing modes; single and double. Getting started is fairly easy – simply hold down the *shift key* while using the *left mouse button* to create a bunch of points in the compositor window. However, to do more than that you will need to understand the buttons and options or you may end up with unexpected results.

(figure 10.66) shows the Sketcher gui and the sketch lines/curves created in the Compositor.

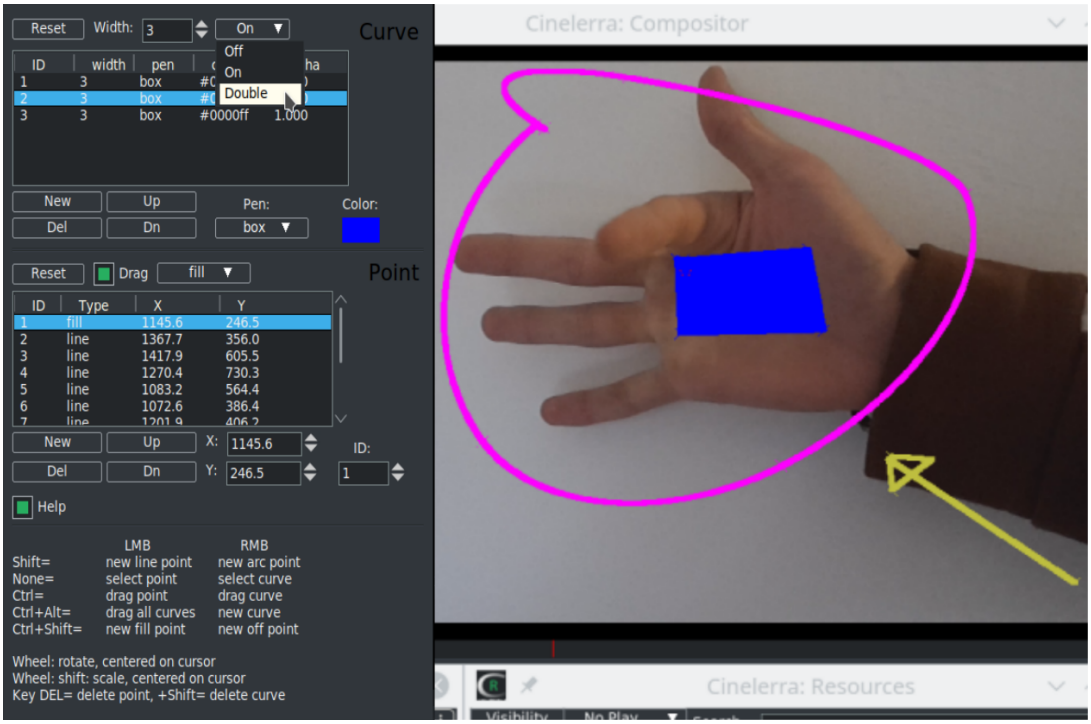


Figure 10.66: *Sketcher control window and sketch on Compositor*

In the screencast, note the Sketcher window gui *Curve top section* and the *Point bottom section*. The pink circle sketch is *id* #1 in the curve section. Since *id* #2 is highlighted in the Curve section, the X/Y coordinates in the Point section below show the points used to create the blue shape. Point 6 is selected so we see a red crosshair (*Drag point*) inside the rectangle. The yellow arrow on the image is *id* #3 curve. Finally the cursor shows Double anti-aliasing.

Some basic rules

Uncheck *Drag* before rendering; if you do not uncheck, the drag points will show in your output. Drag must be checked on to edit the data. Drag must be checked off to use *Click to play*. If the drag button flickers when clicked then another window has drag focus. Unfocus it first. Turn drag off to see what the sketcher figure will look like when rendered.

Curves can be any number of points or just a single point. You must create a *new* curve if you want a new string of points not connected to the current curve. Highlight the curve *id #* for a specific curve to modify its values. Left mouse click or right mouse click on an existing point on the highlighted line/curve *id #* will automatically highlight the selected point in the Point section of the gui and turn red in the image. We can drag this point to the desired position using `CTRL + LMB`; or we can translate the entire selected curve using `CTRL + RMB`.

With the mouse wheel we can rotate our curves; the center of rotation is given by the position of the cursor. Normally the scroll is slow to be more precise. If you make a fast scroll you have an acceleration that allows you to make complete rotations in a short time. Using the mouse wheel with the SHIFT key we can instead scale the size of the curve; the fixed scaling point is still given by the mouse position.

There will always be 1 empty curve automatically defined when you start or even reset. You can not delete the empty default curve but you can use it for a curve.

LMB click almost anywhere on the compositor screen will automatically show, in the currently highlighted curve, the closest point by turning it red.

There is no *undo* recorded between gui updates. It is recommended that you use the option "b" to save a backup if you get to a place where you want to make sure you do not lose your sketch.

Table 10.1: *Sketcher controls*

Compositor: Mouse usage	**Compositor: Action**	**Plugin GUI**
shift+left mouse button	create a new line point	New button in Point (line)
shift+right mouse button	create a new arc (curve) point	New button in Point (curve)
left mouse button	select a single line point	Click point listbox item
right mouse button	select a single arc (curve) point	Click curve listbox item
ctrl+left mouse button	drag point	
ctrl+right mouse button	drag curve	
alt+left mouse button	drag all of the curves together	

alt+right mouse button	create a new curve	New button in Curve section
ctrl+shift+left mouse button	create a new fill point	New button in Point (fill)
ctrl+shift+right mouse button	create a new off point	New button in Point (off)
Wheel mouse	rotate curve	
Wheel mouse + shift	scale curve	
delete key	deletes highlighted points	Del button in Point section
delete key+shift	deletes highlighted curves	Del button in Curve section

Other Button and Label Descriptions

Anti-Aliasing You can set Off; On or Double. Default is On.

Color refers to the current curve pen color. Click on the *Color rectangle* to bring up the Color window to change any of the color values, including alpha.

Drag check to create curves and to be able to see any Off type curves. Uncheck for render/viewing. If you need to clear the Drag enabled, you can easily do this with *Allow keyframe spanning* whose use is described in Allow Keyframe Spanning.

Reset – in the *Curve section*, all of the curve lines will be deleted except for an empty default curve; – in the *Point section*, all points for the highlighted curve in the Curve section will be deleted.

Width integer width of line; width of 1 is the default; width of 0 is the smallest size of 1 pixel.

ID is the label number of the curve or point.

Pen value can be set with the pulldown to *box* (a square), *+*, */*, *x* or *off*. The default is *x*.

Alpha alpha value as set in the Color window. Default is 1 which is totally opaque.

Arc Type value can be set with this pulldown, which is to the right of the `Drag` checkbox, to either *off*, *line*, *curve*, or *fill*. *Off* is the default. If the line type is off when you uncheck drag, you will no longer see the line/curve. *Fill* specifies a point be inside the region to be filled with the chosen color. Multiple fill points can be used within a single curve.

Up moves the highlighted Curve or Point Up in the stacking order. Multiple points, but not multiple curves, can be highlighted and moved.

Dn moves the highlighted Curve or Point(s) down in the stacking order.

X is the point's *X* coordinate.

Y is the point's *Y* coordinate.

ID number detailed information

Points and curves are identified by numeric ids, and not the table position. This is so that deleting and inserting points/curves does not shift the interpretation of which points are associated for interpolation. For purposes of this discussion, *prev* refers to *previous*. The prev keyframe constitutes the draw list of curves and points. The prev ids are used to access the next keyframe interpolation input data. When *next* exists for prev.id the interpolation functions apply. When next does not exist, prev is returned for the interpolation value. The keyframes may be any id sequence; both the prev and next keyframe id sequences are arbitrary non-unique sets. This means that if an id is specified redundantly, it causes a redundant draw when it is the prev keyframe, and only the last item with the id in the next keyframe is used for interpolation.

Fill detailed information

The *fill* point is a marker point, and not really part of the curve. That point identifies the inside of the loop. It can also be used to identify the outside of the loop in order to fill that. The loop is created by drawing a line from the last point to the first point in the curve. The line of this last segment is from the last point of the curve. If there are isolated loops in the curve (it is self intersecting) then you will be able to use multiple fill points to fill these regions.

10.10.77 Speed PerCent

It works like the `ReframeRT` plugin in *Stretch* mode. The interpolation algorithm is simply the slope of a linear curve. It allows you to change the speed of the video track, or an edit, or a region of the track quickly and intuitively. There are presets to set the speed to 100% (default) or 25%; 50%; 200% and 400%. Or we can enter the value directly in the text box or use the slider until the desired value is reached. *Interpolate* option works between keyframes. It requires setting the right limits for applying the effect and then rendering that region, or turning it into a nested clip, which will replace the original one.

10.10.78 Sphere Cam

Converts multiple fisheye images into a panoramic projection.

10.10.79 Swap Frames

Swap the position of two consecutive frames. There are two modes of exchange:

swap: 0-1, 2-3, 4-5... or *swap: 1-2, 3-4, 5-6...*

10.10.80 Swap channels

Swap color, R G B, and Alpha channels with another channel or even turn channels off completely.

10.10.81 Threshold

Threshold converts the image to pure luminance, and replaces pixels with one of three colors based on the luminance. Pixels with luminance values in the low range are replaced with black, pixels in the middle range are replaced with white, and pixels in the high range are replaced with black. *Color* and *alpha* (transparency) for each range are configurable with the buttons and interpolate according to keyframes.

The threshold window shows a *histogram* of luminance values for the current frame. Click dragging inside the histogram creates a range (blue) to convert to white. SHIFT-clicking extends one border of this range. Values for the threshold range can also be specified in the *text boxes*.

This effect is basically a primitive *luminance key* to produce a *matte*. A second track above the track with the threshold effect can be *multiplied*, resulting in only the parts of the second track within the threshold being displayed.

10.10.82 Time average

Time average is one effect which has many uses besides creating trail patterns of moving objects (figure 10.67). The main use is reducing noise in still images (or in the motionless parts of a video). Merely point a video camera at a stationary subject for 30 frames, capture the frames, and average them using time average and you will have a high quality print. In floating point color models, time average can increase the dynamic range of low quality cameras.

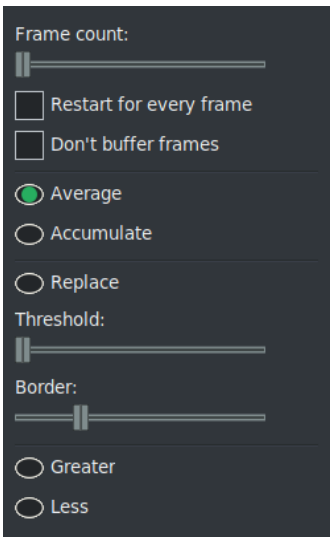


Figure 10.67: *GUI for Time Average*

Inside the time average effect is an accumulation buffer and a divisor. A number of frames are accumulated in the *accumulation* buffer and divided by the divisor to get the average (for 10 accumulated frames the divisor is 10). Because the time average can consume large

amounts of memory, it is best applied by first disabling playback for the track, dropping the time average in it, configuring time average for the desired number of frames, and re-enabling playback for the track.

Frames count this determines the number of frames to be accumulated in the accumulation buffer. Ranges from $1 to 1024$ frames.

Accumulate this outputs the accumulation buffer without dividing it.

Average this causes the accumulation buffer to be divided before being output. This results is the average of all the frames. The result is similar to *Selective Temporal Averaging*, but not configurable.

Replace (*Threshold*, *Border*): causes the accumulation buffer to be replaced by any pixels which are not transparent. In combination with motion tracking it allows entire sequences to be combined to form panoramas.

Restart for every frames if an effect before the time average is adjusted the time average normally does not reread the accumulation buffer to get the change. This forces it to reread the accumulation buffer when other effects change.

Don't buffer frames in order to represent the accumulation of only the specified number of frames, the time average retains all the previous frames in memory and subtracts them out as it plays forward. It would run out of memory if it had to accumulate thousands of frames. By disabling subtraction the previous frames are not stored in memory and only the average function is affected by the frame count.

10.10.83 TimeBlur

The TimeBlur plugin provides a way to blur a number of *Frames*.

10.10.84 Timefront

Space-temporal warping enables time to flow differently at different locations in the video (figure 10.68). This plugin divides the frame into segments (bands) whose type, position and size can be configured at will. Each segment will move with a different speed from the others, greater than the band that precedes it and less than the band that follows.

Type allows you to choose a *Linear* or *Radial* segmentation; or to make it depend on the alpha channel or on another track (with *Shared Effect*). In the case of Linear you can choose the orientation of the strips (*Angle*); in the case of Radial you can choose the coordinates (*X,Y*) of the center. For a more precise adjustment we can make the bands visible with the button *Show Grayscale* (for tuning).

With *Inner* and *Outer Radius* we can position the beginning and end of the bands in the frame (similar to the `Gradient` plugin), so you can choose the spatial range in which to apply the effect.

With *Time Range* we decide the size (and therefore the number) of the bands.

Rate allows you to choose the type of algorithm to use when switching between the start and end bands. You can reverse the direction with the *Invers* button. The three modes are: *Linear*, *Log* or *Quadratic* (exponential). This is a warping framework plugin based on this article: http://www.vision.huji.ac.il/videowarping/HUJI-CSE-LTR-2005-10_etf-tr.pdf

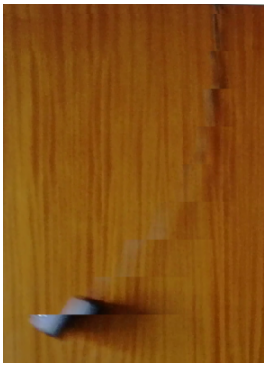


Figure 10.68: *Temporal bands for Timefront*

10.10.85 Timelapse Helper

It is used to get a timelapse quickly and easily. It sets the number of frames to be skipped during playback. The default value of *Number of Frames x block* is 10; playback (and rendering) displays frames 1 - 10 - 20 - ... and not the intermediate ones.

10.10.86 Title

The *Titler* allows you to add text from within CINELERRA-GG. It has many configuration capabilities. Often it is convenient to apply the plugin on a separate track (shared or not), in order to make it independent from the video track and have more freedom in editing and configuration. For example, you can apply the *Rotation* plugin to rotate the title without rotating the underlying video. Another thing to keep in mind when using the Titler plugin is that it works on the *temporary* and therefore it is not affected by Camera movements (curves), but only by Projector movements. If the original source has values outside the range $(0-1.0f)$, the plugin will always produce clipping.

The titler has standard options for font, size, and style plus many options as described next (figure 10.69)

Justify justifies the text relative to the entire frame. Once justified, the X and $Y\ offset$ is applied. This allows text to be justified while at the same time letting you push it within the title safe region.

Motion type scrolls the text in any of the four directions. When using this, the text may disappear. Make sure the speed is set to a reasonably high value (for example 150) and move the insertion point along the timeline until the text is far enough along the animation to reappear. The text scrolls on and scrolls off. Setting `loop` causes the text to scroll completely off and repeat. Without loop the text scrolls off and never reappears. The speed of the animation is determined by speed in pixels per second. Set it higher to speed up the animation.

Smooth with *Chroma Key*, a shadow may show; remove by enabling this to add/remove the shadow.

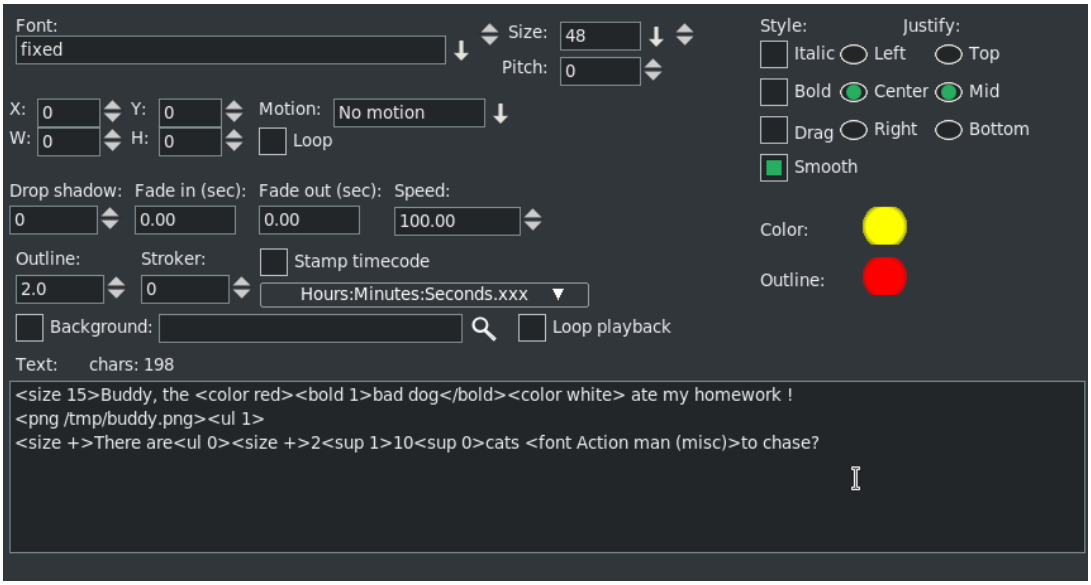


Figure 10.69: *GUI of the Title plugin*

Drop shadow draws a black copy of the text to the bottom right of the original text. This is useful when drawing text over changing video to keep the border always visible.

Fade in/Fade out are a second type of animation. If the fade seconds are 0, no fading is done.

Color and Outline allows you to pick the color for using to draw the text or its outline.

Stamp timecode replaces text with the current position on the timeline in seconds, frames or samples.

To create special effects for your title you can place it on a dedicated track and insert other realtime video effects just under the title effect and/or use camera and projector. With keyframing you can animate your title and make it change position, size, color, transparency, texture, or shape over time.

For improving playback performances of titles with effects, you can reduce the size of the dedicated track. Right click on the track and select *Resize track...* Enter the smallest resolution that still keeps the title visible.

To include graphical elements like *logos*, you may want to import your title as a PNG image (alpha channel transparency is possible), move it with camera and projector or add effects.

Some recently added options

Drag initial default checkbox is *off* so that the Title plugin will work as it always has. With Drag active, we can create a box that works as a region to the title; we can

resize and translate it as we wish, and we can add a background image (or video) to it. See later how to use the background image.

Anchors When you turn on the Drag feature, nine different anchors/handles will appear on compositor window. The *middle anchor* allows you to drag your title wherever you want in the compositor window (X, Y coordinates). The other 8 handles, drawn as arrows in each corner and in the middle of each side, let you change the size of the drag area box so that your title is within that area if it fits and as it is directed. If you need to clear the Drag enabled, you can easily do this with *Allow keyframe spanning* whose use is described in Allow Keyframe Spanning. Another way to reset the box to default, i.e. the entire frame, is to set X and Y to 0 and W and H to the frame resolution (e.g. 1920x1080 for FullHD).

W/H the values in these 2 boxes specify the size of the drag area box measured in pixels as shown in the compositor window. You can set these manually and if you can't see the location of your box or find your handles, set them to zero because 0 sets it to the same as the width/height of the media. The Drag effect ignores all boundaries, including the *Title Safe Region* of the Compositor so that if you drag your titles off the screen, it will look like they disappeared completely. Reset X and Y to reasonable values to have it reappear. The Title *text*, *background*, and *pngs* are applied on a single layer so that they will drag together as an entity. All of the Title capabilities work in conjunction with dragging so if you want to justify the title, you can still use the *Left/Center/Right/Top/Mid/Bottom* within the drag area. Be sure to turn off Drag when rendering or the box will show in the video; keep in mind that drag bars do not appear until there is some text in the text box and you can not actually drag until the Title window controls are available.

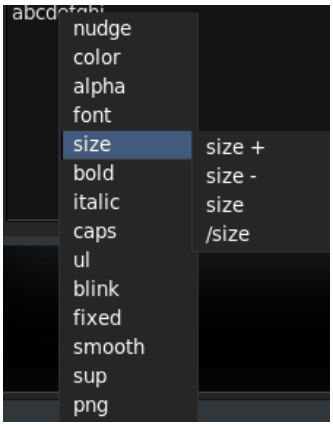


Figure 10.70: *Pulldown Attributes*

Attributes in the Text box where you type your Title information, you can now change several attributes to give you plenty of flexibility (figure 10.70). Each of these special attributes begin with an open angle bracket < and ends with a closing angle bracket >. Until the closing angle bracket is keyed in, the actual characters you

type, will be seen in the compositor window. In addition, if you do not use the exact syntax or you keyin a filename that is not available, all of the characters will continue to show up. This helps to see what needs to be fixed or is missing. The attributes usage is described in the table below.

Table 10.2: *Titler attributes*

Attribute name	Attribute value	Notes
color	color name such as RED from *<cin_path>/guicast/colors.h*	Or use the hex value like *#a000a0*; color-hex.com shows examples
font	exact name from *Font* pulldown	When you set font, bold/size and italic will be as currently set up
alpha	floating-point number between 0 and 1	0 is transparent; 1 is opaque
size	+ increases the size by $\frac{5}{4}$, - decreases $\frac{4}{5}$ or use a number > 0 but less than 2048	Examples: *<size +++>*, or *<size ->* or *<size 14.5>* (floating-point ok)
png	filename of a *.png* file	Example, add a logo. Full pathname needed if not in current directory
bold	1 for *on* (default) or use 0 for *off*	Font used must have bold available
italic	1 for *on* (default) or use 0 for *off*	Font must have italic already set up
blink	a number in seconds to flash on and off; a negative number for fade in and out	Can be a floating-point number; *<blink>* with no value is like 1
ul	underline 1 for *on* (default), 0 for *off*	
caps	1 for *on*, 0 for *off*, -1 for lower case	If no value set, treated same as 1
sup	1 for superscript; 0 ends; -1 for subscript; positive bumps up by $\frac{1}{2}$ of font size; negative sinks by 1/2 of current size	Size of text will be $\frac{1}{3}$ of font size; example: *12<sup 1>10<sup 0>*; <sup> with no values, acts like 1
fixed	number of pixels between characters; fixed with no value uses $\frac{3}{4}$ current size	A fixed font is very useful for tables; /fixed reverts to previous fixed size

nudge	x,y to move the text by that amount; displacement is in number of pixels. Can be negative or positive	Example: *<nudge 8,8>abc</nudge>*
smooth	add anti-aliasing to smooth jagged edges	Turn off smooth for chroma key

These attributes stay in effect until you change them or reset them. Additional cpu time is needed for the *blink* attribute because it requires redrawing every frame and for the background option described below. Note that some Title window controls can not be set, such as Justify and Outline color. The lines below are examples for testing purposes. The accompanying screenshot displays the corresponding CINELERRA-GG output.

Examples of Title line keyins:

Figure 10.71.

```
<size 15>Buddy, the <color red><bold 1>bad dog</bold><color white> ate my
    ↪ homework !<png /tmp/buddy.png><ul 1><size +>There are<ul 0><size
    ↪ +>2<sup 1>10<sup 0>cats <font Action man (misc)>to chase?
```

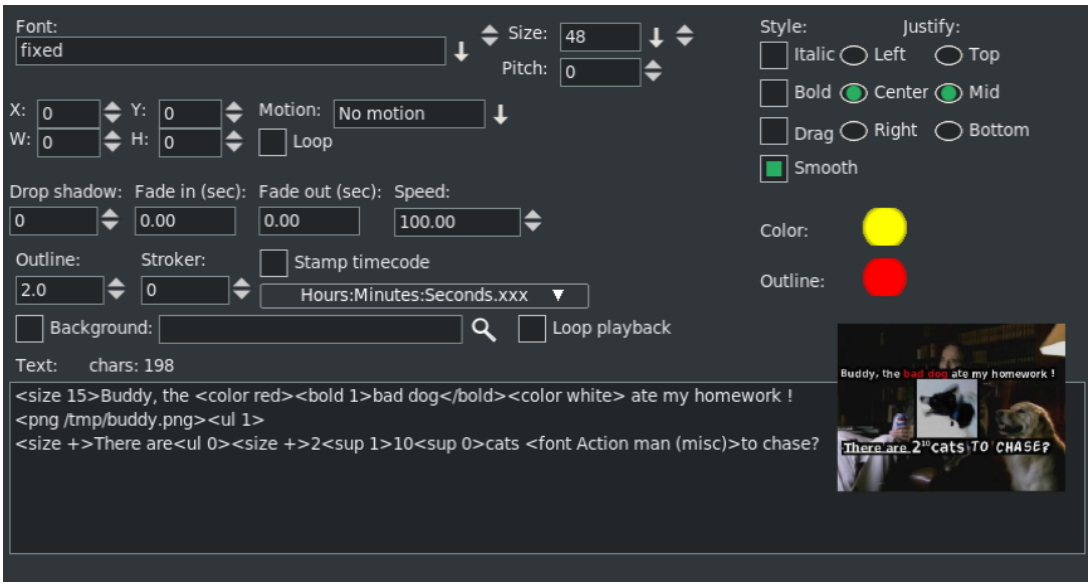


Figure 10.71: *Final result of the exemple*

Special Characters (< > / \#) Besides the previously described <, >, and / characters, there are two special characters: backslash "\", and the pound sign "#". The backslash

character is used for two things. With the advent of the attribute name and value, your line may become quite long so you can use "\" followed by the carriage return to continue on the next line as if it is just a single line. It also can be used to designate that the following character does not represent the beginning of an attribute. For example, if you want to use the opening angle character "<" as a title character, precede it with the backslash character. The pound sign, "#", designates the whole line as a comment or if in the middle of the line, then the rest of the line is a comment (this includes the carriage return).

Background in this box you can keyin the name of a file of the type that CINELERRA-GG accepts and use that file as a background for your Title characters (media video or image). This will be seen in the compositor window on top of the video that is loaded in the main track canvas. Besides typing in the filename, you must also check the checkbox. This makes it easy to turn it *On* and *Off* to see what it looks like. Next to the background box is a *Loop* checkbox. If the background file takes less time than the main track canvas video to run, you can turn on the loop checkbox so that it runs over and over again to match the time size of your video. By default the background image will occupy the whole frame, but if we activate *drag* mode and create a box of the desired size, the image will occupy only the box leaving the rest of the frame visible.

Stroker to add *pen strokes* to the text letters, adjust the stroke width numerically. This looks particularly nice on certain fonts and with a negative adjustment of the *Drop shadow*.

Text: chars: to output and update the number of characters already used. The only limit to the number of characters based on a count of single 8 bit characters is the available resources on the user computer available for CINELERRA-GG use. Keep in mind that unicode or other special characters may consist of 2 to 4 8 bit bytes. Also, newlines are a character and any of the attributes you use count in the total. There is now a horizontal scroll bar as well as the vertical one in the textbox and they only appear when there are more lines or characters than can fit in the original sized textbox.

Unicode Insertion if you want to enter a special character like the mathematical *summation* symbol, you can use the unicode equivalent to do so. Press Ctrl-Shift-U followed by 2022 and a carriage return is an example of the bullet. Refer to section 17.5 for details.

Popup Helper put your cursor where you want to add an attribute, then right mouse will bring up a list of the available attributes for you to choose, along with a submenu to choose from. The program will insert that attribute for you and all you have to add is a value when required! (see figure 10.70).

Color Picker Usage

The Text Color window has several enhanced features as listed here and seen in figure 10.72.

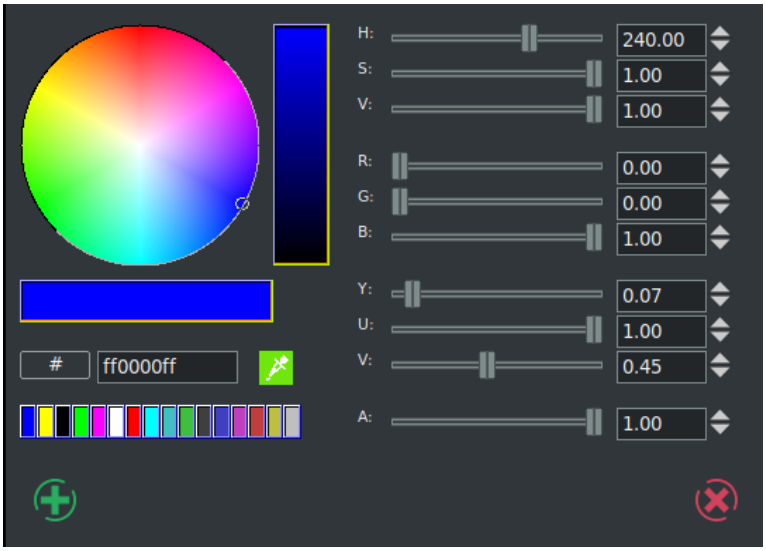


Figure 10.72: *Screencast showing the Color Picker menu.*

1. The hex value of the color you choose shows in the textbox and you can also keyin a value there.

2. A small square next to the hex text box, is a green *eyedropper* color picker. Use the left mouse button to click on the square to enable picking and you will see it turn red to designate that it is enabled. Your cursor will switch to a two-colored reticle. You can now move the mouse around to choose a color anywhere on the screen and then click there to have it picked. If you hold down the right or left mouse button while moving, you can see the color changing in the vertical bar in the color palette area as you move to give you a bigger view of the actual color.

3. A history of 16 of your last chosen colors is available to easily use again. Any time you choose a new color in any methodology, it will become the latest choice in the history either immediately or after checking OK/and leaving. History shows latest color starting left to right.

4. Besides HSV, RGB, there is also a YUV color model to choose from.

5. Several of these items may have associated *Tool Tips*.

Font Choice

The X Window system originally did not have a suitable font renderer for video. It also is restricted to the current bit depth. It does not have a convenient way to know which fonts work with the suitable font renderer in the desired bit depth. The easiest way we have found to support fonts in the titler is to have a directory for them at `/usr/lib/cinelerra/fonts`. Or if not system installed, at `<cinelerra_install_path>/bin/plugins/fonts`.

In order to choose a font faster, you can keyin the first few characters of the font name, being sure to use capital characters if used since it is case-sensitive. The steps to follow are:

1. next to the font box, click on the *down arrow* pulldown;
2. keyin the first character(s) of the desired font and you will see the first match become highlighted;
3. you can see the characters you keyed in the upper right corner of the fonts.

It is important to note that because there are checkboxes for both Bold and Italic, the pulldown will not include these specific fonts as an addition to a Regular font. They will be used via the checkboxes but not viewed in the pulldown.

One last item of information about fonts that may lead to some confusion. The checkbox for Bold and Italic will occasionally be ghosted out if no bold or italic version of the selected font is available. This is no guarantee, but currently as good as it can get due to inconsistency in the creation of fonts. It is mostly just a hint. If boxes are checkmarked, but ghosted, you can not uncheck until you change to a font that does not ghost out the boxes. If you use the popup helper with the boxes checked, and attempt to keyin a font that does not have the bold/italic attribute as checked, the font will be considered illegal.

Also of note, if the video is displayed on a consumer TV, the outer border is going to be cropped by 5% on each side. To avoid text which is too close to the edge looking bad, you may want to enable the *title-safe* tool in the compositor window. The text should not cross the inner rectangle.

Font Addition / Font Subtraction

If using a compiled build (that is, NOT an AppImage) to add true type fonts, copy the .TTF files to the fonts directory at `/usr/lib/cinelerra/fonts` or if not system installed, at `<cinelerra_install_path>/bin/plugins/fonts`. In that directory run

```
# /usr/lib/cinelerra/fonts
ttmkfdir && mv fonts.scale fonts.dir
```

and restart CINELERRA-GG. The new fonts should appear. However, keep in mind that the next time you install a new version of CINELERRA-GG, your changes will be written over so you will have to make sure to save them elsewhere and then re-establish. The usage of ttmkfdir changes frequently so this technique might not work. The titler supports mainly *TTF*, true type fonts. It supports others such as OTF, but TTF are the most reliable.

AppImage does not provide this specific method unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds. However, if you use Ubuntu Studio 2304 with AppImage, you can easily add font files directly to /.fonts and you will have all of these available in the Title fonts list. And, there may be other operating systems with specific implementations that also allow for this easier methodology to add additional fonts.

Some of the system fonts are automatically included in the set of fonts being used by CINELERRA-GG. The easiest way to add additional system or other application available fonts for the Title plugin's set, is to use ones available in specific directories on your computer as long as they have a `fonts.scale` file already set up. You can run `mkfontscale` to create this file within that directory. In order to include a specific directory you set an environment variable before starting CINELERRA-GG which stays in effect until it is unset or until the next reboot. Below is the method and an example.

```
export BC_FONT_PATH=<colon-separated-search-path-for-fonts>
export BC_FONT_PATH=/usr/share/fonts
```

The current set of fonts in CINELERRA-GG's directory will be automatically included and will be the default set if this environment variable is not set. Keep in mind that if you add a lot of fonts, it will considerably slow down the startup every time you bring up the Title plugin.

If you want to only have a limited number of fonts set up, you can manipulate the CINELERRA-GG directory directly at `<cinelerra_install_path>/bin/plugins/fonts`. Here you will find the default set of fonts that come with the install. Copy any other fonts you would like to include here with read permission, delete any fonts you do not want to have, then execute `mkfontscale` which creates the file `fonts.scale` that CINELERRA-GG will read. However, the next time you install a new version of CINELERRA-GG, your changes will be written over so you will have to make sure to save them elsewhere and then re-establish.

If you have problems with a specific font or set of fonts, there is a debug option available to determine which font is an issue. When starting CINELERRA-GG, you should set up the variable:

```
export BC_FONT_DEBUG=1 (default is 0 for no debug)
unset BC_FONT_DEBUG  (to remove debug messages)
```

Then start CINELERRA-GG from a terminal window to see the fonts being loaded and previewed in the Titler. This should point out any issues. Another debug methodology is to remove all fonts from being used and subsequently add in the ones that you want. This not only speeds the load up, it ensures that the fonts you actually are using are the ones

you expect because the order that CINELERRA-GG finds various fonts makes a difference in which one of potential duplicates shows up in the pulldown. So you might not see the one you expect if it was written over by a system font.

```
export BC_FONT_PATH=: #(the : "colon" removes all automatic system and Cinelerra fonts)
export BC_FONT_PATH=:/usr/share/fonts #(remove all fonts and then add /usr/shar/fonts
    ↪ )
```

Kerning

When using the Titler, kerning is applied in order to allow parts of a letter to go outside the standard sized letter box. Kerning is the process of adjusting the space between individual letters. It is not the same as proportional spacing. In kerning, a bounding box is allowed to overlay another bounding box. The philosophy here is to aim at the ability to have boxes overlap in order to make the letters look more visually appealing. Bounding box and escapement are tracked separately. In addition to adjusting individual letter spacing, the program will also expand the render box in order to have any parts of the letter extend outside the standard letter box. Kerning is applied to any and all fonts (figure 10.73).

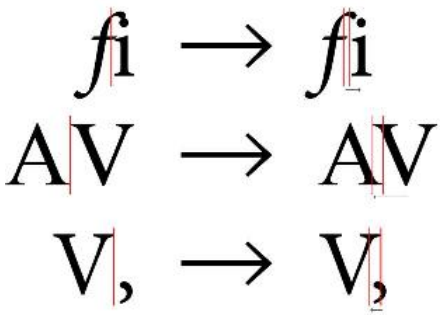


Figure 10.73: *Kerning in action*

10.10.87 Tracer

Tracer creates an outline around an object after a few points are designated, so it traces the object. Its operates in a similar manner to that of Gimp's *magic wand*. You can then Fill the alpha channel with a mask, like with a gradient or another object on another track or Invert that fill. Although it works best for still images or objects that stay in the same place on a video like a logo, you can also add keyframes. To limit the mis-tracking that occurs when working on a moving object, we can increase the number of points so that the tracking is more accurate and relies on a solid foundation. Rather than *points* as in a mask, tracer is based on *edges* to form an outline (figure 10.74). Frequently it will be desirable to use either RGB or YUV as the color model without the -A for Alpha.

New to create a new point.

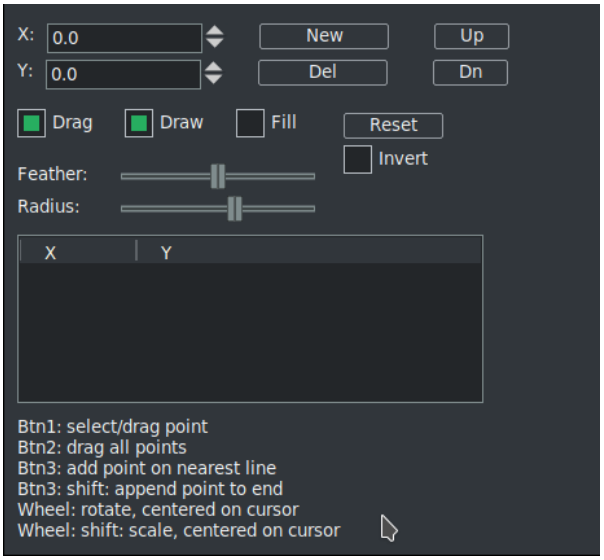


Figure 10.74: *Gui for Tracer plugin*

Up/Dn to move highlighted point up or down.

Del to delete the highlighted point.

X, Y are the point's x/y coordinate.

Invert to change the mask to outside.

Feather to sort of blur the edges (inward and outward). Feather influences the softness of the edge. The higher the number Feather is set to, the softer the edges become.

Radius for varying the size a little. Radius extends the area further in or out to cover more or less area.

Drag to enable or disable moving. If you need to clear the Drag enabled, you can easily do this with *Allow keyframe spanning* whose use is described in Allow Keyframe Spanning.

Draw to draw the outline for visual.

Fill to black out the object (no alpha); figure 10.75.

Mouse wheel rotate the outline, centered on cursor. Note that this rotates the outline around the nearest point. The slower you scroll, the more precise the the result. If you scroll faster, an accelerator kicks in that allows you to make complete rotations more quickly.

Mouse wheel + shift scale the outline, centered on cursor.

Be sure to uncheck *Draw* and *Drag* before rendering so that the lines do not show in the video output.

Figure 10.75: *Rug in the top picture is traced in order to black it out or replace later*

10.10.88 Translate

This effect allows displacing, cropping, and/or scaling the source video horizontally and/or vertically. The *In* and *Out* parameters operate similar to the *camera* and *projector* functions in the Compositor:

In X/Y specifies how many pixels from the left/top of the source you want to start (camera).

Out X/Y defines where on the screen you want the output to start (projector).

In W/H defines how many pixels of the source you want to include in each direction.

Out W/H defines how many pixels on the screen you want that source to take up.

Identical values for both *In* and *Out* that are less than the source dimension will simply crop the source. Different values will stretch (or compress if $Out > In$) the source in that direction (and crop if *In* is less than the source dimension).

This effect supports keyframes so these parameters can change smoothly over time. You can use this effect for many things such as having a cropped insert clip move across the screen, or have it change size or stretch while doing so. Be forewarned though, that for interlaced footage horizontal displacements are likely to destroy the field order, resulting in all sort of flickering and jumping movements.

10.10.89 Unsharp

This effect applies a traditional *darkroom* technique, the so called *unsharp mask* to every video frame. With different parameter values, this can be used to soften or to sharpen the image. You can use sliders or, for maximum precision, you can use the input box. Its parameters are:

Amount moving the slider to the right makes dark areas get darker and light areas get lighter.

Radius this slider controls how much blurring is used in the *edge-finding* stage. The practical effect of this is to specify how large a region is darkened or lightened.

Threshold this slider controls how big a difference between a pixel in the blurred copy and the original copy is needed before any darkening or lightening will be applied.

Clear to reset its slider to default without affecting others.

10.10.90 Videoscope

Videoscope summarizes intensity and color on a calibrated display. The Videoscope can be used in conjunction with other CINELERRA-GG plugins such as *Color 3 Way*, *YUV*, *Brightness*, *Color Balance* or *Histogram* to accurately correct video for contrast, clarity, conformance (to normalize various videos shot under different light settings), or for cinematic purposes. The human eye is not specialized to match precise level of light and color, but Videoscope is. Videoscope contains three displays: the waveform scope and the vectorscope, plus the histograms (figure 10.76). Instead of applying the plugin to the tracks/edits we want to examine, we can use the Videoscope buttons in the Composer and Viewer windows. In this way the monitors act on the frame indicated by the insertion point, without taking into account the stack of tracks or on which edits to apply the plugin.

If the original source has values outside the range (0—1.0*f*), the plugin will always produce clipping. The display will stop at -10% or +110%, but there is no clipping. For example, by varying the brightness all out-of-range values become visible on Waveform, even over 110%. However you must disable the "smooth" option which always causes clipping.

The Videoscope menu window has many options.

Scopes: we can choose between two histograms:

- *Histogram*
- *Histogram RGB*

three waveforms:

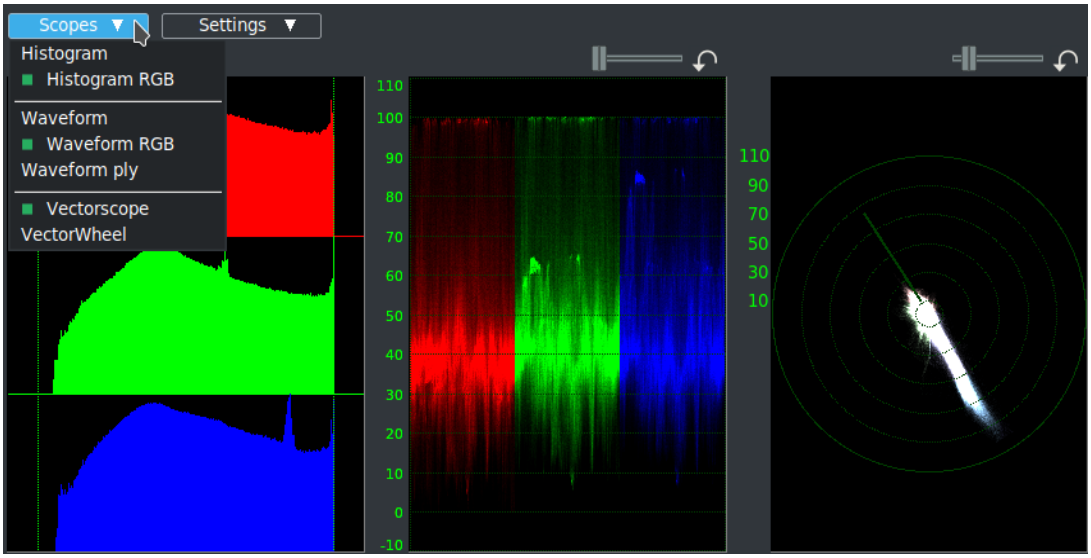


Figure 10.76: *GUI of the Videoscope. You see Histogram, RGB Parade and Vectorscope*

- *Waveform*
- *Waveform RGB* (aka RGB Parade)
- *Waveform Ply* (which superimposes the three Waveform RGB graphs in a single view.)

and two vectorscopes:

- *Vectorscope*
- *VectorWheel*

Settings: It is divided into two sections. The upper section contains two items:

Smooth: serves to make the graph more homogeneous, improving its visualization. However, it has a side effect: in case of frames that have data in the range of 100-110% (i.e., outside the normal range of 0-100%, also called super-white), it causes a clip by cutting off this data and reporting only of normal data. If we don't want this clip disable Smooth.

Refresh on Stop ON: [checked – only for Transport buttons] scopes are updated when you stop playback at a given location. Instead, they are locked at the start position while you playback. This saves system resources and makes playback smoother. By dragging the cursor the scopes are updated in realtime.

Refresh on Stop OFF: [unchecked – for Transport buttons and dragging cursor] the display of the scopes is synchronized with the playback. Every variation of the graphs is in realtime. There may be some decrease in fps during playback.

Refresh on Release: This works for the Viewer and Compositor windows. Scopes are not updated during playback. The update occurs only when you stop playback, that is at the final position (either by dragging the cursor or using the Transport buttons). When in the timeline, if you drag on the TimeBar or reposition in the TimeBar in either Drag and Drop or Cut and Paste mode, the release of the button also will update the Scopes. This saves system resources and makes playback smoother. Because there is no update when playing in the main window, you can still easily get a videoscope update simply by moving the mouse to the Compositor and a single click there will update the scopes without changing the frame (as long as Click to Play is not enabled).

The lower section, called *VectorWheel Grids*, it is only active when VectorWheel is set up in Scopes pull-down and contains four entries (figure 10.77):

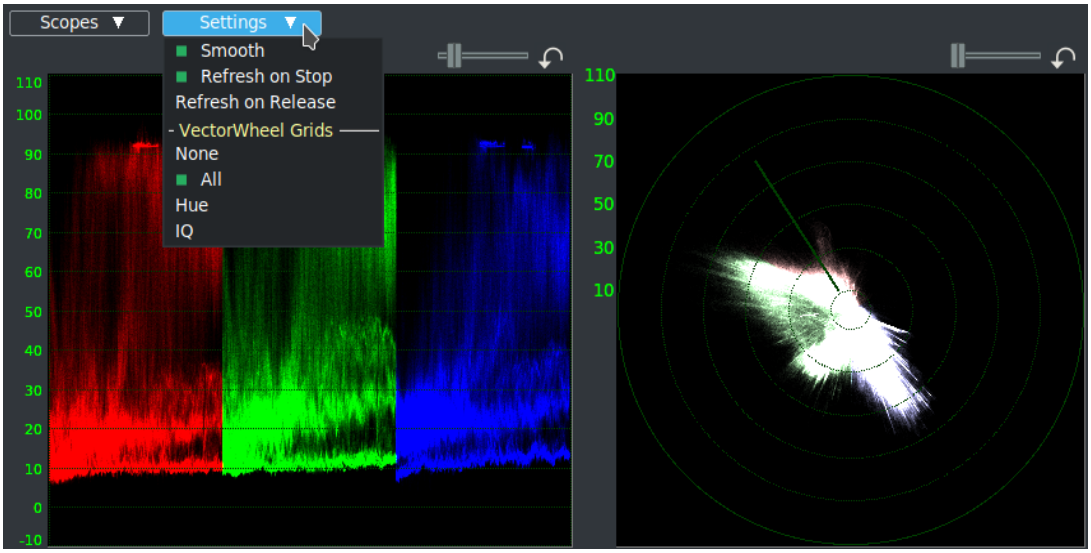


Figure 10.77: *Pull-down Settings*

None: It only shows the colour wheel, without any superimposed graticula.

All: Shows the most complete grid that allows precise and useful measurements. It also shows the skintones area.

Hue: Show a more minimal grid. It also shows the I-Line.

IQ: Show shows only I-Line and Q-Line.

You can create custom grids. These are raster .png images of adequate resolution and must be put in:

```
[your cinelerra path]/bin/plugins/scopes/custom_grid.png
```

More information in the later section on Vectorscope.

Sliders: allow for varying the *solidity* of the dots shown in Waveforms and Vectorscope. You can get to the subpixel level for precision. They are accompanied by Reset buttons that restore them to default values.

Waveform/RGB Parade/Waveform Ply

The *Waveform Scope* displays image intensity (luminance) versus image *X* position. The *Waveform RGB* displays image RGB intensity versus image *X* position (one graph per channel). The *Waveform Ply* shows the three channels in a single graph. The Waveform Scope appears on the left side or in the middle of the Videoscope window. The display is calibrated vertically from 0% intensity (black) at the bottom up to 100% intensity (white) at the top. Each column of pixels in the image corresponds to one column of pixels in the Waveform Scope (figure 10.78). Note that the height of the values of a waveform/waveform RGB corresponds exactly to the values on the *x axis* in the *histogram*. A vertical/horizontal correspondence is therefore obtained.

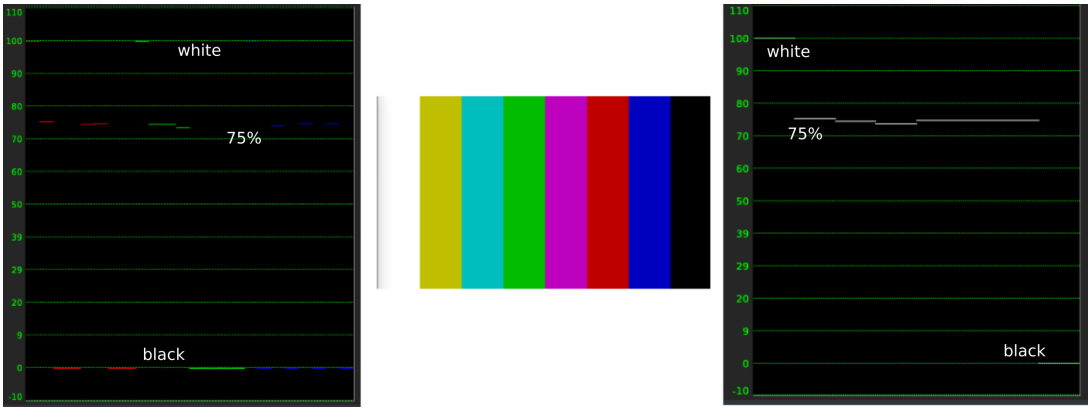


Figure 10.78: *Colortest 75% with Waveform RGB (left) and Waveform (right)*

On the left is shown waveform RGB: instead of the color shadows as in figure 10.76, we have lines representing the color bar test at 75%. They are pure colors, so all pixels have the same value. In fact, they are all at the level of 75% except for the 100% white band and the 0% black band. In the waveform on the right, we have the same behavior with regard to luminance: the white band is 100%; the black band is 0% and all the others 75%.

If we left-click on the graph with the mouse, we will see a crosshair that we can place exactly where we want to measure. We can read the precise values of X and Luminance (Value) in the pop-up box that appears at the bottom right (figure 10.79).

The Waveform scope helps correct image light levels for contrast range or for conforming light levels on various scenes originally shot on different light settings. The same can be done with Waveform RGB or the convenient overlapping representation (Waveform Ply).

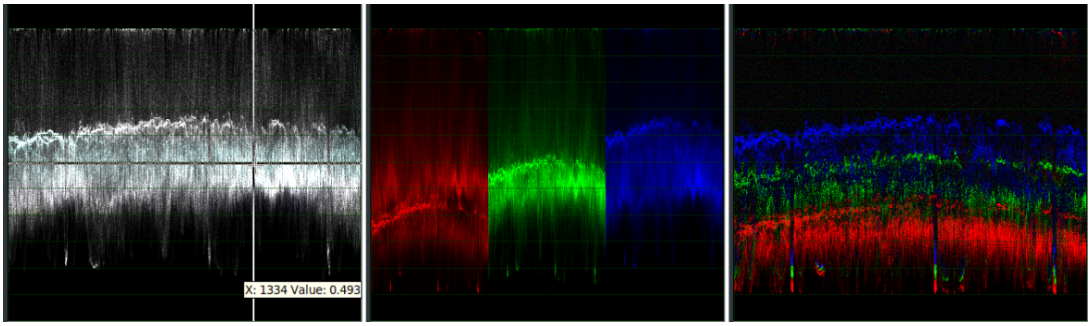


Figure 10.79: *Examples of waveform (with crosshair and the coordinates' box), waveform RGB and waveform Ply*

Adjusting luminance

1. Add the *Brightness/Contrast*, *Histogram*, *Color 3 Way* or another video adjustment effect on your track/edit.
2. Add the Videoscope effect on the track. Make sure that it is placed below in the stack of effects, so it can see the adjustment effect's results. If it is not, right-click and move it down. Or do not use the plugin but activate the Videoscope button in the Compositor window.
3. Show both the effect and Videoscope.
4. Adjust the contrast while observing the waveform to match the desired light level.
5. Precise adjustments can be made by measuring the values on the waveform with the crosshair (by clicking LMB, and reading numeric values on the pop-up box) and reporting these numbers in the effects window (*Histogram Bézier*, for example).

For example, if you are looking for maximum contrast range, adjust the *Brightness/Contrast* levels to align the darkest point on the scope with the 0% level and the brightest portion with 100%. Anything above 100% is over saturated. Limits may be highlighted with checkbox controls.

HDTV or sRGB (ITU-R BT.709)

The maximum pixel range for HDTV or sRGB is $[0, 255]$. This range corresponds with levels 0% and 100%.

MPEG or Analog video (ITU-R BT.601)

For analog video or MPEG (including DVD), the maximum range for RGB is $[16, 235]$ $(8\,bit)$. For YUV, the maximum range for intensity (Y) is $[16, 235]$ $(8\,bit)$. This range corresponds to gray levels from 6% to 92%. Reference *RGB-601* plugin.

NTSC Television broadcast

If you are producing a video for NTSC television broadcast, keep the intensity between 7.5% and 100%. The minimum black value which can be broadcast is IRE 7.5% (indicated by the *7.5 level*), and values below this level are no darker.

Vectorscope

The Vectorscope displays *hue* (angle on the color wheel) and *saturation* (radius). Each pixel in the source image is drawn as a point on the color wheel. The distance from the center is the color saturation. Gray values are close to the center, and high saturation values are near the perimeter (100%). In the center there is pure white (0%). By clicking with the mouse on the color wheel a radius and circle will appear whose values of hue and saturation are shown in the pop-up box at the bottom right of the window, similar to the values of *X* and luminance of the Waveform and RGB Parade (figure 10.80).

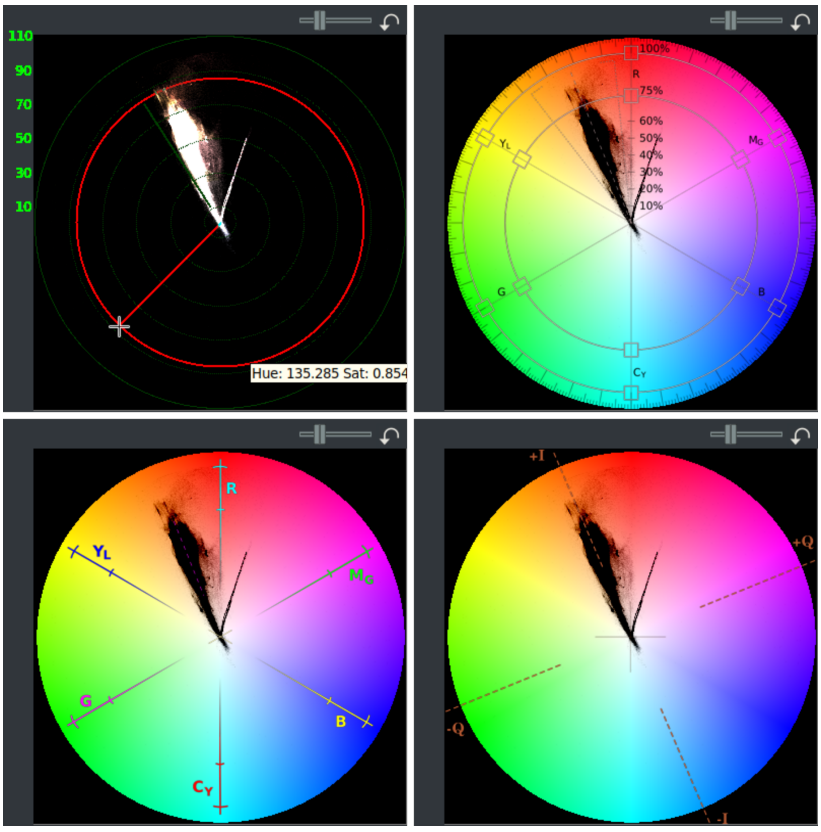


Figure 10.80: *Vectorscope (with H/S pop-up box) and VectorWheel (with 3 default grids)*

Note that when you choose *VectorWheel* from *Scopes* pulldown, you can choose between different grids in the *Settings* pulldown. In addition, any number of user-supplied grid patterns can be added in the form of a square image of type png. The user can design and maintain individual grid overlays for various purposes. The user would keep their overlays

in a safe spot on their disk and make a copy of them in the `{cinelerra_pathname}/bin/plugins/scopes` every time a new version of CINELERRA-GG is installed.

Generally the Vectorscope has the following uses:

1. Compare saturation levels (radius) between Edits.
2. Compare the hues (angle) between Edits.
3. Compare skin tones, sky and vegetation.
4. Display the color balance or color cast.
5. Check the legal limits of the video signal.

For example, skin tones are found along an axis (*+ I-line*) between yellow and red, and between 0 and 50% saturation values. The blue of the sky is more or less along the opposite axis to that of the skin (*- I-line*), with a much wider saturation range.

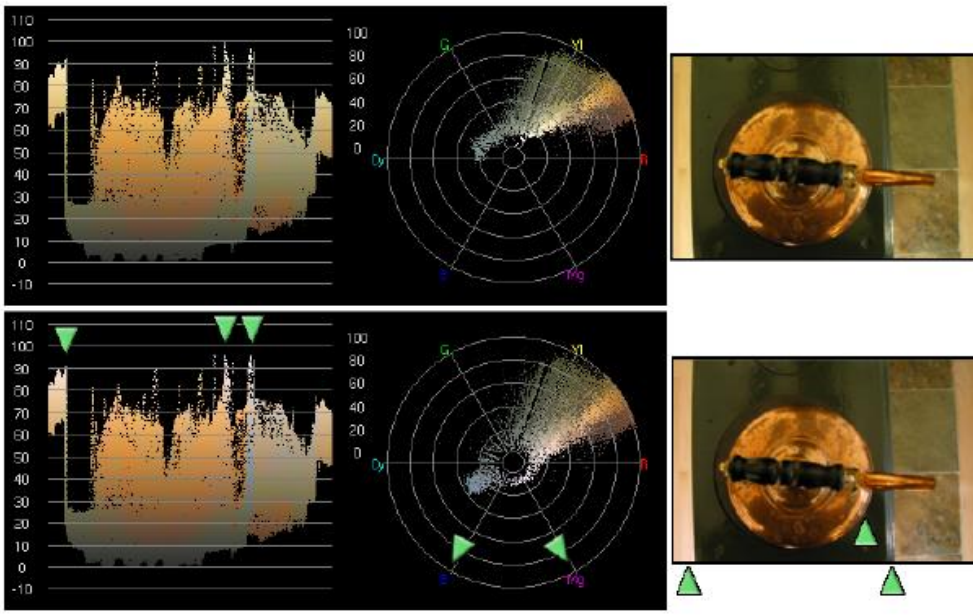


Figure 10.81: *Balancing a yellow color cast*

In figure 10.81, the top image is white balanced. Vectorscope shows many pixels in the yellow region and few in the white region. To remove the yellow tint, the Color Balance effect is used to first shift the vectorscope plot towards magenta (Mg), and then towards blue (B) until the region previously near the center surrounds the center. In the bottom image, yellow highlights have become white highlights (green triangles). Note that the corresponding features in waveform also appear whiter (green triangles). The Vectorscope can also be used to verify that the video output will display properly on various monitors.

Any points along the inner radius will be displayed as pure white and any points above the 100% radius, will probably not be correctly displayed on the screen.

Histogram

You can also display the 4 histograms (master or RGB) on the left of the window. (see figure 10.76).

10.10.91 Wave

The *wave* effect adds waves on the image. *Amplitude*, *Phase*, and *Wavelength* parameters can be adjusted. You can use sliders or, for maximum precision, you can use the input box. With the *Clear* buttons we can bring the slider to default values without affecting the other parameters.

10.10.92 Whirl

Creates a whirl (spiral) of the video around the center. You can use sliders or, for maximum precision, you can use the input box. With the *Clear* buttons we can bring the slider to default values without affecting the other parameters.

10.10.93 YUV

Modify the Y, U, V settings. You can use sliders or, for maximum precision, you can use the input box. With the *Clear* buttons we can bring the slider to default values without affecting the other parameters.

10.10.94 YUV411

YUV411 plugin[16] modifies the 411 yuv to look like 420 color space instead. If the edit to which the effect is applied is not YUV type, a red warning message will appear.

10.10.95 YUVShift

This effect is used for YUV input video from older cameras using 3 sensors. It is possible to have misalignment of the 3 sets of numbers: *Y*, which represents the luminance or brightness component, and for *U* and *V* representing the chrominance (color) components. If they were misaligned in the video, you can use *YUVShift* to realign. To move a specific component up/down, modify the *dy* value using the slider bar in the YUVShift window. To move a component left/right, modify the corresponding *dx* value. For maximum precision you can use the input box. With the *Clear* buttons we can bring the slider to default values without affecting the other parameters. If you are using an RGB color space, you will want to use the *RGBShift* effect instead.

[16] credit Eric Olson

Figure 10.82: *Before and after YUVShift adjusting*

Figure 10.82 (top) shows the blue *U* component aligned too far to the right/down. And the red *V* component is misaligned too far to the right/down. Note the *U_dx* current slider bar set to 0 as shown by the input box value in the YUVShift plugin window. All components are currently at zero. A corrected video image is shown in the bottom. Now the red and blue colors are correctly aligned. Note how *U_dx* is now at +26 and *V_dx* is now +41 to realign the image.

10.10.96 Zoom Blur

Blurs the video from the center outwards, like the sun's rays, and uses a zoom effect.

Figure 10.83: *For clarity of presentation only 3 fields are shown*

X,Y center of the origin field.

Radius Zoom on the fields.

Steps number of blur steps to be used in the calculation. Increasing the number takes more CPU.

Clear With the Clear buttons we can bring the slider to default values without affecting the other parameters.

Reset To reset all parameters to default at the same time.

You can use sliders or, for maximum precision, you can use the input box. Figure 10.83 shows the parameters: Radius=21 and Steps=3.

10.11 CUDA plugins

Only for Nvidia GPU and Cuda SDK

10.11.1 Mandelbrot

Produce fractal figures (figure 10.84). Use case:

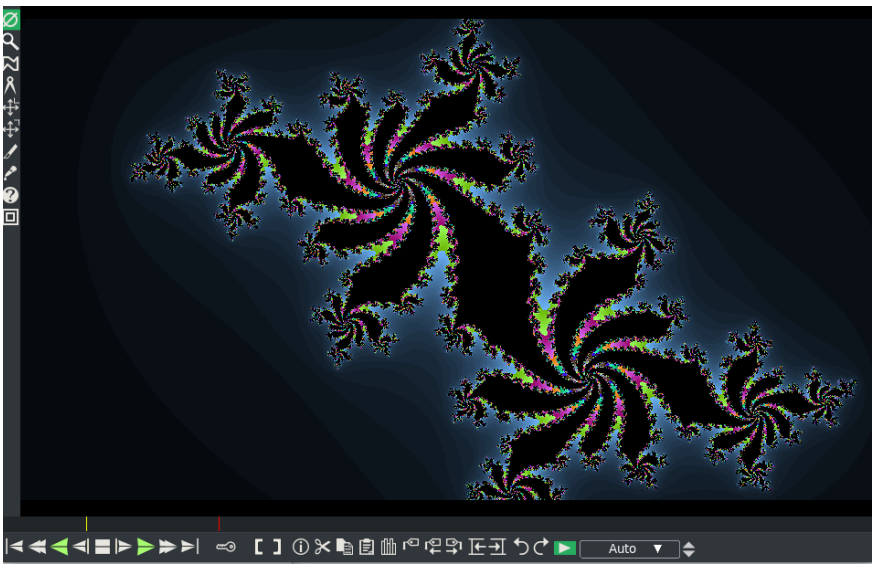


Figure 10.84: *Mandelbrot Cuda plugin*

1. add the Mandelbrot Cuda video effect to the timeline
2. you must enable Drag in the plugin popup menu
3. drag the mouse with the RMB down and you will see the fractal change

4. enable "Generate keyframes while tweaking" and move to another spot on the timeline and RMB drag again

5. so now when you play the video from the beginning because you have keyframes, the fractal will be animated between keyframes.

10.11.2 N_Body

Produce animated particles (figure 10.85).

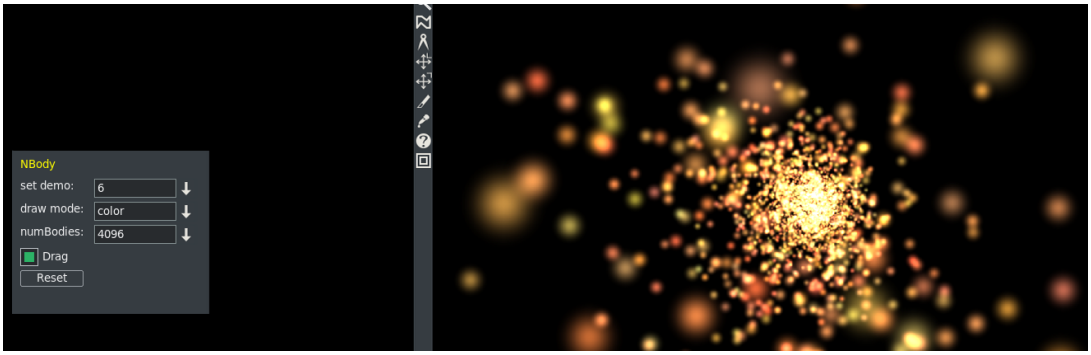


Figure 10.85: *N_Body Cuda plugin*

10.12 OpenCV plugins

There are currently 6 OpenCV plugins hooked into Cinelerra static and dynamic tars with the most widely known being FindObj. The *Find Object* plugin searches a Scene for an Object and then the Object can be overlaid with a Replacement object. The other 5 are FlowObj, GaborObj, MoveObj, PuzzleObj, and StylizeObj. These plugins require the third-party *OpenCV* (Open Computer Vision) library. Of note for OpenCV is that the *Surf* algorithm is non-free and has been removed from the user supplied versions of OpenCV being built.

There are some potential pitfalls with using OpenCV to include:

1. openCV code is constantly changing, making it difficult to keep the plugins up to date

2. there is some confusion about licensing/patents for specific algorithms, notably Surf

3. some parts don't work all that well and it can be slow

4. internally the colormodel is converted to greyscale, which means the color information is not used

5. cmake is minimally required to be at version 3.5.1 for OpenCV version 4.2; therefore for some older verions of distros with a lower cmake version, OpenCV is only at

3.4.1; and in really old versions such as Ubuntu14 32/64 bit, it is not in the build at all

Because of the build size of OpenCV, the source is not included in the thirdparty directory. Building OpenCV adds a lot of storage demand (4 *GB*) to the CINELERRA-GG build tree, and its compile time is significant. For these reasons, *findobject* and the other 5 plugins are not included in System build packages. We recommend using the static library build which has the OpenCV plugins included for consistency and to avoid problems.

10.12.1 Adding OpenCV Plugins to the System Package Builds

The OpenCV plugins are built only in the 64-bit tarball builds, both static and dynamic, but due to size these plugins are not included with pkgs, i.e. the System builds. However it is relatively easy to add the current plugins for your distro via a simple procedure of copying the plugins from the static tarball to the cin5 install plugin path (AppImage does not provide this capability). They are:

```
cin/plugins/opencv/findobj.plugin
cin/plugins/opencv/flowobj.plugin
cin/plugins/opencv/gaborobj.plugin
cin/plugins/opencv/moveobj.plugin
cin/plugins/opencv/puzzleobj.plugin
cin/plugins/opencv/stylizeobj.plugin
```

1. do your package install of the current build for your distro as usual
2. look in `https://cinelerra-gg.org/download/tars` to see your distro name's static tar
3. download the corresponding distro static tarball; for example for arch:

`https://cinelerra-gg.org/download/tars/cinelerra-5.1-arch-{date}-x86_64-static.txz`

4. create a temporary directory on your computer
5. `cd` that-directory
6. `tar -xf` location-of-the-tarball-you-downloaded
7. `cp plugins/opencv/*obj.plugin <see below for your location>/.` (note the period on the end)

8. Start CINELERRA-GG and look for the six plugins in Video Effects
9. To reverse this, simply delete the six plugin files (for example: `rm /usr/lib*/cin*/*obj.plugin`).

The location for most User installs is:

`<cinlib_path>/plugins/video`

Location for some System installs is:

`/usr/lib/cin/plugins/video` (most ubuntu distros)

`/usr/lib64/cin/plugins/video` (Leap distro)

10.12.2 How to Build OpenCV Plugins

For users who do their own build or for administrators who want to create a System build that includes OpenCV, here are some notes to facilitate doing so. Building findobject will configure and build OpenCV into the thirdparty library build area if needed (it does not rebuild if done previously). Network access is required to obtain the OpenCV source, at least once. You will need a minimum of 4 *GB* in the thirdparty build directory and more time to compile.

Build using stable tarballs

Instead of using the latest git version, there is a custom archive source tarball on the website already created to build and test OpenCV versions. The reason for this is that OpenCV is quite volatile, and may or may not work on any given day so this is an already tested version that is considered to be more stable. A build using the tarball requires that your system have *wget* already installed to initially download the prepared source tar. There are currently 2 tarballs of stable versions on the website used for user builds. The 20180401 version:
https://cinelerra-gg.org/download/opencv/opencv-20180401.tgz
works with older version of cmake, and 20200308 is the current 4.2 release:
https://cinelerra-gg.org/download/opencv/opencv-20200306.tgz

To build using the tarballs on the website, use this bld.sh procedure (you may want to run "make clean" first):

```
#!/bin/bash
( ./autogen.sh
  ./configure --with-single-user --with-booby \
    --with-opencv=sta,tar=http://cinelerra-gg.org/download/opencv/opencv-
        ↪ 20200306.tgz
```

```
  make && make install ) 2>&1 | tee log
mv Makefile Makefile.cfg
cp Makefile.devel Makefile
```

Build using current git version

To build findobject and the other plugins using OpenCV, access the src using git:

```
git clone --depth 1 git://git.cinelerra-gg.org/goodguy/cinelerra.git cinelerra5
```

To use the latest version, the method for creating a tarball is:

1. download both opencv and opencv_contrib source tarballs from the web: https://sourceforge.net/projects/opencvlibrary/files/4.2.0/ https://github.com/opencv/opencv_contrib/archive/4.2.0.tar.gz
2. create a temporary directory opencv_src and unpack the tarballs into the src directory
3. rename these sources to opencv and opencv_contrib
4. run a test build, using opencv_build for config:

```
cd opencv\_src
mkdir build
cd build && cmake \
  -DCMAKE\_BUILD\_TYPE=RELEASE \
  -DWITH\_IPP=OFF \
  -DWITH\_LAPACK=OFF \
  -DWITH\_GPHOTO2=OFF \
  -DBUILD\_SHARED\_LIBS=ON \
  -DINSTALL\_C\_EXAMPLES=ON \
  -DINSTALL\_PYTHON\_EXAMPLES=ON \
  -DBUILD\_EXAMPLES=ON .. \
  -DBUILD\_opencv\_python3=no \
  -DCMAKE\_INSTALL\_PREFIX=/usr/local \
  -DOPENCV\_EXTRA\_MODULES\_PATH="\$(opencv)\_contrib/modules/"
```

Next configure the build, but add the `--with-opencv` configure parameter. An example of usage of shown next.

```
cd <path>/cinelerra-5.1
./autogen.sh
./configure <add std params, eg: --with-single-user> --with-findobect=sta
--with-opencv may be set to [<bld>][,<src>]
bld=typ sta,dyn,sys: sta=static, dyn=shared, sys=system libraries
src=typ git,tar,git=url,tar=url: git (default git\_url), tar (default: tar\_url)

# Use one of the following:
--with-opencv=sta
--with-opencv=dyn
--with-opencv=sta,tar=https://cinelerra-gg.org/download/opencv/opencv-20180401.
    ↪ tgz

# For example: ./configure <params> --with-opencv=sta,git
```

Once thirdparty/opencv is built, it will be reused. To remove `thirdparty/opencv*`, use `mrclean`.

Ordinarily OpenCV requires the network to download pieces. But to avoid downloading the same source multiple times as you do test builds, you can instead use the following procedure.

1. Create a new opencv_src directory with the needed source (src) files, plus the missing pieces in ./opencv/.cache/; This source base does not require the network, and so is usable for batch builds.
2. Store the tarballs in a safe location, and reference them from an httpd server running locally on the build system. for example:
   /var/www/html/cinelerra.patch
   /var/www/html/opencv-20180401.tgz -> /mnt0/opencv-20180401.tgz
   /var/www/html/opencv-20200306.tgz -> /mnt0/opencv-20200306.tgz
3. systemctl start httpd

10.12.3 Description of Find Object Plugin

As in the standard OpenCV FindObj program, there are 5 *detector* methods and 2 *matcher* methods which can be selected. They detect features and match them as a rectangular projection. The matched region will be overlayed with a replacement image if replace object is enabled. This is done using a variety of feature detectors and region matches. The match works by creating sets of Feature points. These points are generated for both the source and reference object frames. Then the two sets are matched for *Homography* (a regional similarity).

Matchers

FLANN Fast Library for Approximate Nearest Neighbors

BF (flann not selected) Brute Force

Detectors Algorithm

SIFT Scale-Invariant Feature Transform

SURF Speeded-Up Robust Features Be aware that *Surf* is non-free and has been removed from versions of OpenCV being built. If you have used it in a past project, it could result in a SEGV crash.

ORB Oriented fast and Rotated Brief

AKAZE Accelerated Keypoint detector And descriptor Extractor

BRISK Binary Robust Invariant Scalable Keypoin

Don't Calculate Do nothing

The CINELERRA-GG plugin has several additional features for ease of use and more flexibility. A description of parameters is outlined below.

Mode:
- Square
- Rhombus
- Rectangle
- Parallelogram
- Quadrilateral (default)

Reset: to get back to the default values.

Algorithm: detector selections as described above.

Replace object: checked if replacing the object is the desired outcome.

Use FLANN: use the Matcher, or if unchecked just use brute force.

Draw match: will display a blue line around the matching object only for the purpose of verifying the placement. The starting point for the box is designated by a white outlined circle in corner #1.

Scene/object/replace layer: number designating the attachment track of each findobject plugin element.

After matching, the resulting projection corners are used to do reshaping. The general procedure is:

- Calculate centroid, translate center to 0, 0; check for bowties and rotation direction.
- Calculate replacement object corners using *mode*, *central angles*, *scale*, and *aspect* of the projection.
- Using the replacement, apply the inverse aspect, rotation, scaling, and translation as specified.

Selectable specifications

Aspect when set, the matching aspect ratio is applied, otherwise the original object aspect is applied.

Scale when set, the match surface area is used for scaling, otherwise the original surface area is used.

Rotate when set, the match orientation is used in the projection, otherwise the original orientation is used (orthonormal).

Translate when set, the match replacement is centered to the projection center, otherwise the original object center is used.

Once the replacement corners are calculated, the replacement object is overlayed to the scene frame.

Scene/Object/Replace Drag only 1 can be checked on at a time (and that includes counting the *Title* plugin *drag box*). The Drag boxes are faster alternatives to using the column dials.

Scene/Object/Replace columns of X/Y/W/H x and y positions along with width and height; $0 \div 100\%$ + Replace ΔX and ΔY deltas for the Replace column.

Draw keypoints when enabled shows the detection points created by the feature detection and used in the matcher Algorithm calculation.

Draw scene/object/replace border displays each of the borders which helps to determine correctness.

Object blend amount can dampen (smooth) the corner motion from frame to frame to reduce jitters.

10.12.4 An example of how to use Find Object

The following steps were used to set up the example in figure 10.86.

1. For best results, set *Play every frame* in `Settings` → `Preferences` → `Playback A`.
2. Load 3 tracks of png/jpg files – this is one of the more useful working cases:

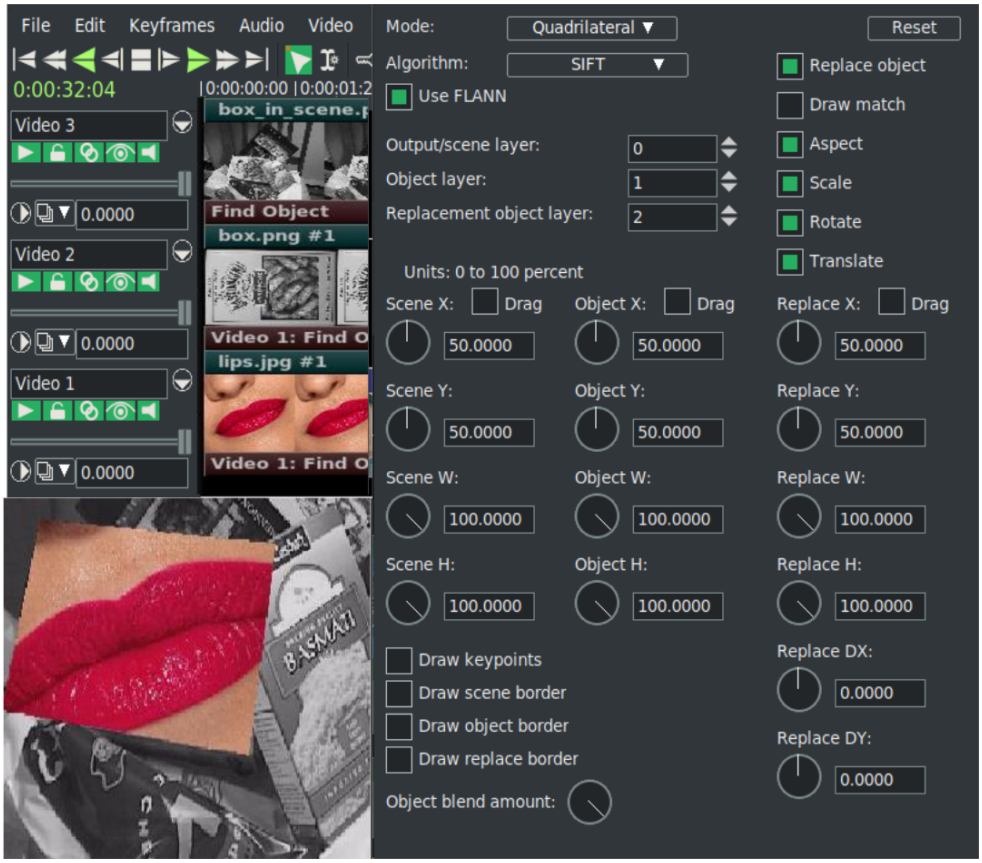


Figure 10.86: *Timeline; control window and compositor for FindObj*

- 1^{st} track should be the *scene*; that is the output
- 2^{nd} track is the *object* to find
- 3^{rd} track is the *replacement* object

3. Drag the *Find Object* plugin onto track #1.

4. On each of other 2 tracks, click the right mouse button; choose *attach effect*, highlight *Find Object* in the *shared effect* column, and click OK. All three tracks should now have a findobj plugin effect bar. One, (probably the first one) is the master plugin and the others are attached input tracks.

5. Check the plugin show icon on the master track to bring up the controls for the FindObj plugin. You will see that *Use FLANN* is checked for using nearest neighbors.

6. Set *Output scene*, *Object*, and *Replacement object* layers' track number accordingly (numbered from zero).

7. Check *Draw scene* border and you will see a white border in the compositor window surrounding the whole image of the scene. This assumes default settings for *Scene center X,Y* (at 50%), and area *W,H* (100%). Adjust these however you need to via

the dials or more simply by checking *Drag* and dragging any of the 9 drag points. As shown above this in the controls, units are in 0 — 100%.

8. Turn off *Play track* in the patchbay on the first track so you can see track #2 in the compositor and then use the *Object X,Y,W,H* dials. You will see a blue-dotted line which should then be adjusted to surround the desired object. Since it can be a little tedious to use the dials, you will want to turn off the previous Drag box of track #1 and check this instead to do your dragging placement from drag points.

9. Turn off *Play track* in the patchbay on this track #2 so that track #3 is visible and if necessary, adjust the dials/drag box in the same manner. The border will have a pink-dotted line around it.

10. Turn back on all of the desired *Play tracks* in the patchbays.

11. Set the detector choice (should have *Don't calculate* set initially), for example use *SIFT*.

12. Turn off all of the *Draw borders* and whichever *Drag box* that may still be checked.

13. Check *Replace object* to see the replacement overlay. At this time you may still want to change the checkboxes of *Aspect*, *Rotate*, *Scale*, and *Translate* and adjust *DX/DY* in the Replace column.

10.12.5 MoveObj, FlowObj, GaborObj, StylizeObj, PuzzleObj

MoveObj plugin moves an object and stabilizes it.

FlowObj plugin retards image motion as shown with optical flow. You can turn on/off draw vectors.

GaborObj creates an interesting fractalius-like image effect using the Gabor filter.

PuzzleObj makes a puzzle out of an image. You can make the puzzle pieces smaller or larger with the Pixels slider bar. The Iterations slider bar allows for varying morphing distance (figure 10.87).

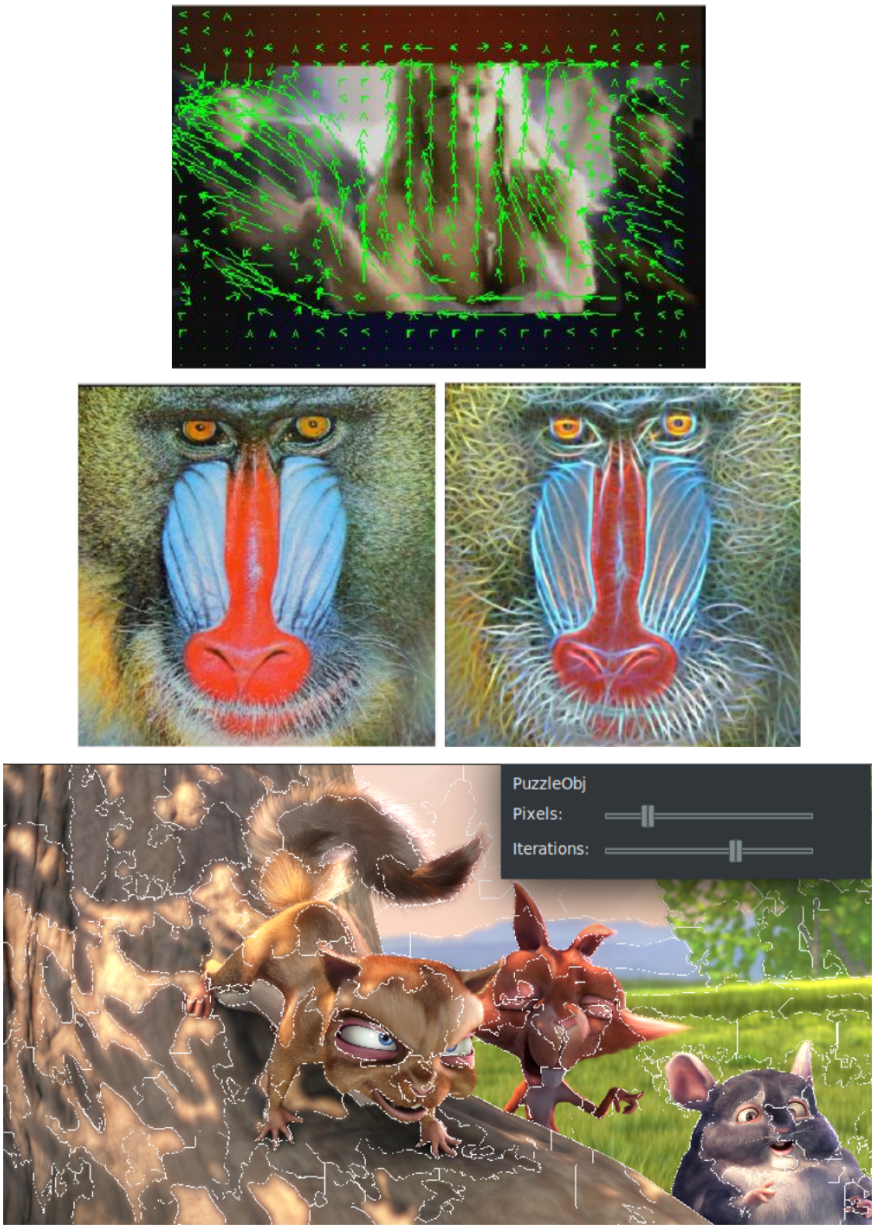


Figure 10.87: *FlowObj; GaborObj (before and after) and PuzzleObj*

The *StylizeObj* plugin can be used to create some interesting edge effects using various options (figure 10.88).

We can apply 6 different styles:

Edge Smooth / Edge Recursive

There are two edge preserving methods available – one to smooth the image edges, and the other to not smooth the edges/color boundaries, but instead replace the color values

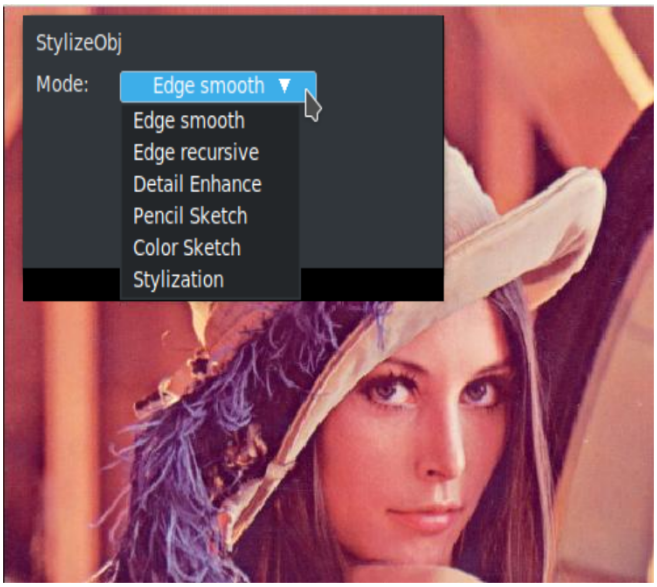


Figure 10.88: *Pulldown of Stylize mode on the original image*

at a pixel by the average of the pixels around the area which have color similar to that pixel.

Detail Enhance

This option enhances the details of an image to make it look sharper.

Pencil Sketch / Color Sketch

Pencil like line drawings – either applied to a grayscale version of the images, which can give you thin pencil to charcoal like results, or applied to the color input image, which will look like a colored pencil drawing. For best results you can vary the Smoothing (the size of the surrounding area), the Edges (how dissimilar colors in the surroundings will be averaged; a larger value results in large regions of constant color) and the shading (simple scaling of the output image intensity with the higher the value, the brighter the result).

Stylization

Produces output that looks like the images were painted using water colors, which often gives the impression of cartoon-like pictures. It abstracts regions of low contrast while either preserving or enhancing features of high contrast.

Figure 10.89 show the images after adding the 6 various styles.

Figure 10.89: *Clockwise: Edge Smooth and Recursive; Detail Enhance; Stylization; Color and Pencil Sketch*

10.13 FFmpeg Audio and Video Plugins

CINELERRA-GG INFINITY currently comes with more than 140 of the video plugins and 55 of the audio plugins developed by the FFmpeg project www.ffmpeg.org. These plugins do not have a GUI with buttons like the rest of plugins, therefore to change settings it is necessary to change the variables by hand by highlighting the *option*, typing a value in the *Range* box, and then hitting the *Apply* button. Many of these plugins provide tooltips at the bottom right corner of the window when the option is highlighted. A *slider* bar and a *dial* for numerical values can be used to easily vary the values which take effect immediately.

Figure 10.90 shows what an FFmpeg video plugin looks like; example is `F_chromakey`.

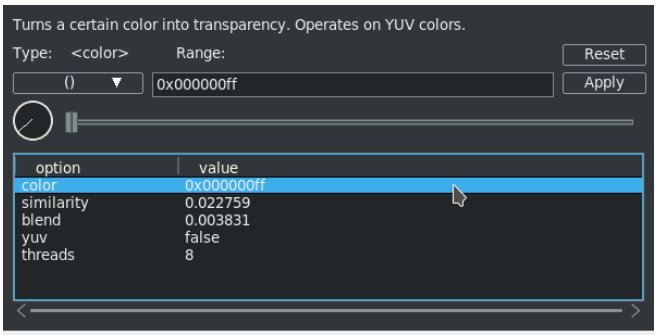


Figure 10.90: *GUI for FFmpeg plugins*

10.13.1 FFmpeg Icons and how to Replace

Currently FFmpeg audio and video plugins that do not have a personalized icon use one of the default icons as shown below.

Default Audio Icon: Default Video Icon:

If you want to replace the icon with a more descriptive picture, see a previous section on Updatable Icon Image Support and Details on where to put Plugin Icons.

10.13.2 How to Use FFmpeg Audio/Video Plugins

Simply drag and drop the plugin on the timeline. To enter the settings option, once you have added the plugin to the timeline, right click on the colored bar that appears below the timeline track of the plugin and highlight the *show* option. Alternatively, you can left click the *magnifying glass* icon that appears on the right side of the colored bar. For the audio plugins, if the plugin is not working at all, you will hear a trouble tone waveform to indicate that. If the value is unspecified, it uses default. If you specify the default, the value becomes unspecified which means that nothing gets sent into the program.

Some of the ffmpeg plugins are not usable with CINELERRA-GG INFINITY due to input/output requirements. Also, some do not come with legal initial supplied values for the parameters (ffmpeg works on filtergraph, while CINELERRA-GG works on stack). These plugins get tested at least once and if they crash, cause problems, or are deemed unusable, they are commented out in the `plugin.opts` file in the CINELERRA-GG ffmpeg subdirectory. Generally they are not retested so if the ffmpeg software changes them making them usable, they will still not be accessible until the `plugin.opts` file is changed. You can easily retest these, by temporarily uncommenting the copy of the `plugin.opts` file in the `bin` subdirectory and using `Settings → Preferences → Interface tab → Reload plugin index`

FFmpeg's plugin guide is at the link: ffmpeg-filters.

10.13.3 FFmpeg Audio Plugins

The following is a list of the integrated audio plug-ins[17] .

F_abench
: Benchmark part of a filtergraph.

F_acompressor
: Audio compressor.

F_acontrast
: Simple audio dynamic range compression/expansion filter.

[17] credit to WPfilmmaker

F_acrusher
Reduces audio bit resolution.

F_acue
Delay filtering to match a cue.

F_adecorrelate
Apply decorrelation to input audio stream.

F_adelay
Delays one or more audio channels.

F_adenorm
Remedy denormals by adding extremely low-level noise.

F_adynamicequalizer
Apply dynamic equalization to input audio stream.

F_adynamicsmooth
Apply dynamic smoothing to input audio stream.

F_aderivative
Compute derivative of input audio.

F_aecho
Adds echoing to the audio.

F_aemphasis
Audio emphasis.

F_aeval
Filters audio signal according to a specific expression.

F_aeexciter
Enhance high frequency part of audio.

F_afade
Fades in/out input audio.

F_aformat
Convert the input audio to one of he specified formats.

F_afreqshift
Apply frequency shifting to input audio.

F_afwtdn
Reduce broadband noise from input samples using Wavelets.

F_agate
Audio gate.

F_alatency
Measure filtering latency.

F_allpass
Applies a two-pole all-pass filter.

F_aloop
Loops audio samples.

F_anoisesrc
Generates a noise audio signal.

F_aperms
Set permissions for the output audio frame.

F_aphaser
Adds a phasing effect to the audio.

F_aphaseshift
Apply phase shifting to input audio.

F_apsyclip
Apply Psychoacoustic clipper to input audio stream.

F_arealtime
Slows down filtering to match realtime.

F_aresample
Resamples audio data.

F_asetrate
Change the sample rate without altering the data.

F_aspectralstats
Display frequency domain statistical information about the audio channels. Statistics are calculated and stored as metadata for each audio channel and for each audio frame.

F_astats
Shows time domain statistics about audio frames.

F_asubboost
Boost subwoofer frequencies.

F_asubcut
Cut subwoofer frequencies.

F_asupercut
Cut super frequencies.

F_asuperpass
Apply high order Butterworth band-pass filter.

F_asuperstop
Apply high order Butterworth band-stop filter.

F_atempo
Adjusts audio tempo so that when you speed up audio, it will adjust the pitch accordingly to make audio sounds more normal, but faster without being badly distorted. Its behavior is similar to what can be seen when you watch a youtube video, click the gear icon, and select a speed greater or less than 1.

F_atilt
Apply spectral tilt filter to audio stream.

F_atrim
Pick one continuous section from the input, drop the rest.

F_bandpass
Applies a two-pole Butterworth band-pass filter.

F_bandreject
Applies a two-pole Butterworth band-reject filter.

F_bass
Boosts or cuts lower frequencies.

F_biquad
Applies a biquad IIR filter with the given coefficents.

F_chorus
Adds a chorus effect to the audio.

F_compand
Compresses or expands audio dynamic range.

F_compensationdelay
audio compensation delay line.

F_crossfeed
Apply headphone crossfeed which blends the left and right channels of a stereo audio recording. It is mainly used to reduce extreme stereo separation of low frequencies in order to produce more speaker like sound.

F_crystalizer
Simple Expand Audio Dynamic Range filter.

F_dcshift
Applies a DC shift to the audio.

F_drmeter
Measure audio dynamic range where setting window length in seconds is used to split audio into segments of equal length.

F_dyaudnorm
Dynamic Audio Normalizer. When using this plugin, be sure to *attach effect* to all audio tracks by dragging the plugin to the 1^{st} audio track and then right mouse clicking all subsequent audio tracks which brings up an menu. Highlight the effect shown in the middle section and click OK.

F_earwax
Widens the stereo image. When using this plugin, be sure to *attach effect* to all audio tracks by dragging the plugin to the 1^{st} audio track and then right mouse clicking all subsequent audio tracks which brings up an menu. Highlight the effect shown in the middle section and click OK.

F_equalizer
Applies two-pole peaking equalization (EQ) filter.

F_extrastereo
Increases difference between stereo audio channels. When using this plugin, be sure to *attach effect* to all audio tracks by dragging the plugin to the 1^{st} audio track and then right mouse clicking all subsequent audio tracks which brings up a menu. Highlight the effect shown in the middle section and click OK.

F_flanger
Applies a flanging effect to the audio.

F_haas
Apply Haas Stereo Enhancer for a more natural sounding pan effect or more clarity in the center of the mix. With this filter applied to mono signals it gives some directionality and stretches its stereo image.

F_highpass
Applies a high-pass filer with $3\,dB$ point frequency.

F_hilbert
Generate a Hilbert transform FIR coefficients.

F_loudnorm
EBU R128 loudness normalization.

F_lowpass
Applies a low-pass filter with $3\,dB$ point frequency.

F_mcompand
Multiband Compress or expand audiodynamic range. The input audio is divided into bands which is like the crossover of a loudspeaker, resulting in flat frequency response when absent compander action.

F_pan
Remix channels with coefficients (panning).

F_silenceremove
Removes silence.

F_sine
Generate sine wave audio signal.

F_stereotools
Applies various stereo tools. When using this plugin, be sure to *attach effect* to all audio tracks by dragging the plugin to the 1^{st} audio track and then right mouse clicking all subsequent audio tracks which brings up an menu. Highlight the effect shown in the middle section and click OK.

F_stereowiden
Applies stereo widening effect. When using this plugin, be sure to *attach effect* to all audio tracks by dragging the plugin to the 1^{st} audio track and then right mouse clicking all subsequent audio tracks which brings up an menu. Highlight the effect shown in the middle section and click OK.

F_tiltshelf
Boost or cut the lower frequencies and cut or boost higher frequencies of the audio using a two-pole shelving filter with a response similar to that of a standard hi-fi's tone-controls. This is also known as shelving equalization (EQ).

F_treble
Boosts or cuts upper frequencies.

F_tremolo
Applies tremolo effect.

F_vibrato
Applies vibrato effect.

F_virtualbass
Apply audio Virtual Bass filter.

F_volume
Change input volume.

F_volumedetect
Detect audio volume.

10.13.4 FFmpeg Video Plugins

The following is a list of the integrated video plug-ins [18] .

F_addroi
Mark a region of interest in a video frame.

F_amplify
Amplify changes between successive video frames.

F_atadenoise
Apply an Adaptive Temporal Averaging Denoiser.

[18] credit to WPfilmmaker

F_avgblur
Apply average blur filter.

F_bbox
Compute bounding box for each frame.

F_bench
Benchmarks part of a filtergraph.

F_bilateral
Apply bilateral filter, spatial smoothing while preserving edges.

F_bitplaneoise
Measure bit plane noise.

F_blackdetect
Detect video intervals that are (almost) black.

F_blackframe
Detect frames that are (almost) black.

F_blockdetect
Determines blockiness of frames without altering the input frames.

F_blurdetect
Determines blurriness of frames without altering the input frames.

F_boxblur
Blurs the input video. Through the settings you are able to change the power and the radius of the boxblur applied to luma, chroma and alpha.

F_bwdif
Deinterlaces the input image.

F_cas
Apply Contrast Adaptive Sharpen filter to video.

F_chromakey
Turns a certain color into transparency. Operates on YUV colors.

F_chromanr
Reduce chrominance noise.

F_ciescope
Video CIE scope.

F_color
Provide an uniformly colored input.

F_colorbalance
Adjusts the color balance.

F_colorchannelmixer
Adjusts colors by mixing color channels.

F_colorchart
The colorchart source provides a colors checker chart.

F_colorcontrast
Adjust color contrast between RGB components.

F_colorcorrect
Adjust color white balance selectivity for blacks and whites.

F_colorkey
Turns a certain color into transparency. Operates on RGB colors.

F_colorlevels
Adjusts the color levels.

F_colormatrix
Converts color matrix.

F_colorize
Overlay a solid color on the video stream.

F_colorspace
Converts color space.

F_colorspectrum
Provides a color spectrum input.

F_colortemperature
Adjust color temperature of video.

F_cover_rect
Find and cover a user specified object.

F_crop
Crops the input video.

F_cropdetect
Auto-detect crop size.

F_curves
Adjust components curves.

F_datascope
Video data analysis.

F_dblur
Apply Directional blur filter.

F_dctdnoiz
Denoise frames using $2DDCT$.

F_deband
Debands video.

F_deblock
Deblocks video.

F_deflate
Applies deflate effect.

F_deflicker
Remove temporal frame luminance variations.

F_dejudder
Removes judder produced by pullup.

F_delogo
Removes logo from input video. When using this plugin a green box will appear on the screen, once the logo is inside the box the plugin will hide it. Through the settings you can specify the position of the logo to hide (on a $X-Yaxis$) and the size of the box (so you can adjust it to the size of the logo).

F_deshake
Stabilizes shaky video.

F_despill
Remove uwanted foregrond colors, caused by reflected color of green or blue screen.

F_dilation
Applies dilation effect.

F_doubleweave
Weave input video fields into double number of frames.

F_drawbox
Draws a colored box on the input video. Through the settings you are able to choose the position of the box on X/Y coordinates, the size of the box, the color and the thickness of the lines.

F_drawgraph
Draw a graph using input video metadata.

F_drawgrid
Draws a colored grid on the input video. Through the settings you can select the horizontal and the vertical offset, set the width and height of the grid cell, and the color and thickness of the lines. When using the Presets button on the plugin bar on the timeline, you can choose a preset of *rule_of_thirds*. The Rule of Thirds is a 3×3 grid on top of an image which is commonly used in filmmaking. The concept is that you align the key elements in the image using this grid at the intersection of the lines or along and within the vertical/horizontal lines. In general the bottom and right sides of the video will not be bordered by the grid. So for example, if you only add the effect to the video, there will be a frame around the video with the size of

the *thickness* but the frame is only visible on the top and left and not on the bottom or left side. You can get a line to show by using a larger thickness and specifying a negative value for x and y of -2, -3, or more.

F_edgedetect
Detects and draws edge.

F_elbg
Apply posterize effect, using the ELBG algorithm.

F_entropy
Measure video frames entropy.

F_epx
Scale the input using EPX algorithm.

F_eq
Adjusts brightness, contrast, gamma and saturation.

F_erosion
Applies erosion effect.

F_estdif
Apply Edge Slope Tracing deinterlace.

F_exposure
Adjust exposure of the video stream.

F_fade
Fade in/out input video.

F_fftdnoiz
Denoise frames using $3DFFT$.

F_fftfilt
Apply arbitrary expressions to pixels in frequency domain.

F_field
Extract a field from the input video.

F_fieldorder
Set the field order.

F_fillborders
Fill borders of the input video.

F_floodfill
Fill area of the same color with another color.

F_format
Convert the input video to one of the specified pixel formats.

F_framerate
Upsamples or downsamples progressive source between specified frame rates.

F_framestep
Select one frame every N frames.

F_fspp
Applies Fast Simple Post-processing filter.

F_gblur
Apply Gaussian Blur filter.

F_gradfun
Debands video quickly using gradients.

F_gradients
Draws a transparent gradient.

F_graphmonitor
Show various filtergraph stats.

F_grayworld
A color constancy filter that applies color correction based on the grayworld assumption.

F_greyedge
Estimates scene illumination by grey edge assumption.

F_haldclutsrc
Provide an identity Hald CLUT.

F_hflip
Horizontally flips the input video.

F_histeq
Applies global color histogram equalization.

F_histogram
Computes and draws a histogram.

F_hqdn3d
Applies a High Quality 3D Denoiser.

F_hqx
Scales the input by 2, 3 or 4 using the $hq * x$ magnification algorithm.

F_hsvhold
Turns a certain HSV range into gray values.

F_hsvkey
Turns a certain HSV range into transparency.

F_hue
Adjust the hue and saturation of the input video.

F_huesaturation
Apply hue-saturation-intensity adjustments to input video stream.

F_idet
Interlace detect Filter.

F_il
Deinterleaves or interleaves fields.

F_inflate
Applies inflate effect.

F_interlace
Convert progressive video into interlaced.

F_kerndeint
Applies kernel deinterlacing to the input.

F_kirsch
Apply kirsch operator.

F_latency
Measure filtering latency.

F_lenscorrection
Rectifies the image by correcting for lens distortion.

F_life
Generate a life pattern.

F_limiter
Limit pixels components to the specified range.

F_loop
Loops video frames.

F_lumakey
Turns a cerai luma into transparency.

F_lut
Compute and apply a lookup table to the RGB/YUV input video.

F_lut1d
Adjust colors using a 1D LUT.

F_lut3d
Apply a 3D LUT (lookup table) to an input video. LUTs are used to map one color space to another and are frequently supplied with high-end cameras as a `.cube` file to use as input.

F_lutrgb
Compute and apply a lookup table to the RGB input video.

F_lutyuv
Combine and apply a lookup table to the YUV input video.

F_mandelbrot
Render a Mandelbrot fractal.

F_mcdeint
Applies motion compensating deinterlacing.

F_median
Pick median pixel from rectangle defined by radius.

F_mestimate
Generate motion vectors.

F_monochrome
Convert video to gray using custom color filter.

F_mpdecimate
Remove near-duplicate frames.

F_mptestsrc
Generate various test pattern.

F_negate
Negates input video.

F_nlmeans
Non-local means denoiser. Example usage is for recovery of VHS tapes which look bad.

F_noise
Adds noise to the video. Through the settings you can select the variables of the noise (strength, flag and seed).

F_normalize
Normalize RGB video.

F_oscilloscope
2*D* video oscilloscope. Useful to measure spatial impulse, step responses, and chroma delays.

F_owdenoise
Denoises using wavelets.

F_pad
Add paddings to the input image, and place the original input at the provided x, y coordinates.

F_pal100bars
Generate PAL 100% color bars.

F_pal75bars
Generate PAL 75% color bars.

F_perms
Set permissions for the output video frame.

F_perspective
Corrects the perspective of video.

F_phase
Phases shift fields.

F_photosensitivity
Filter out photosensitive epilepsy seizure-inducing flashes.

F_pixscope
Pixel data analysis for checking color and levels. It will display sample values of color channels.

F_pixelize
Apply pixelization to video stream.

F_pp
Filters video using libpostproc.

F_pp7
Applies Postprocessing 7 filter.

F_prewitt
Apply prewitt operator.

F_pseudocolor
Make pseudocolored video frames.

F_readeia608
Read *EIA-608* Closed Caption codes from input video & write to frame metadata.

F_readvitc
Reads vertical interval timecode and writes it to frame metadata.

F_realtime
Slows down filtering to match realtime.

F_removegrain
Removes grain.

F_repeatfields
Hard repeat fields based on MPEG repeat field flag.

F_rgbtestsrc
Generate RGB test pattern.

F_roberts
Apply roberts cross operator which performs a simple/quick 2*D* spatial gradient measurement on the video (usually a grayscale image). It highlights regions of high spatial frequency which most ikely correspond to edges.

F_rotate
Rotates the input image by an arbitrary angle expressed in radians. If you want to rotate by degrees, you can use the *rotate* plugin instead.

F_sab
Applies shape adaptive blur.

F_scale
Scale the input video size and/or convert the image format.

F_scdet
Detect video scene change.

F_scharr
Apply scharr operator to input video stream.

F_scroll
Scroll input video horizontally and/or vertically by constant speed.

F_separatefields
Split input video frames into fields.

F_setparams
Force field, or color property for the output video frame.

F_setrange
Force color range for the output video frame.

F_shear
Shear transform the input image.

F_showpalette
Display frame palette.

F_shuffleframes
Shuffles video frames.

F_shufflepixels
Shuffles video pixels.

F_shuffleplanes
Shuffles video planes.

F_sierpinski
Generate a Sierpinski carpet/triangle fractal, and randomly pan around.

F_signalstats
Separates statistics from video analysis.

F_siti
Calculate Spatial Info (SI) and Temporal Info (TI) scores for a video, as defined in ITU-T P.910: Subjective video quality assessment methods for multimedia applications.

F_smartblur
Blurs the input video without impacting the outlines. Through the settings you can select the radius, the strength and the threshold of luma and chroma. This plugin can be used to correct over sharpened footage. For example, on a DJI 3840x2160 D-cinelike footage using these settings:

- luma radius: 3.247993
- luma strength: 0.360153
- luma threshold: -2

the over-sharpened edges will be smoothed. All other settings are at default values.

F_smptebars
Generate SMPTE color bars.

F_smptehdbars
Generate SMPTE HD color bars.

F_sobel
Applies sobel operator.

F_spp
Applies a simple post processing filter.

F_stereo3d
Converts video stereoscopic 3*D* view.

F_super2xsai
Scales the input by 2x using the $Super2xSal$ pixel art algorithm.

F_swaprect
Swaps 2 rectangular objects in video.

F_swapuv
Swaps U and V components.

F_tblend
Blend successive frames.

F_testsrc
Generate test pattern.

F_testsrc2
Generate another test pattern.

F_thistogram
Compute and draw a color distribution histogram for the input video across time.

F_tile
Tile several successive frames together.

F_tinterlace
Performs temporal field interlacing.

F_tlut2
Compute and apply a lookup table from 2 successive frames.

F_tmedian
Pick median pixels from successive frames.

F_tmidequalizer
Apply Temporal Midway Equalization.

F_tmix
Mix successive video frames.

F_transpose
Transposes input video.

F_unsharp
Sharpen or blur the input videlo.

F_untile
Untile a frame into a sequence of frames.

F_uspp
Applies Ultra Simple/Slow Post-processing filter.

F_v360
Convert 360 videos between various formats.

F_vaguedenoiser
Applies a Wavelet based Denoiser.

F_vectorscope
Video vectorscope.

F_vflip
Flips the input video vertically.

F_vfrdet
Variable frame rate detect filter.

F_vibrance
Boost or alter saturation.

F_vignette
Makes or reverses a vignette effect. Through the settings you can set the circle center position on a $X-Yaxis$,choose the angle, the aspect and set the dithering of the vignette.

F_w3dif
Applies Martin Weston three field deinterlace.

F_waveform
Video waveform monitor.

F_weave
Weaves input video fields into frames.

F_xbr
Scales the input using *xBR* algorithm.

F_yadif
Deinterlaces the input image.

F_yaepblur
Yet another edge preserving blur filter.

F_yuvtestsrc
Generate YUV test pattern.

F_zoompan
Applies Zoom & Pan effect.

10.14 Rendered Effects

Besides the *Realtime* effects, as has been described in the previous sections, another type of effect is performed on a section of the track and the result stored somewhere before it is played back. The result is usually pasted into the track to replace the original data. The rendered effects are not listed in the resources window but instead are accessed through the `Audio → Render effect and Video → Render effect` menu options. Each of these menu options brings up a dialog for the rendered effect. In the Select an effect dialog is a list of all the realtime and all the rendered effects. The difference here is that the realtime effects are rendered to disk and not applied under the track. Rendered effects apply to only one type of track, either audio or video. If no tracks of the type exist, an error pops up.

A region of the timeline to apply the effect to must be defined before selecting *Render effect*. If no in/out points and no highlighted region exist, the entire region after the insertion point is treated as the affected region. Otherwise, the region between the in/out points or the highlighted region is the affected region. The tracks to apply the rendered affect to,

need to be armed, other tracks are ignored. The rendered effect processes certain track attributes when it reads its input data but not others. Transitions in the affected track are applied, but nudge and effects are not. This allows the new data to be pasted into the existing position without changing the nudge value.

How to use a rendered effect

- Highlight an effect in the list (*Select an effect*) to designate it as the one being used.
- Define a file to render the effect to in the `Select a file to render to` box. The magnifying glass allows file selection from a list.
- Select a file format which can handle the track type. The *wrench* allows configuration specific to the file format.
- There is also an option for creating a new file at each label. For example, if you have a CD rip on the timeline which you want to divide into different files, the labels would become dividing points between the files if this option were selected. When the timeline is divided by labels, the effect is re-initialized at every label. Normalize operations take the peak in the current file and not in the entire timeline.
- Finally there is an *insertion strategy* just like in the render dialog. It should be noted that even though the effect applies only to audio or video, the insertion strategy applies to all tracks just like a clipboard operation.
- When you click OK in the effect dialog, it calls the GUI of the effect. If the effect is also a realtime effect, a second GUI appears to prompt for acceptance or rejection of the current settings.
- After accepting the settings, the effect is processed.

10.14.1 Rendered Audio Effects

Resample

This multiplies the number of each output sample by a *scale factor* to arrive at the number of the input sample. The output file's sample rate is set to the project sample rate but its length is changed to reflect the scaled number of samples. It also filters the resampled audio to remove aliasing.

If the scale factor is 2, every 2 input samples will be reduced to 1 output sample and the output file will have half as many samples as the input sequence. If it is 0.5, every 0.5 input samples will be stretched to 1 output sample and the output file will have twice as many samples as the input sequence.

Normalize

Time Stretch

CD Ripper

10.14.2 Rendered Video Effects

Reframe

This does exactly the same thing as *ReframeRT* in *Stretch* mode. It multiplies the output frame number by the scale factor to arrive at the input frame number and changes the length of the sequence. Unlike *ReframeRT*, this must run from the Video menu and render its output. Be aware Reframe does not write the scaled frame rate as the frame rate of the rendered file. It produces a file of scaled length and equal frame rate as the project. The new length is $\frac{1}{scalefactor}$ as big as the original sequence.

To create a slow-motion of fast moving video:

- Select the video clip you wish to reframe and put it on a video track.
- Select the area you wish to reframe.
- From the `Video` menu, select the *Render Effect* option.
- From the effect list, select *Reframe*.
- Enter the output format and insertion strategy for the new clip to be created.
- Press OK.
- At the popup menu, enter the *scale factor* 2 to run twice as fast, and 0.5 to run at half speed.

720 to 480

Chapter 11

Transition Plugins

When playing a section of media where one edit ends and another edit begins on the timeline, the usual result is that the first edit's output immediately is followed by the second edit. Transitions provide a better method whereby the first edit's output becomes the second edit's output. There are several different audio and video transitions listed in the Resources window as figure 11.1.

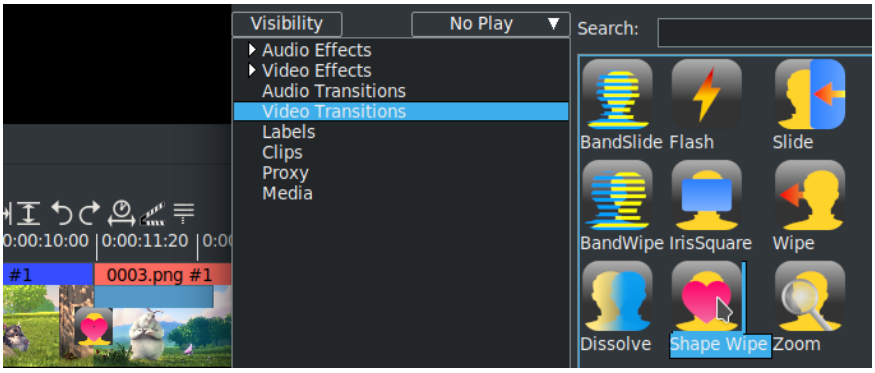


Figure 11.1: *Resources window displaying the Video Transitions.*

Note the bluish colored bar above the *Shape Wipe* transition. This bar near the transition symbol shows the position and the length of the transition. We can extend or reduce the length of the bar (and therefore the duration of the transition) by simply dragging the right edge of the bar to the desired position. In the *zoom bar*, at the bottom left where the words `Welcome to Cinelerra` appears, this is replaced by the real time length of the bluish bar. The unit of measure adopted is that of the timebar –the default is *hh:mm:ss:frame*; we can change it as we like (RMB on timebar).

Transitions only apply to the matching track type; that is audio transitions only apply to audio tracks and video transitions only apply to video tracks.

An example usage of a transition follows:

1. Load a single video file and delete a sizable section from within the video which will result in two edits of that file. Now you should see the edit boundary between the two edits on the timeline.

2. Move to the *Resources window* and click on the *Video transitions* folder. Choose a transition by highlighting it and then drag and drop it on the second video edit on the timeline. A colored box shows where the transition will appear and when you release the mouse the 2^{nd} edit applies the transition between the 1^{st} and 2^{nd} edit. The beginning of the edit will be covered by the transition, if the insertion point or the In point is over an edit.

Some Transitions have parameters that can be modified. To see these, move the pointer over the transition and right click which brings up a menu. A `Show` option will pop up a window if there are parameters that you can change to different values. An *On* option makes it possible to turn off the transition so that it will not be in effect in case you want to only enable it under certain conditions. The default value for this will be checked On. A *Length* option lets you adjust the length in seconds of the time that the transition will be in play. Values modified in the Show or Length will be saved for use the next time that transition is used until changed again. The *Detach* option deletes the transition from the timeline. When you drag and drop a different transition on top of an existing transition on the timeline, it replaces the previous one (see figure 11.2).

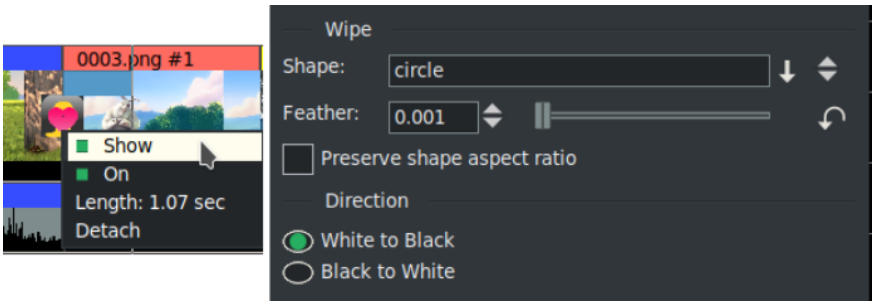


Figure 11.2: *GUI for Wipe transition*

There are some shortcuts to alleviate the dragging and dropping of transitions when you want to do a lot of them in various places on the timeline. After you have established the parameter values for a transition that you have dragged from the Resources window, you can use "U" and "u" keys to paste the same transition; the "U" key pastes the last video transition while the "u" key pastes the last audio transition on all recordable tracks. Alternatively, you can add the same transition to multiple edits when in *Arrow mode* (Drag and Drop editing), by selecting edits to add the transition to and use the *Video/Audio* pulldown to *Attach transition*. Select the desired transition and then click the checkmark OK. You can set a default transition in the Attach transition popup box – by highlighting your choice, then click on the button *Set Default Transition*, and you will see that transition become the new default.

The way that transitions work is that two edits overlap for some length of time and no edits are actually moved during the transitions. Instead extra frames from the source file will be used to lengthen the first edit enough to make it overlap the second edit for the length of the transition. The transition takes effect exactly at the beginning of the second edit and lasts for the specific length of time you set into the second edit. What this means

is that if you set a duration of 2 seconds for a flash transition, it will not start at the last 1 second of the first edit and continue 1 second into the second edit. Instead the transition will start exactly at the beginning of the second edit and last for 2 seconds into that second edit. This is why it is necessary that the first edit needs to have extra data after the end boundary to provide enough fill for the transition length into the second edit. In the case where the last frame on the timeline is the last frame of the source, CINELERRA-GG will lengthen the first edit using only that last frame. The result will not be what you want because the first edit will freeze into the transition.

When playing transitions, software rendering is used. This means that if you are using hardware, as with the video driver set to OpenGL, hardware acceleration will usually be turned *off* during the transition and *on* after the transition. Consequently, you may notice small anomalies while playing this section but you can avoid this by switching to using X11 video driver instead or just ignore it because when you create your final render that is always done in software only.

When *Info on* is enabled via the right mouse button over an empty space in the Resources window (or the shortcut of the letter "i" is used), a short description will be provided in the lower right hand corner of that window for the current transition that the mouse is on.

Once you have dragged and dropped a transition to the timeline, right mouse click on the transition and a pop-up menu will appear which provides an opportunity to make some changes. These are described next for all video and audio transitions (see figure 11.2).

Show If available, clicking on this will pop up a transition specific menu.

On Toggle on/off the transition effect so that you can see what it looks like when played. Ordinarily this will be checked to indicate it is On.

Transition length When you click on Length, a pop-up menu will come up. Set the length in seconds, frames, samples, *H:M:S:frm* or *H:M:S.xxx* for the transition to complete either by typing a value into the text box or using the tumbler arrows. In addition you can use the mouse wheel to change the length in real time. You will see the colored bar get longer or shorter as you change the length. There is a down arrow next to the Seconds box allowing for changing to the other dimensions of frames, samples, etc. The mathematics is automatically done for you to convert it to seconds if not already in that dimension.

Detach Remove the transition from the timeline.

11.1 Audio Transitions

Crossfade

Creates a smooth transition from one audio source edit to another. The crossfade has the first source *fade out* while the second *fades in*.

11.2 Video Transitions

BandSlide

Bands slide across video and you see the image slide.

BandWipe

Bands wipe across the video and you see the mask slides.

Dissolve

A soft dissolve where the In segment becomes more transparent while the Out segment begins to gradually show in its place until fully there.

Flash

The video flashes when transitioning between segments.

IrisSquare

Video switches segments via a small rectangular view that gradually grows to full size.

Shape Wipe

Wipe a specific shape across the video. Currently available 63 shapes are:

asimetric_clocks	*asimetric_clocks_2*	*audio*
burst	*Butterfly*	*Circle-h_01*
Circle-h_02	*Circle-v_01*	*Circle-v_02*
circle	*clock*	*Clouds_01*
Clouds_02	*crazy_clock*	*cream*
Cross-Iris_01	*Cross_01*	*Diagonal-Curve*
Diagonal-Linear	*Diamond-Iris_01*	*Diamond_01*
Diamond_02	*Diamond_03*	*Double-Door-h*
Double-Door-v	*film*	*film2*
Gravity	*heart*	*Linear-h*
Linear-v	*mosaic*	*multi_box*
Plasma_01	*rare_spiral*	*specks*
spiral	*spiral_clock*	*square*

squares	*star*	*tile2x2h*
tile2x2v	*Venetian-Blinds-h-x05*	*Venetian-Blinds-h-x10*
Venetian-Blinds-v-x05	*Venetian-Blinds-v-x10*	*water*
Cinelerra16-9-Heavy	*Cinelerra16-9-Light*	*CinelerraGG*
16-9_boart	*16-9_boxes*	*16-9_cat_eyes*
16-9_film_bands	*16-9_multi_circle*	*16-9_multi_spiral*
16-9_multi_square	*16-9_rare_spiral*	*16-9_rectangles*
16-9_star	*16-9_stars*	*16-9_wood*

The menu for Shape Wipe that popups when you click on *Show* has several possible choices. First, the *Shape* allows for choosing from the list of shapes as outlined previously either by typing in the textbox, using the down arrow, or clicking on the tumbler down/up arrows. Next, there is a *Feather* textbox with tumbler arrows or a slider bar to easily change the blending amount. A reset button is conveniently located to the right of the slider bar. There is a checkbox to *Preserve shape aspect ratio* and a *Direction* of *White to Black* or *Black to White* (figure 11.3).

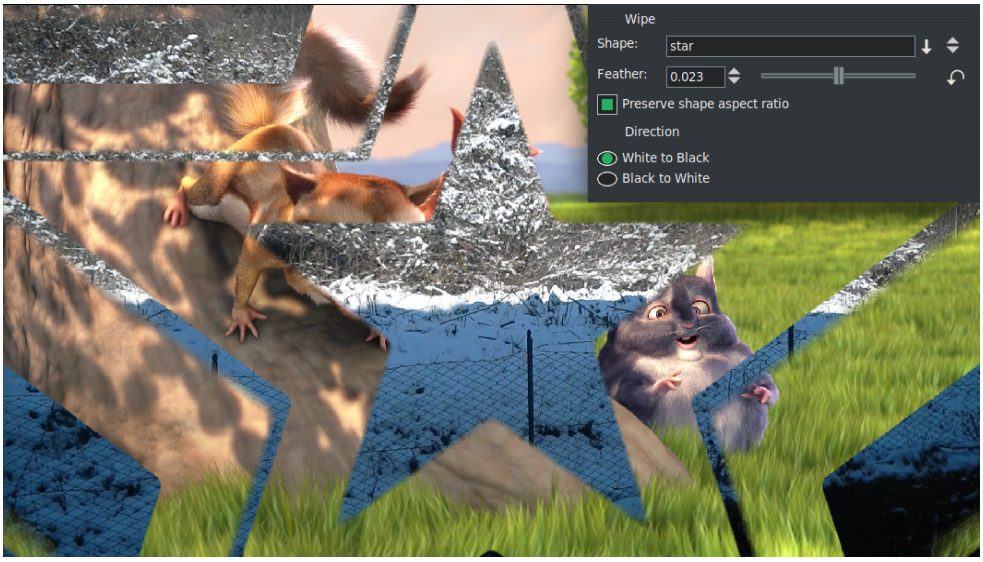


Figure 11.3: *Example of the Shape Wipe -> Star*

You can add your own images to the Shape Wipe transition and there are some free ones available to download at `http://assistcg.com` - click on the *Transitions* option under the *Video* pulldown on that site. AppImage does not provide this capability unless you use the workaround as described in the Appendix FAQ, Known Problems and Workarounds.

To include new images in the Shape Wipe Transition, simply copy the file *shape.jpg* or *shape.png* to the subdirectory `plugins/shapes` in your CINELERRA-GG directory path. If you prefer to have a better quality png used instead of the included 90% jpg version, you can download the equivalent png versions at `https://cinelerra-gg.org/download/testing/ShapeWipe_pngs.txz`. You will then need to untar this file, choose the ones

you want, and add them to your directory path. And if that is not enough, you can download another 32 different transitions at https://cinelerra-gg.org/download/testing/ShapeWipe_additional.txz. One particularly interesting ShapeWipe in this set is *Fleur-de-lis-of-Florence*. After an update of CINELERRA-GG, they will have to be restored each time.

Slide

Image slides into view; you can set Left/Right/In/Out.

Wipe

Wipe the image across screen starting left or right.

Zoom

Zoom out video at $\frac{X}{Y}$ magnification for some seconds.

Chapter 12

Overlays

The purpose of the Overlay Modes is to control the foreground and background stacking and use blending to reshape image object boundaries. It normally makes use of a binary type alpha blending system for all in or all out. To use the available operations in CINELERRA-GG, follow these steps:

1. In the main window, look at the Patchbay on the far left.
2. Click on the small arrow that points to the right, inside the panel to see it expand.
3. A down arrow shows up with a tooltip of *overlay mode*.
4. Use the down arrow to get to the popup menu and choose the desired effect from the 30 possibilities. *Normal* is the default.

Figure 12.1 shows the pulldown in the patchbay and the tool tip *Overlay mode* which provides access to 4 types of overlays. Each will expand further as shown in the screenshots below for *PorterDuffs* and *graphic art*.

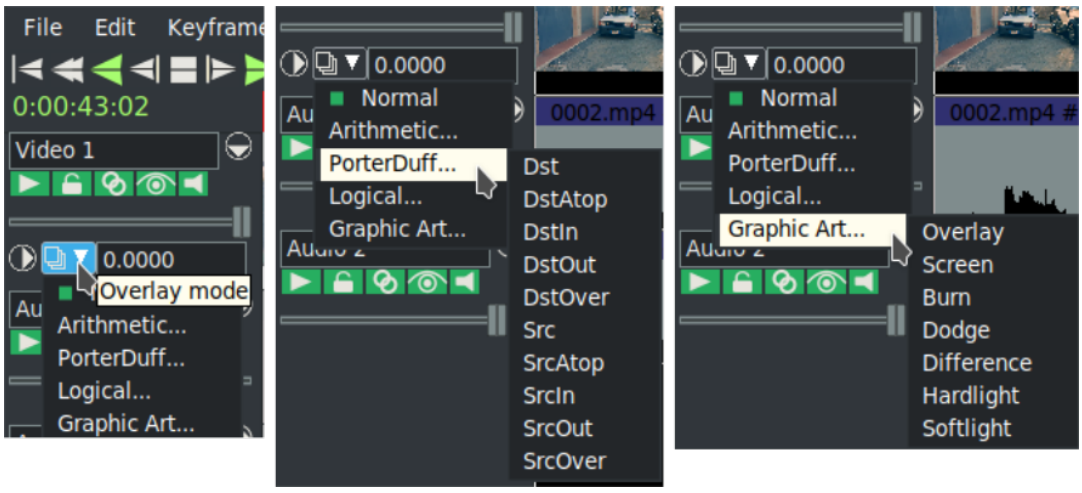


Figure 12.1: *Patchbay pulldown with Porter Duff and Graphic Art overlays expanded*

Porter-Duff is the industry standard for alpha blending operations. Only a short explanation follows here, but there is much more information to be found on the internet with

complete descriptions and examples. Every pixel has 3 color channels (like RGB), and may have 1 alpha channel value. If there is no alpha defined for a color model, the alpha value is assumed to be 1. Regions of the image are created with the alpha image map. These regions are manipulated using the blending operations described below. Alpha blending is the process of combining a foreground color with a background color which produces a new blended color. The *alpha* channel describes how much opacity is present in a pixel. It may be completely transparent, completely opaque, or any range of translucency.

Conceptually, when the foreground color is completely opaque, the resulting blended color will be the foreground color. If it is transparent, the blended color will be the color of the background. When the value of the alpha channel is 1, the image is all there, if it is 0, there is no image at all, otherwise it is only partially there. In other words, the alpha value goes from 0 to 1, where full transparency is 0 and opaque is represented by 1. Alpha blending models opacity.

When blending source and destination shapes (Dst and Src), the shape boundaries can be changed with the alpha blending effects. There are a total of 10 standard Porter-Duff operators, but there are 30 possible overlay modes used in CINELERRA-GG. Each is characterized by its value in the four regions: source, destination and both, with the *neither* region always being blank. The source and destination regions can either be blank or filled with the source or destination colors. A specific compositing math formula is used to calculate effect. This is only applicable to RGB; some effort has been made to accommodate YUV, but the effects are not as predictable, and may not be useful.

Below, in figure 12.2, are the results of utilizing the 30 available operations within CINELERRA-GG as listed on a following page. Src is the solid green rectangle and Dst is the solid red rectangle. There are better illustrations of what alpha blending can do, however for consistency sake, these are the results when using standards. For an alternative set of source video illustrations, see Alternative Views of Overlay Modes.

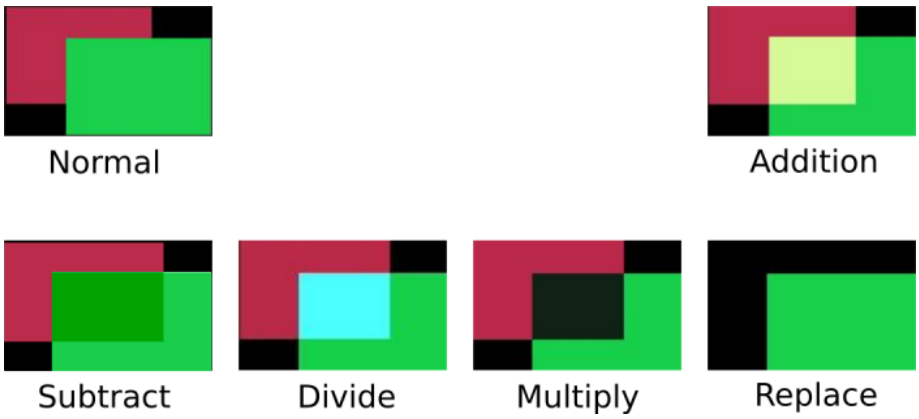


Figure 12.2: *Normal and Arithmetic overlays*

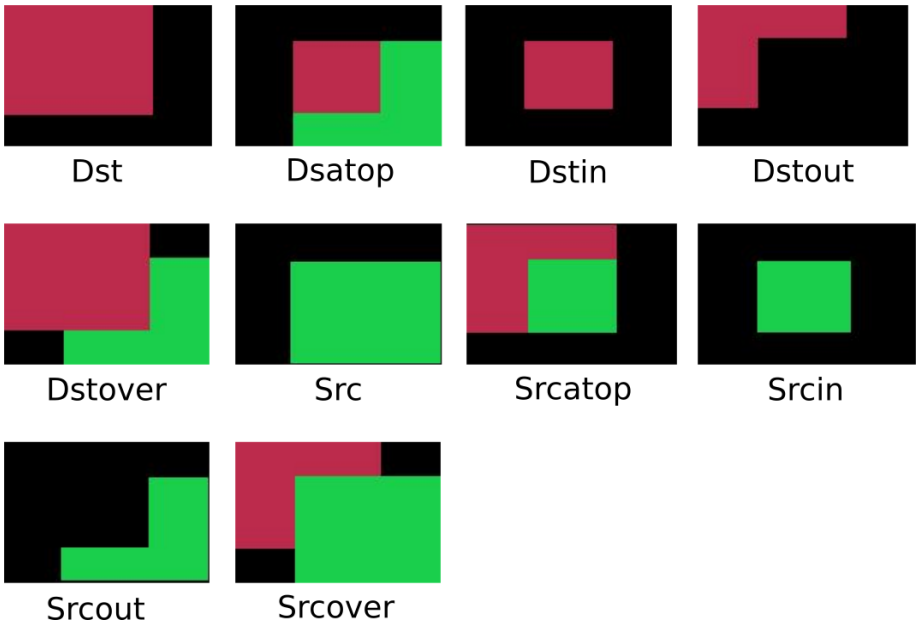


Figure 12.3: *Porter Duff overlays*

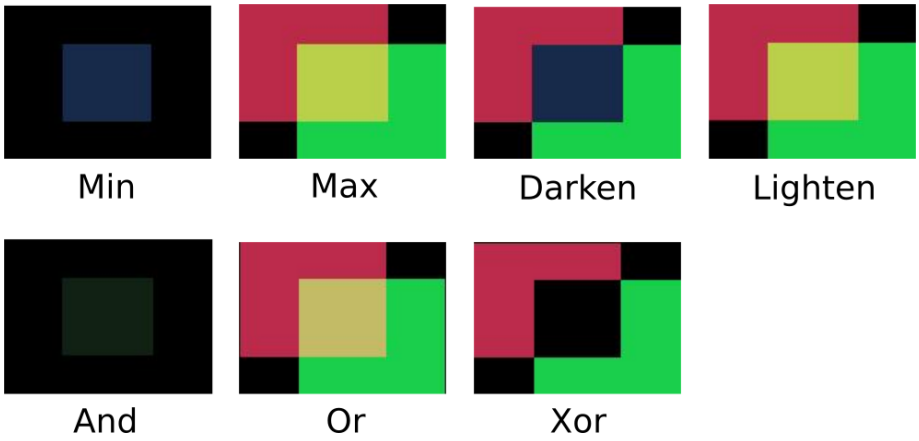


Figure 12.4: *Logical overlays*

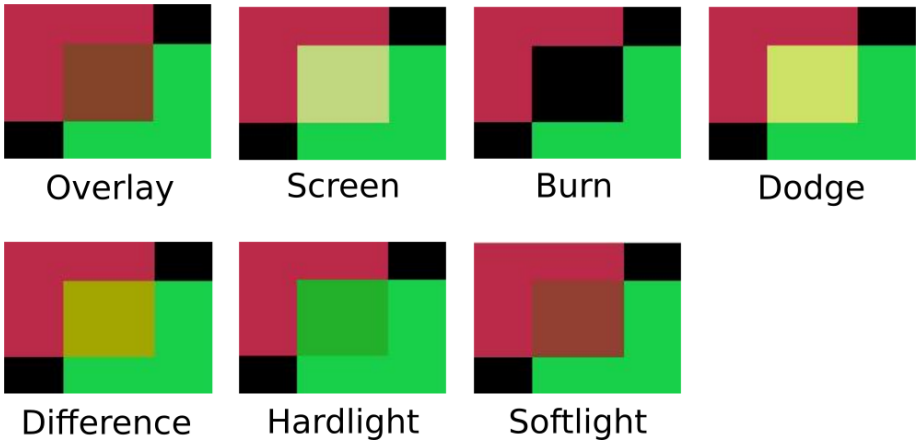


Figure 12.5: *Graphic Art overlays*

12.1 Math algorithms

The implementation math forms are subsequently listed, where:

```
Legend:
D = Destination
S = Source
a = alpha
c = chroma (color)
|| = OR (logical operator);
? : = if (true/false) ... then (conditional ternary operator)
```

Each line describes a pair with the left one for alpha and the right one for chroma.

Normal

NORMAL: $[Sa + Da \times (1 - Sa), Sc \times Sa + Dc \times (1 - Sa)])$

Arithmetic:

ADDITION: $[(Sa + Da), (Sc + Dc)]$

SUBTRACT: $[(Sa - Da), (Sc - Dc)]$

MULTIPLY: $[Sa + Da - Sa \times Da, Sc \times (1 - Da) + Dc \times (1 - Sa) + Sc \times Dc]$

DIVIDE: $[Sa + Da - Sa \times Da, Sc \times (1 - Da) + Dc \times (1 - Sa) + Sc/Dc]$

REPLACE: $[Sa, Sc]$ (fade $= 1$)

Porter-Duff

DST: $[Da, Dc]$

DST_ATOP: $[Sa, Sc \times (1-Da) + Dc \times Sa]$

DST_IN: $[Da \times Sa, Dc \times Sa]$

DST_OUT: $[Da \times (1-Sa), Dc \times (1-Sa)]$

DST_OVER: $[Sa + Da - Sa \times Da, Sc \times (1-Da) + Dc]$

SRC: $[Sa, Sc]$

SRC_ATOP: $[Da, Sc \times Da + Dc \times (1-Sa)]$

SRC_IN $[Sa \times Da, Sc \times Da]$

SRC_OUT: $[Sa \times (1-Da), Sc \times (1-Da)]$

SRC_OVER: $[Sa + Da - Sa \times Da, Sc + (1-Sa) \times Dc]$

Logical

MIN: $[min(Sa, Da), MIN(Sc, Dc)]$

MAX: $[max(Sa, Da), MAX(Sc, Dc)]$

LIGHTEN: $[Sa + Da - Sa \times Da, Sc \times (1-Da) + Dc \times (1-Sa) + max(Sc \times Da, Dc \times Sa)]$

DARKEN: $[Sa + Da - Sa \times Da, Sc \times (1-Da) + Dc \times (1-Sa) + min(Sc \times Da, Dc \times Sa)]$

AND: $[Sa \times Da, Sc \times Dc]$

OR: $[Sa + Da - Sa \times Da, Sc + Dc - Sc \times Dc]$

XOR: $[Sa + Da - 2 \times Sa \times Da, Sc \times (1-Da) + Dc \times (1-Sa)]$

Graphical Art

OVERLAY: $[Sa+Da-Sa \times Da, Sc \times (1-Da)+Dc \times (1-Sa)+2 \times Dc < Da \quad ? \quad 2 \times Sc \times Dc : Sa \times Da - 2 \times s(Da-Dc) \times (Sa-Sc)]$

SCREEN: $[Sa + Da - Sa \times Da, Sc + Dc - (Sc \times Dc)]$ (same as OR)

BURN: $[Sa+Da-Sa \times Da, Sc \times (1-Da)+Dc \times (1-Sa)+Sc \leqslant 0 \parallel Sc \times Da + Dc \times Sa \leqslant Sa \times Da \quad ? \quad 0 : (Sc \times Da + Dc \times Sa - Sa \times Da) \times Sa/Sc]$

DODGE: $[Sa+Da-Sa \times Da, Sc \times (1-Da)+Dc \times (1-Sa)+Sa \leqslant Sc \parallel Sc \times Da + Dc \times Sa \geqslant Sa \times Da \quad ? \quad Sa \times Da : Dc \times Sa/(1-Sc/Sa)]$

DIFFERENCE:
$[Sa + Da - Sa \times Da, Sc \times (1-Da) + Dc \times (1-Sa) + abs(Sc \times Da - Dc \times Sa)]$

HARDLIGHT: $[Sa+Da-Sa\times Da, Sc\times(1-Da)+Dc\times(1-Sa)+2\times Sc < Sa \quad ? \quad 2\times Sc\times Dc : Sa\times Da-2\times(Da-Dc)\times(Sa-Sc)]$

SOFTLIGHT: $[Sa+Da-Sa\times Da, Sc\times(1-Da)+Dc\times(1-Sa)+Da > 0 \quad ? \quad (Dc\times Sa+2\times Sc\times(Da-Dc))/Da : 0]$

12.2 Description

The previous math forms are the only truly accurate description of each blending operation, but short descriptions are below where *Source* is the output from the next track and *Destination* is the output from the lower track stacking. Blending starts with new Source and combines it with the previous render stack output, which is referred to as Destination. The new output becomes the next Destination and the next up stack level becomes the new Source. Source is above; Destination is below.

Some notable items that affect the results are listed here.

1. So as not to be surprised by the results, set your Video Driver to X11, not X11-OpenGL, because when rendering your video no graphics library is used. Rendering is handled in software and there is sometimes anomaly differences when using the OpenGL driver.
2. The Graphic Art group operates principally on color, and the others operate principally on alpha.
3. When using the Title plugin with text, blending or changes in the text will only take affect in Subtract, Divide, and Difference. This matches the implmentation of the original Cinelerra version.
4. If comparing results with the original Cinelerra version for Subtract and Divide, you may have to switch the order of the source and destination tracks to see the same output. If you do not want to switch tracks, you can use the *overlay* plugin as shared and change the Layer order and/or Output layer. The reason for having to switch tracks to get equivalent output is because of differences in the math formulas used. Subtract exactly matches the original Cinelerra version if you switch the 2 tracks; Divide matches in one order only.
5. Instead of using the patchbay overlays, you may prefer to use the *overlay* plugin instead.
6. If you see undesirable results, always check under Settings->Format and check your Color Model as that will vary or interfere with the look.

Normal Group

Normal: Normal mode is the default layer mode. The result color is the source color. The layer on top covers the layers below it. If you want to see anything below the top

layer when you use this mode, the layer must have some transparent areas. It is *stacked on top*. Math formula used is different than that used by Gimp; there is no SVG equivalent.

Arithmetic Group:

Standard numerical operations.

Addition: The source is added to the destination and replaces the destination. Addition mode is very simple - the pixel values of the upper and lower layers are added to each other. The resulting image is normally brighter. The equation can result in color values greater than 255, so some of the light colors may be clipped to the maximum value of 255. Math formula is the same as that used by SVG but slightly different than Gimp.

Subtract: Subtract mode reduces the pixel values of the upper layer by the pixel values of the lower layer. The resulting image is normally darker. You might get a lot of black or near-black in the resulting image. The equation can result in negative color values, so some of the dark colors may be clipped to the minimum value of 0. Math formula used is different than that used by the Gimp "Default" version, but you can get the same as the "Legacy" version if you switch the 2 tracks or use the *overlay* plugin and switch to Top first. There is no SVG equivalent.

Multiply: The source color is multiplied by the destination color and replaces the destination. The resulting color is always at least as dark as either the source or destination color. Multiplying any color with black results in black. Multiplying any color with white preserves the original color. Math formula is the same as used by SVG and Gimp.

Divide: Divides source color by destination color. If the source color is white, the result color is the underlying color. The resulting image is often lighter. Math forumula is different than that of Gimp, but by changing the order of the Source and Destination tracks you can get the same results in some cases (this is true for the original Cinelerra version also). There is no SVG equivalent.

Replace: Replace mode will cause any existing destination to be replaced by the source media. Math formula used is the same as used by Gimp; there is no SVG equivalent.

Porter-Duff Group

Industry standard compositing operators.

DST: The destination is left untouched. Only the destination will be present in the output. Math formula is the same as that used by Porter-Duff and SVG; there is no Gimp equivalent.

DST_ATOP: The part of the destination lying inside of the source is composited over the source. The destination outside the source is dropped. You will see the existing canvas is only kept where the shapes overlap. Math formula is the same as that used by Porter-Duff and SVG; there is no Gimp equivalent.

DST_IN: The part of the destination lying inside of the source is displayed. This effectively allows for cutting out the background using the current layer/feature as a mask. You will see the new shape is drawn only where both the source and the destination overlap. Math formula is the same as that used by Porter-Duff and SVG; there is no Gimp equivalent.

DST_OUT: The part of the destination lying outside of the source is displayed. This is basically a reverse mask, compared to destination-in. You will see the existing content is kept where it doesn't overlap the source. Math formula is the same as that used by Porter-Duff and SVG; there is no Gimp equivalent.

DST_OVER: The destination is composited over the source. Math formula is the same as that used by Porter-Duff and SVG; there is no Gimp equivalent.

SRC: The source is copied. The destination is not used as input. Only the source will be present in the output. Math formula is the same as that used by Porter-Duff & SVG; there is no Gimp equivalent.

SRC_ATOP: The part of the source lying inside of the destination is composited over the destination. The source outside the destination is dropped. You will see the existing canvas is only kept where the shapes overlap. Math formula is the same as that used by Porter-Duff and SVG; no Gimp equivalent.

SRC_IN: The part of the source lying inside of the destination is displayed. This effectively allows the destination to act as a mask for the layer/feature being shown. You will see the new shape is drawn only where both the source and the destination canvas overlap. Math formula is the same as that used by Porter-Duff and SVG; there is no Gimp equivalent.

SRC_OUT: The part of the source lying outside of the destination is displayed. This is basically a reverse mask, compared to source-in. You will see the new shape is drawn where it doesn't overlap the destination. Math formula is the same as that used by Porter-Duff and SVG; there is no Gimp equivalent.

SRC_OVER: The source is composited over the destination. Math formula is the same as that used by Porter-Duff and SVG; there is no Gimp equivalent.

Logical Group

Min: The output color is the component-wise minimum value of the source and destination colors. There is no SVG or Gimp equivalent math formula.

Max: The output color is the component-wise maximum value of the source and destination colors. There is no SVG or Gimp equivalent math formula.

Lighten: Selects the lighter of the destination and source colors. The destination is replaced with the source when the source is lighter, otherwise it is left unchanged. Completely black layers have no effect on the final image and completely white layers result in a white image. Math formula is the same as used by SVG and Gimp's *lighten only*.

Darken: Selects the darker of the destination and source colors. The destination is replaced with the source when the source is darker, otherwise it is left unchanged. Completely white layers have no effect on the final image and completely black layers result in a black image. Math formula is the same as used by SVG and Gimp's *darken only*.

And: This operation intersects source and destination. If either image is not white, it down factors the other image. Usually you will end up with a lot of darker areas. There is no SVG or Gimp equivalent math formula.

Or This operation is the union of the source and the destination. If either image is not black, it adds to the other image. Usually you will end up with a lot of lighter areas. Math formula is the same as that used by "Screen" operation in SVG; there is no Gimp equivalent.

Xor: The union of the source and destination with the intersection removed. Results in black if they are equal. Shapes are made transparent where both overlap, and drawn normal everywhere else. Math formula used is that used by Porter-Duff; there is no Gimp equivalent and SVG formula differs.

Graphical Art Group

Typical operations from popular *paint* packages.

Overlay: Multiplies or screens the colors, dependent on the destination color. Source color overlay the destination while preserving its highlights and shadows. The destination color is not replaced, but is mixed with the source color to reflect the lightness or darkness of the destination. Dark parts on the base layer become darker, and light parts become lighter. Math formula is same as SVG and Gimp.

Screen: The source and destination are complemented and then multiplied and then replace the destination. The resultant color is always at least as light as either of the two constituent colors. Screening any color with white produces white. Screening any color with black does not change the other layer. Using the Screen mode is similar to projecting multiple photographic slides simultaneously onto a single screen. A lighter picture is the result. Math formula is the same as used by SVG and Gimp. This is identical to *OR*.

Burn: Darkens the destination color to reflect the source color. Burning with white produces no change. In photography, burning is a technique used in a darkroom to increase the exposure in particular areas of the image. This brings out details in the highlights. Math formula is the same as used by SVG and Gimp.

Dodge: Brightens the destination color to reflect the source color. Dodging with black produces no change. The result color is a lightening of the source color to reflect the underlying layer color by decreasing the contrast. If the source color is pure black, the result color is the underlying color. In photography, dodging is a technique used in a darkroom to decrease the exposure in particular areas of the image. This brings out details in the shadows. Math formula is the same as used by SVG and Gimp.

Difference: Subtracts the darker of the two constituent colors from the lighter. This is the same as *Subtract* except that the absolute value is used. *Subtract* colors which would be black due to clipping become visible instead. Differencing with white inverts the destination color. Differencing with black produces no change. Math formula is the same as used by SVG and Gimp.

Hardlight: Multiplies or screens the colors, dependent on the source color value. If the source color is lighter than 0.5, the destination is lightened as if it were screened. If the source color is darker than 0.5, the destination is darkened, as if it were multiplied. The degree of lightening or darkening is proportional to the difference between the source color and 0.5. If it is equal to 0.5 the destination is unchanged. Using pure black or white produces black or white. The effect is similar to shining a harsh spotlight on the destination. This mode is useful for creating the appearance of shadows on a layer. You might use this mode to combine two photographs and obtain bright colors and sharp edges. Math formula is the same as used by SVG and Gimp.

Softlight: Darkens or lightens the colors, dependent on the source color value. If the source color is lighter than 0.5, the destination is lightened. If the source color is darker than 0.5, the destination is darkened, as if it were burned in. The degree of darkening or lightening is proportional to the difference between the source color and 0.5. If it is equal to 0.5, the destination is unchanged. Using pure black or white produces a distinctly darker or lighter area, but does not result in pure black or white. The effect is similar to shining a diffused spotlight on the destination. A layer with pure black or white becomes markedly darker or lighter, but does not become pure black or white. Soft light is not related to “Hard light” in anything but the name, but it does tend to make the edges softer and the colors not so bright. Math formula is the same as used by Gimp; SVG formula differs.

12.3 Alternative Views of Overlay Modes

An alternative set of source videos used to provide a different real world view is shown here to help illustrate alpha blending. They use a landscape photo as one and the other is half of a color and half of a black and white gradient photo. First is shown the Aritmetic, Logical, and Graphic-Art views. After that the Porter Duff overlays are shown using the same source videos but with each of the Source videos offset in a different direction so that the generated output is more obvious.

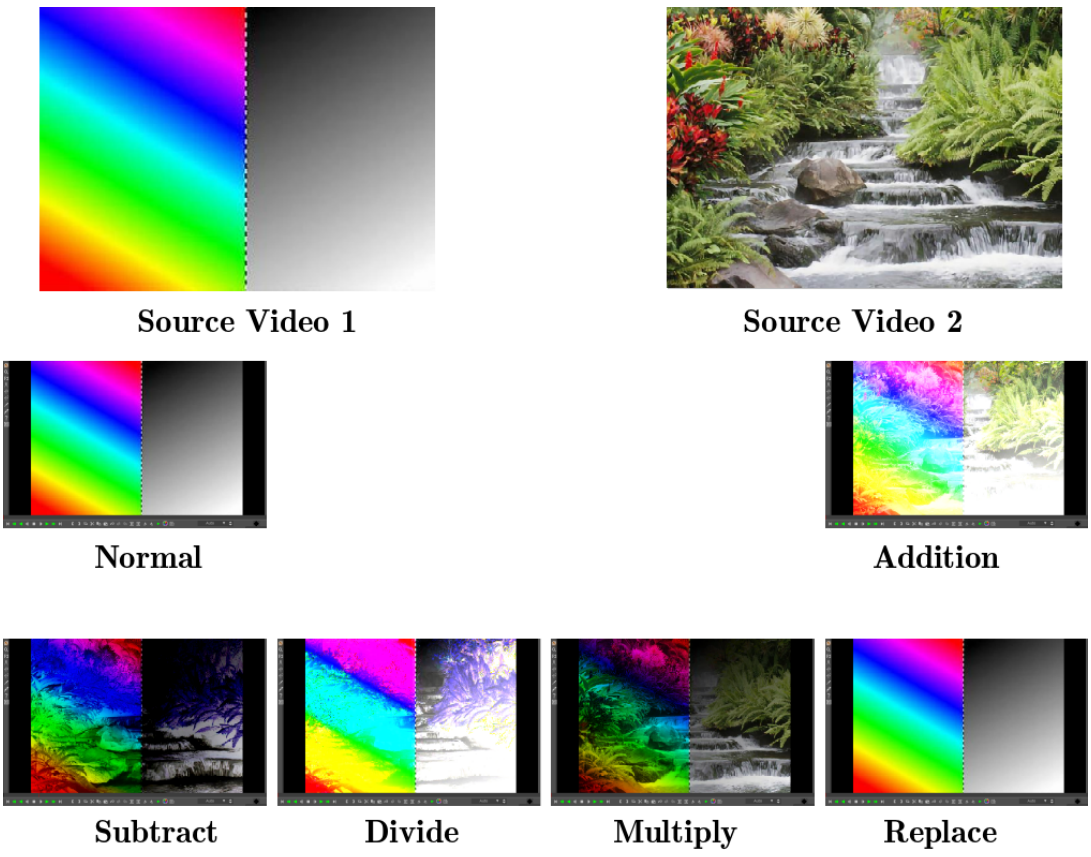


Figure 12.6: *Normal and Arithmetic overlays*

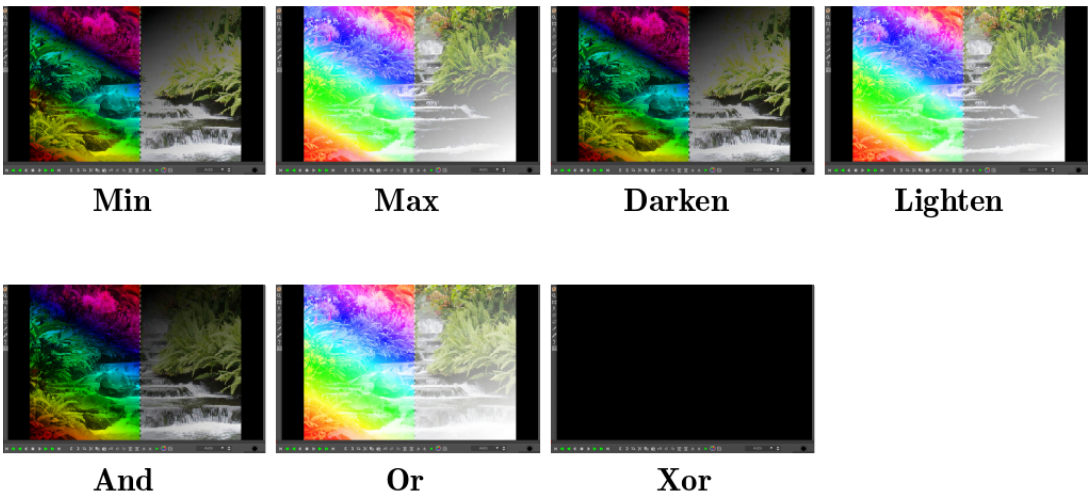


Figure 12.7: *Logical overlays*

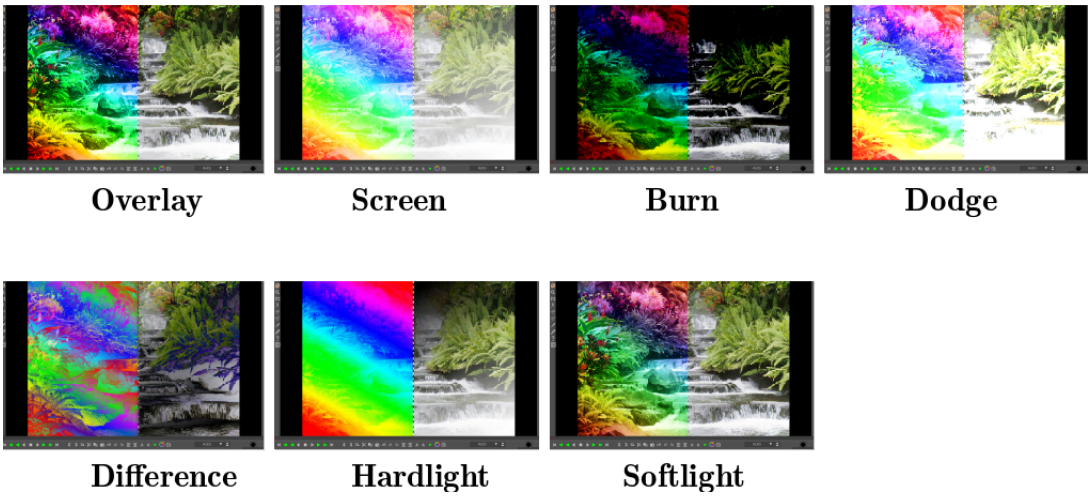


Figure 12.8: *Graphic Art overlays*

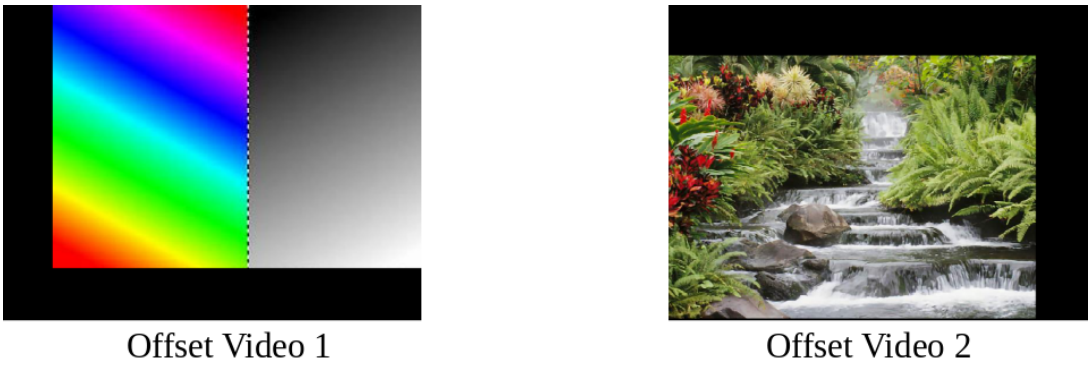


Figure 12.9: *Offset source videos to illustrate Porter Duff overlays*

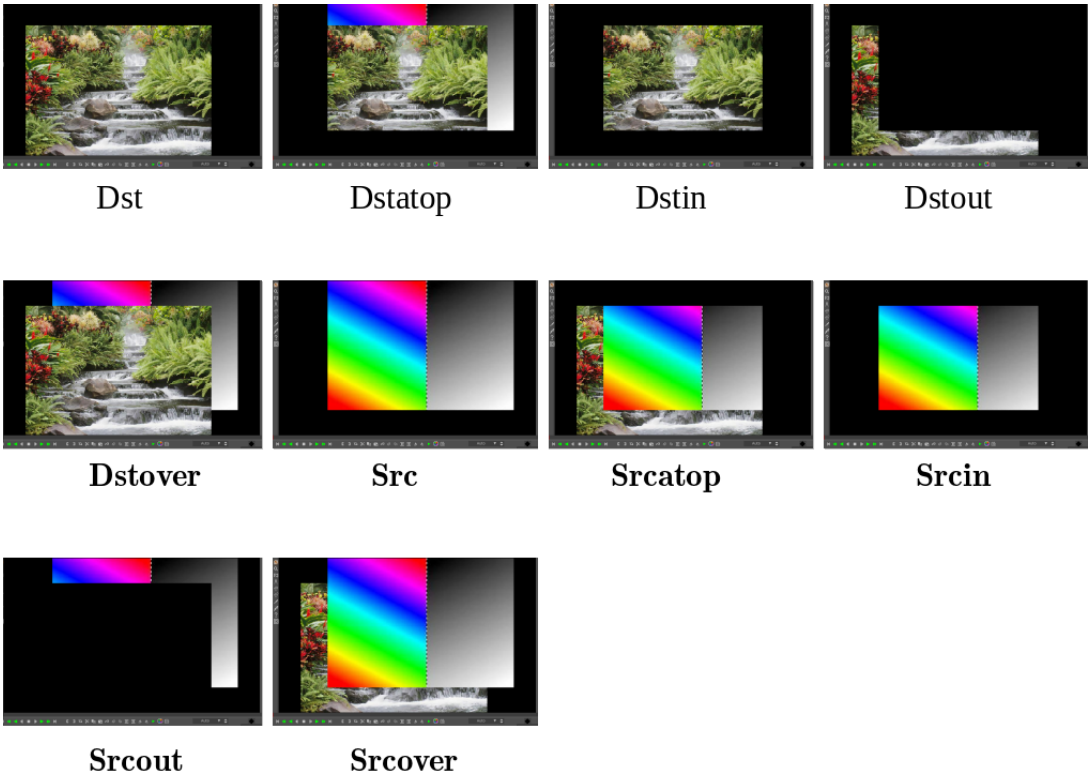


Figure 12.10: *Porter Duff overlays*

12.4 Problems with transparencies

Often in applying overlays, or some other type of blend, we get unexpected results that can look wrong. Sometimes the cause may be the different formula used by CGG compared to those in Gimp or other software. Other times they are a result of setup or order as listed next. Keep in mind that blends can be applied not only from the Patchbay but also as an Overlay plugin (see Overlay (Video)) and the *Blend Algebra* plugin see Blend Algebra and Blend Program. Here are some factors that can lead to unexpected results:

1. The colors of the 2 tracks that will be blended; we often consider only these channels to guess the final result.
2. The Alpha channel of the 2 tracks that can interact with each other in ways that are intuitively (but not mathematically) unexpected.
3. The choice of Source track and Destination track (*top* or *bottom*).
4. Overlays effect generally depends on the *colorspace* of the project - YUV can differ from RGB. With the *Blend Algebra* plugin this dependency is addressed; the necessary colorspace conversion can be done automatically.
5. Overlays can depend on implicit clipping. In `RGBA-FLOAT` no clipping is done; in `RGBA8888` clipping to 8-bit unsigned cannot be avoided per definition. The *Blend*

Algebra plugin addresses this dependency also, clipping for float colorspace can be switched on explicitly, and then the results for float and RGBA8888 should be identical.

6. Unclipped overlays can be displayed differently depending on the graphic driver (*X11*, *X11-Xv*, *X11-OpenGL*, etc.). The OpenGL driver could be more robust in these cases; it runs perhaps under control of some CINELERRA-GG shader and seems to clip the bounds correctly. However, all rendering is done in software without the use of OpenGL so be sure to check results without OpenGL.

7. The presence of a third black background track, (or even the color of the `Composer Background Color`, which in CINELERRA-GG is black by default but whose color can be varied manually) which can interfere with viewing the blend in the Compositor and show unexpected results (see MantisBT #559).

8. If the footage material in a track has transparency by itself, it might be modified by overlay formula and develop some color hidden by transparency before overlay was applied.

9. Overlays can be applied at two places: via the *Overlay* plugin or the *Blend Algebra* plugin in which sequence in the video processing pipeline plugins are processed, is defined; and in the `Patchbay` in which sequence in the video processing pipeline plugins are processed, is not defined.

10. There may be *masks*. The mask can be applied either after plugins, or before them. Results of the *Overlay* plugin depend on it.

11. There is `fader` control. Probably, fader is applied after all plugins and overlays, but this order has not been verified.

12. There are two buttons in the patchbay, `Play track` (can be switched off) and `Don't send to output` (can be switched on). In both cases video from the affected track disappears, but the background left is different in both cases.

13. Effects from some other plugins can have transparent parts (*Titler*, for example).

Ultimately, if we get a result that looks wrong to us, we need to look at the formula for that blend and calculate our mathematical result. Then compare our calculation with the result we see on the screen. In principle, you could attach *Blend Algebra* in addition to *Overlay* plugin, switch one of them on, another off, change the formula in *Blend Algebra* and see how the modifications affect the result, and even print and analyze the intermediate color values. This can be most helpful if you use some specially drawn regular color patterns instead of your working footage. Another possibility is to attach the *Blend Program* plugin which modifies any given tracks directly.

Chapter 13

Capturing and Recording Media

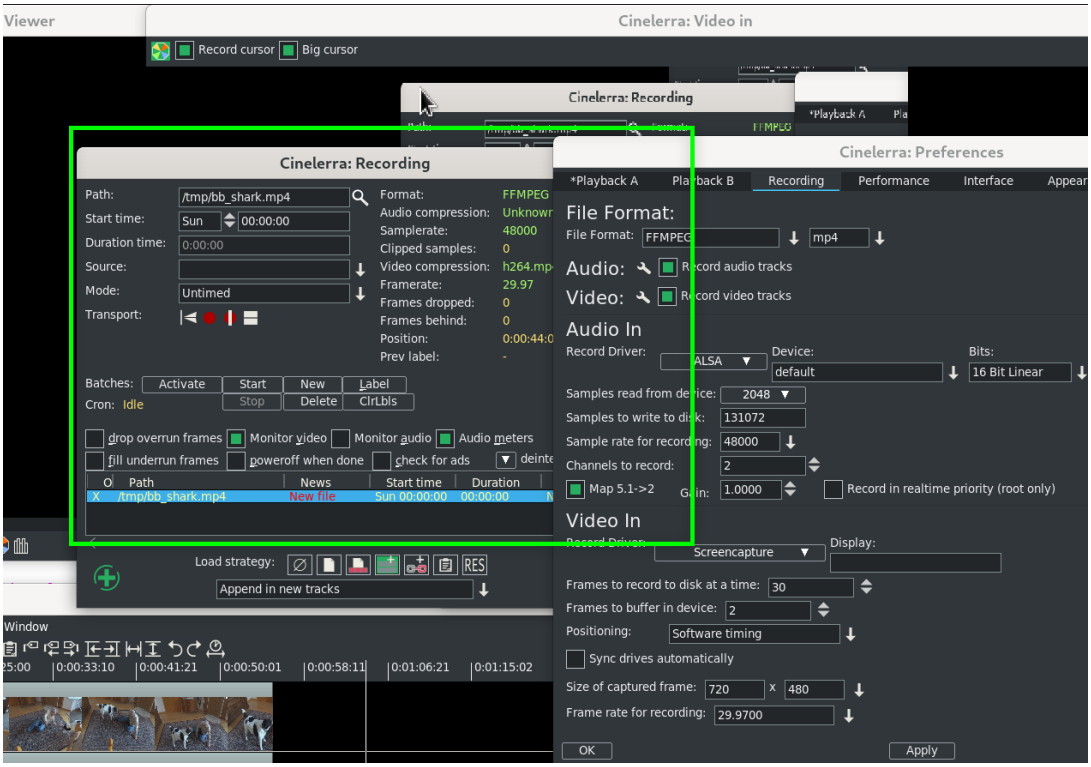


Figure 13.1: *Actual Recording Session - note the bright green boundary box and the big arrow cursor within the "Video In" window*

This section covers the areas of capturing media from the web or television by the use of recording.

Access the Record function via `File` → `Record`... The recording application really only does one thing, capture media from some external source and write it onto a file in a specified way.

Path:	output media file path	
Start time:	weekday/time of day	to begin capture
Duration:	hrs:mins:secs	until end of capture
Source:	channel/device input	selection of input subsource
Mode:	timed	use start time/duration
	untimed	use transport controls

The media file will be written using the format and codec specified in the `Settings → Preferences → Recording tab`, which you will need to set up first. See the Settings/Preferences section for parameters. Only ffmpeg can record both audio and video simultaneously, and some ffmpeg formats require too much cpu to do a realtime compression. A setting which is more likely to be usable (requires less cpu/memory for realtime encoding) is `ffmpeg → qt → mp4.qt` with bitrates like *audio*/256000 & *video*/6000000.

13.1 Record Web Media in real-time

Below describes the necessary steps for recording freely available media from the internet for your own personal use. You have to be on a system using pulseaudio, such as ubuntu, fedora, or centos.

1. Start CINELERRA-GG and select `Settings → Preferences → Recording` From a terminal (with a wide text window) run: `pactl list`. You will see all of the audio sources and sinks on your system. Identify the source associated with the normal output your system uses. Example: *Source* #1. Locate the source which monitors your normal audio output. For example: *front stereo*.
2. Choose a File Format and File Type (for example: FFMPEG & qt).
   - Check *Record audio tracks*. Click `Audio wrench` tool:
     - Select *mp4*, for example.
     - Set *Bitrate* (256000 is a reasonable number).
   - Check *Record video tracks*.
   - Click *Video wrench* tool:
     - Select *mp4* video, for example.
     - Set `Bitrate` 1000000 – 4000000 (bigger=better, but more cpu)
   - Select `Audio In → Record Driver: ALSA` (or whatever you have instead)
     - Set the *Bits*: *16 Bit Linear* (probably, because reasonable and fast)
     - Select *Device*: the source from above *pactl* list search, for example Source #1
     - Samples read from device: aprox dev buffer size ($2k - 16k$ probably)

  - Samples to write to disk: 131072 (a good size)
  - Sample rate for recording: 44100 (will automatically change)
  - Channels to record 2 (probably)
  - Uncheck Map 5.1 → 2
  - Gain 1.0
- Select `Video In` → `Record Driver`: `Screencapture`
  - Set the *Display*: leave blank (probably) or use ": 0.0" (default screen/display)
  - Frames to record to disk at a time: 30 (a good number)
  - Frames to buffer in device: $(2-6)$ (probably)
  - Positioning: `Software Timing` (important)
  - Uncheck: `Sync drives automatically`
  - Size of captured frame: $720x480$ (this is your choice, actual capture size; $600x320$ youtube)

    This defines a *screen capture* rectangular area on the display.

    Try to make it the size you need to cover the screen playback area.
- Frame rate for recording $23.97fps$ (a good choice, not all choices work)
- OK

When recording from the screen, a large green-colored boundary box appears to allow you to easily frame the screen area to be recorded when you move it around to where you want to position it. In addition there are *record cursor* and *big cursor* check boxes which allow for also recording the cursor and for making the cursor bigger. You can see the bright green boundary box and these checkboxes around the middle at the top in figure 13.1.

Screenshot to illustrate some appropriate settings which are described above/below the image (figure 13.2).

3. Make sure the Compositor window is up since you will want to see the captured media later. From the main window press (lower case "r") (move windows and resize if needed) Using the mouse pointer, mouse over the *Video In* Recording monitor display. Press and hold the left mouse button and drag the display until the desired target screen capture area is properly positioned inside the record monitor *Video In* display. At first this may seem confusing, but keep in mind that what you are actually doing is positioning the portion of your monitor display you want to capture over the *Video In* window.
4. From the *Recording window* select a Path file name for the recording.
5. Select mode untimed.
6. Make sure the *monitor audio* is unchecked else audio becomes looped and is very loud.

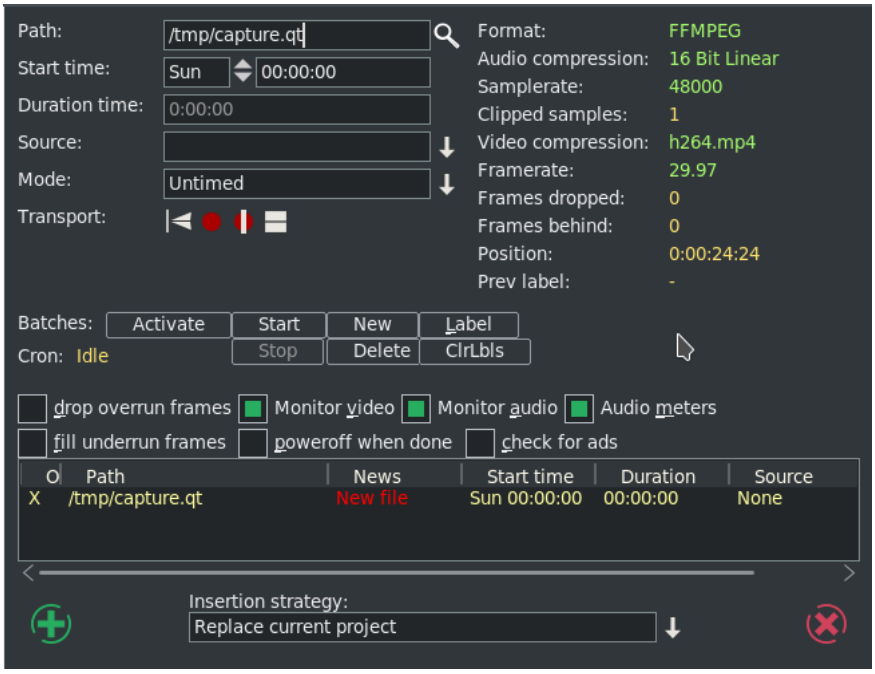


Figure 13.2: *Recording window with typical settings*

7. Press Transport *start recording* (red circle button).
8. Operate the playback of the media source, i.e. start playing the web video.
9. When the media ends, stop on the playback application.
10. Press Transport *stop recording* button (white square button).
11. Before clicking chk-Ok, be sure to set the *Insertion Strategy*.
12. Press chk-Ok to review the capture, or "X" to cancel/dismiss application.

Two things are happening during recording, the data is being written, and it is being rendered. When recording, if video and audio are not in sync, it will usually be video behind audio as opposed to the other way around because video is more difficult to decode. During recording, video frame and audio sample positions are checked and efforts to maintain synchronization are used. Below is an explanation for some of the less obvious settings for recording as shown in the previous screenshot.

Frames dropped more input than cpu(s) can process. This is the number of frames dropped when the system is too slow to keep up with the video. This value serves as a warning that it is behind and is dropping frames to keep up. You might want to consider stopping to make some parameter adjustments and start over to reduce drops.

Frames behind active input buffers. This is the number of frames that the video is behind and it will drop frames to get caught up. *Frames behind* depends on the synchronization source, for example, time as the synchronization. This means that frames are not being written as fast as they are being captured and can occur when the system is too slow to keep up. This situation can happen with older systems or cpus where there is more input then can be processed in a timely manner. So that when there are *Frames behind*, frames will be dropped.

The algorithm for determining how many frames to drop is as follows:

- If the number of frames behind > 3, only 3 frames will be dropped before it will then do 1 frame;
- if the number of frames behind is 2, only drop 1 frame;
- if only 1 frame behind, no frames will be dropped.

Drop overrun frames checkbox when enabled, frames will be dropped. Drops occur when you capture a frame and there is no storage space available to save it, usually because buffer space is exhausted. This situation can occur when there are too many frames coming in too fast and the I/O system is not keeping up in emptying the buffers.

Fill underrun frames checkbox if enabled, duplicate last frame when no input frame ready for output. As the tooltip states *write extra frames when behind*. This is desirable in order to keep audio and video synched with regards to sample and frame rates. Use underrun padding to fill frames when the input capture rate is lower than the output frame rate. Since demand is constant, you have to output something, so the last frame is duplicated as many times as needed to keep up. You should stop and adjust the frame/sample rates to match the input if not filling underruns, then start over. This situation can occur if the input stops for some reason like lightning interrupts the signal or the internet connection is too slow.

Transport controls these control buttons mimic the functions of tape recorders from the old days.

- *Reverse button/left arrow* – remnant from the past; does the same as the Start button really.
- *Red round button* – this is the Start button to start recording.
- *White square* – this is the Stop button to stop recording.
- *Red round button with white line* – start recording in single frame mode. The way this works is to just *take a picture now, take a picture now...* The reason to use this mode is to get a still shot like you would with a camera. Sometimes input is continuous, for example looking at stars with a telescope – surveillance goes on

for hours, but you just want to take a picture now when something of significance interest comes up

Cron The batch recording watcher, cron, is either *Idle* or *Active*. When you start or stop batch recording at specific times, there is a cron thread watching timers to perform the timed action. Idle/Active indicates whether the timers are running.

Position this is a timebase which tracks frames/samples when obeying frame/sample rate. When a recording starts, it resets to 0. Timing is against audio (when available). Audio time and video time are based on position. You can select synchronization time source *Positioning* in:

`Settings` → `Preferences` under the *Recording* tab.

Possible choices for time base are:

- *Presentation Timestamps* – use time code which is in both the audio and video media input stream. Uses these timestamps to sync the 2 streams.
- *Device Position* – this is the device hardware position of where you are. It is usually only on the audio side.
- *Sample Position* – Sample # or frame # divided by frame rate tells you where you are.
- *Software Timing* – usually used for things like YouTube; it just will *take a picture now* . . . *take a picture now* . . . over and over again until you tell it to stop.

Positioning & Timing needs more detailed explanation for complete understanding and application. The overall goal is to maintain media stream timeline synchronization. The reason for providing different *Positioning* options is that different input media may have different timebase standards. Additionally, the input may be damaged. Damaged data can skew the timeline during presentation. The timebase standards make it possible to correctly resynchronize the media presentation to the original time position. For example, the transport layer may have timestamps provided in it. These timestamps record *audio time* and *video time* and are called *presentation timestamps*.

The kind of positioning used depends on the input device and the media format. In the transport stream example, the media has already been coded at least once as a stream and contains position information. In other situations, sample position or device position can provide timeline position. If the input device provides data at a variable or asynchronous rate, then software timers can provide timeline position. Sample position is position from the perspective of software, and device position is position from the perspective of hardware. The difference is buffer time. Device position is usually more accurate, but may not be available, or may contain errors.

Examples of media which usually has a particular type of positioning is:

1. mpeg has timestamps therefore Presentation Timestamps is a good recording choice;
2. raw media streams/screen capture have no timestamps, so tell it time via Software;
3. a web cam may supply data at variable rates, so again you would use *Software timing*;
4. an example of Device Position is audio (timestamps can come off the device);
5. if Software Positioning working for you, try Sample Position for really bad audio.

Other *Recording* settings are more pertinent to capturing and editing broadcast television with all of its many commercials. Their usage will be explained in more detail in a later section.

Label create a label. During capture, put a time marker at the designated spot. When the recording gets pulled in later for review, you will see a green arrow marking the spot that was clicked.

ClrLbls clears all of the previously set labels.

Check for ads check for commercials. In real time, labels are added when a commercial that matches a previous ad is already in the database (the TDB, Traveling Data Base).

One other noted new feature is a new choice for recording - `V4L2 MPEG`. Some digital tuners now stream mpeg data for broadcast data streams. V4l2 has been upgraded to be able to process data from these device streams. Figure 13.3 shows the available options to include v4l2 mpeg.

Some notes for usage on slower CPUs are outlined at appendix Using Screen Capture on slower CPUs as discovered in actual usage.

13.2 Digital Video Broadcasting (DVB)

You can do real-time capture of a full transport stream of Broadcast TV. Record television programs using CINELERRA-GG by setting up ahead of time to start recording later at the specified time. Then you can use CINELERRA-GG to watch later and easily fast forward through the commercials or edit the program, deleting the commercials, to watch uninterrupted. Some details may be slightly different than what it is in the United States. There are 3 requirements you must have to take advantage of this capability.

1. You need to have a suitable DVB Adapter such as the Hauppauge WinTV-HVR 950Q usb adapter. Make sure it is the linux version, not the new version.

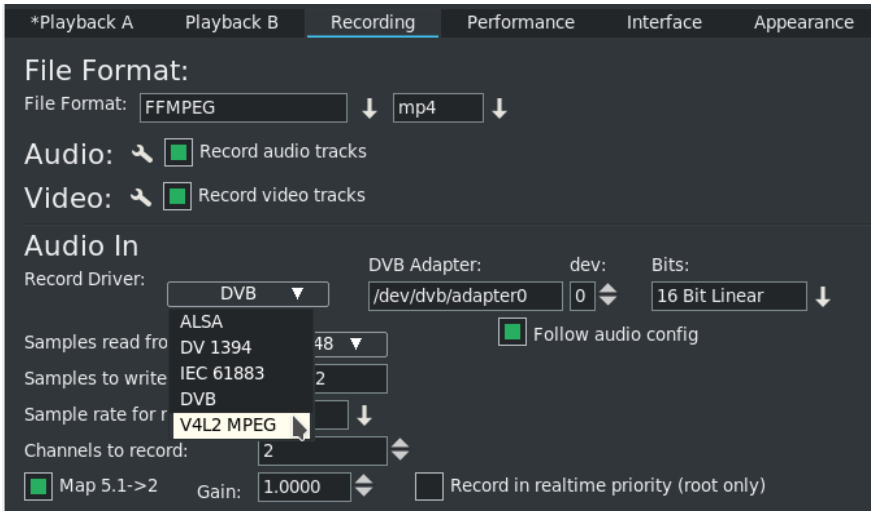


Figure 13.3: *new feature V4L2 mpeg*

2. Your DVB adapter on your PC should connect to a broadcast antenna input.
3. For now, you may have to run as the *root* user.

13.2.1 Overview

There is a lot of detailed explanation below to be thorough, but an overview is presented here.

1. Set up Record Preferences.
2. Scan the Channels.
3. Watch TV by selecting the desired channel.
4. Or set up a batch job to record a program and watch later via Ctrl-Alt-s in the main window.
5. Or record a currently running program by using the buttons in the Record window.
6. Load a previously recorded program.

13.2.2 Initial Setup for DVB Capture

Listed below are the steps for the initial DVB setup. Setup will be preserved until you make changes.

1. Start CINELERRA-GG, and click *Settings* → *Preferences* → *Recording*
2. Select *File Format* checkbox: `Realtime TOC` (Table of Contents)

- It is not necessary to set File Format at this point as it will automatically be set later.
- Check *Record Audio Tracks*.
- Check *Record Video Tracks*.

3. Select `Audio In` → `Record Driver: DVB`
   - Set the `adapter path = /dev/dvb/adapter0` (probably)
   - Set the *adapter dev = 0* (probably)
   - Set the *Bits: 16 Bit Linear* (probably)
   - Check: *Follow audio config*
   - *Samples read from device*: 2048 or approximate dev buffer size ($2k-16k$ probably)
   - *Samples to write to disk*: 131072 (a good size)
   - *Channels to record*: 2
   - Uncheck *Map* 5.1 → 2
   - *Gain*: 1.0
   - Uncheck: *Record in real-time priority*
4. Select in *Video In*: *Record Driver: DVB* (note: File Format above changes to *MPEG Stream*)
   - Set the `DVB adapter` path = `/dev/dvb/adapter0` (probably)
   - Set the adapter *dev* = 0 (probably)
   - Check: *Follow video config*
   - *Frames to record to disk at a time*: 30 (a good number)
   - *Frames to buffer in device*: 2 (2 − 6 probably)
   - *Positioning: Presentation Timestamps* (**this is important and may not be already set**)
   - Uncheck: *Sync drives automatically*
   - *Size of captured frame*: $720x480$ (will automatically change)
   - *Framerate for recording*: 29.97 (will automatically change)
   - Click OK

Figure 13.4 shows the Preferences window as a good example of what you should see.

Scan the channels once the devices are setup by performing the following.

From the main window pulldown use: `File` → `Record` (lower case **r**) & *resize windows*.

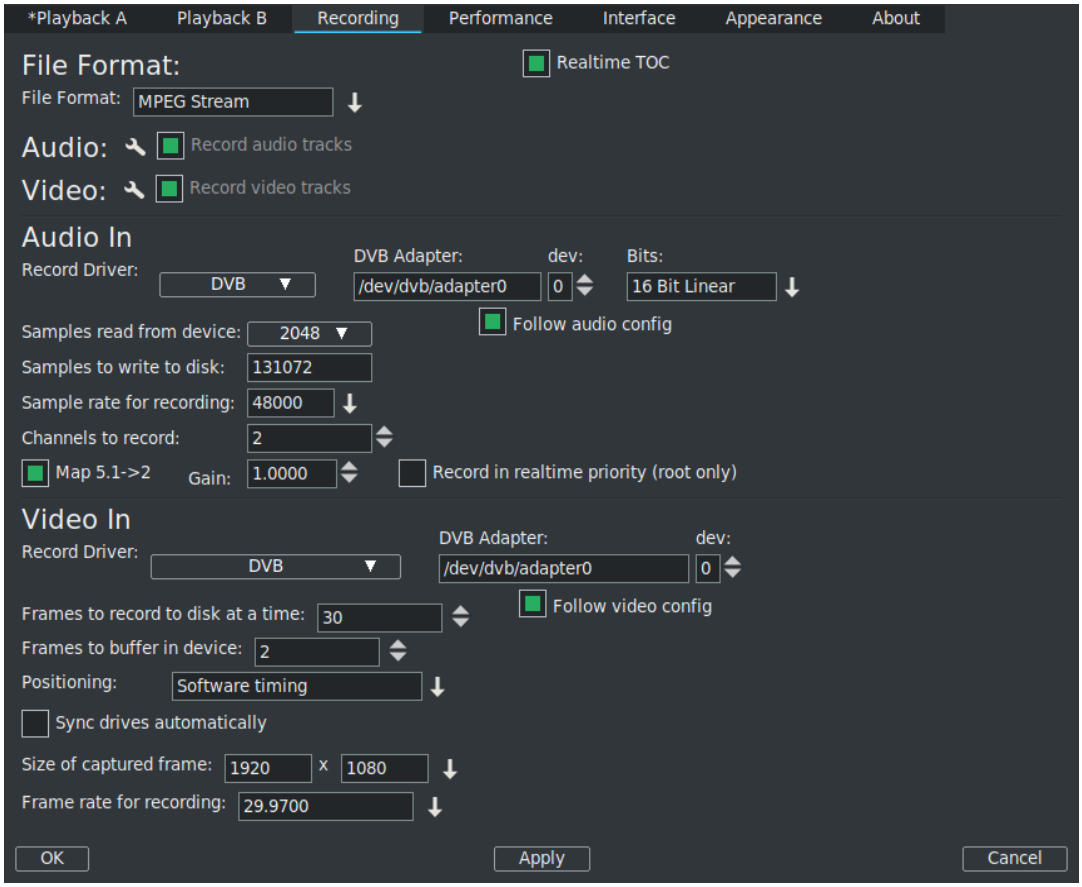


Figure 13.4: *Example of the Recording settings*

- Two windows appear – the *Recording* and the *Video In*. The Recording window should have `Monitor video` checked and *Monitor audio* unchecked. Remember to check the Monitor audio later when you want to listen to a TV program.
- On the *Video In* toolbar, click the *antenna* button (*Edit channels* tool tip).
- When the *Channels* window opens, click *scan* – *Scan confirm* window appears.
- Make sure *Freq table NTSC_DVB*, *Norm NTSC*, and *input0* are chosen automatically.
- Click OK and then wait for scan to end which may take several minutes.

Figure 13.5 on the left shows the Channels window and the popup waiting for scan confirmation. On the right side is the usual Recording window with the Video in window hidden behind.

Figure 13.6 shows scanning in progress. Note the *Video In* with the antenna icon (3rd symbol).

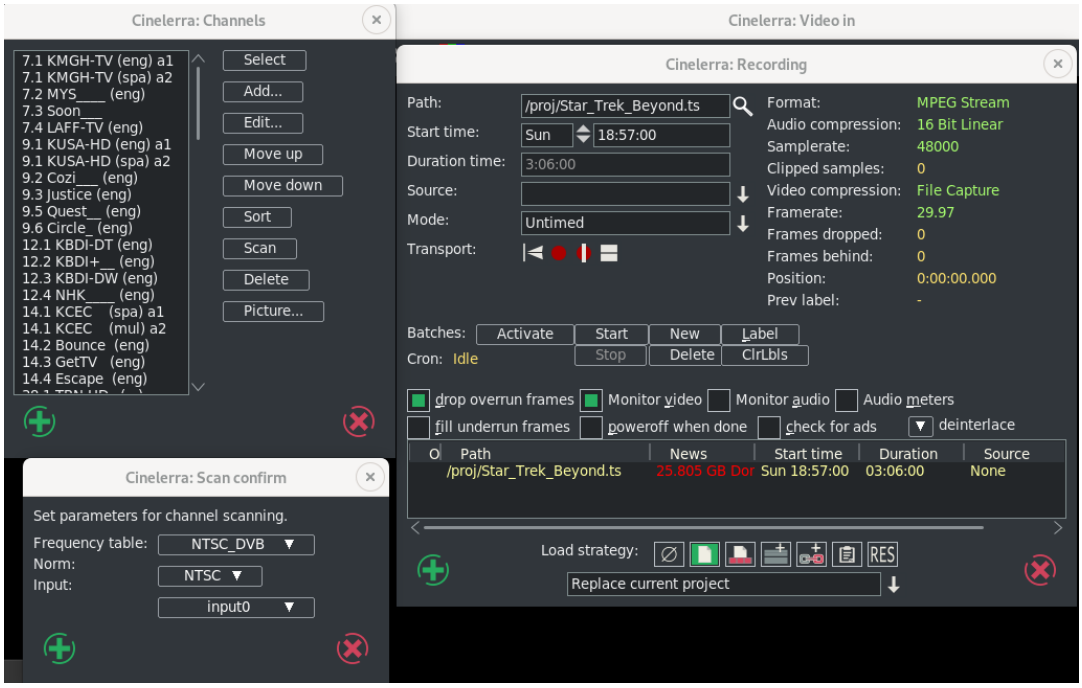


Figure 13.5: *Clockwise: Channels window; Video In (behind); Recording and Scan confirm*

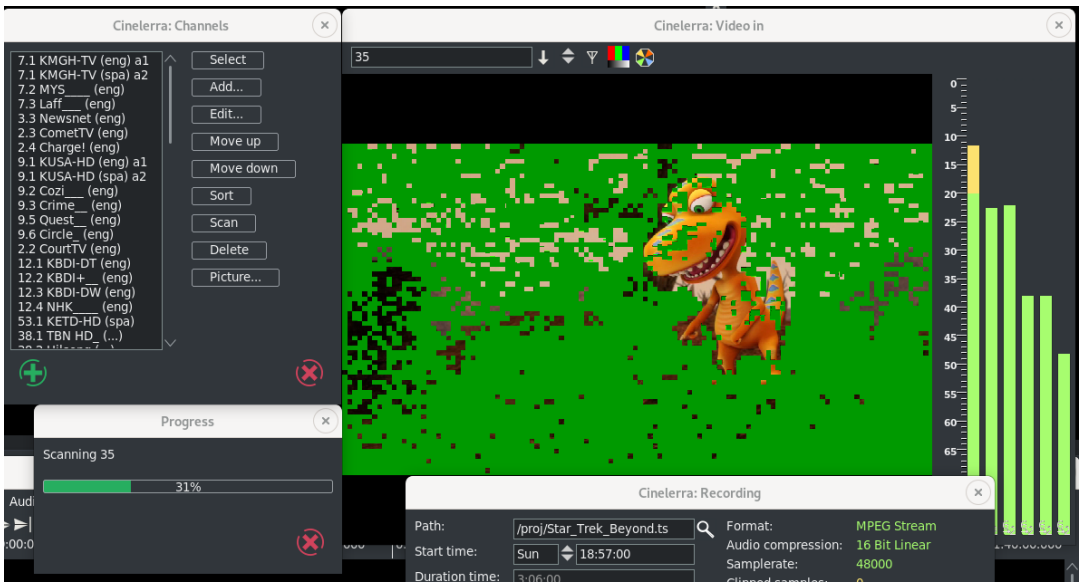


Figure 13.6: *Scanning in progress (31%)*

Any time after the channels have been scanned, you can edit them from the *Recording* window channel selection dialog (shown on left of screenshot above, such as *Sort*). You do not have to scan the channels every time as they are saved in a database referred to as the *channel-db*.

- Click chk-OK when done scanning channels and making any edits you want.

After the channel-db is ready, you can watch any of the selectable channels.

13.2.3 Watching TV using CINELERRA-GG

If you have setup for DVB capture as outlined in the section above, just start here to watch TV.

- From the main window use `File` → `Record` to request the *Recording* application.
- In the *Video In* window, press the down arrow on the left to see the *Channel* pull-down.
- Select the desirable channel, and watch TV. You will have to enable the audio by checking the *Monitor audio* in the Recording window for sound.

Helpful hints: for any computer that you might choose to use that is a little slower, you can improve the available resources for watching TV by unchecking *Audio meters*, checking *drop overrun frames*, and checking *fill underrun frames*. Also, uncheck *Realtime TOC* in the `references` → `Recording` window.

Dismiss the *Recording* window with chk-OK, or "X" to cancel when finished.

13.2.4 Recording TV to Capture at some Future Time via Batch

After the initial setup and channel scan completion, with the Recording application down (so that the `/dev/dvb/adapter0` hardware is not in use) you can scan the DVB channel program data by pressing Ctrl-Alt-s from the main window. It can take a minute or two to scan the channels. The resulting display is a large array of program buttons, with tooltips which are the TV program info text.

Figure 13.7 shows *Channel Info* window after the Ctrl-Alt-s completes scanning TV program data.

Now if you mouse over a program button, you see the TV program's info. Each of the programs is represented by a button. If you press a button, it preloads the program recording information into the batch record box below. In the bottom right corner box, you should

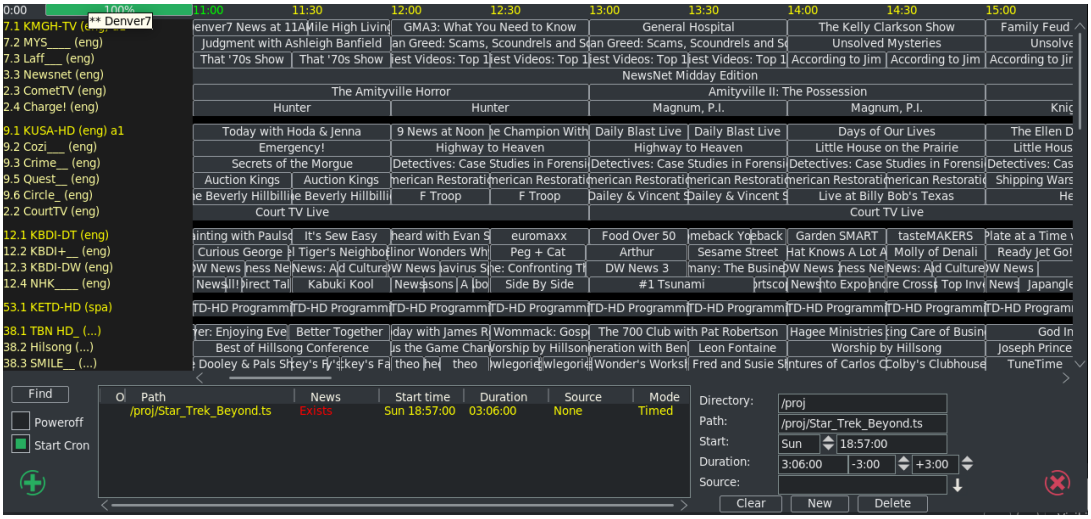


Figure 13.7: *Channels Info window with many TV channels buttons*

set the *Directory* path of where to write the recording before pressing a button, so that the files will be written to an area of sufficient size to handle the recording (9.5Gb/hr). The two little spinboxes on the Duration line in the right bottom corner are leadin/leadout times to start recording a few minutes early, and end a few minutes late to prevent losing program endpoints. Instead of setting it to the original default of 0, you might want to start 3 minutes (−3.00) before and stop 3 minutes after (+3.00). Also, be sure your computer has the correct time or it will miss. Check to make sure the *Start Cron* option is checked (default). The batch job will start monitoring the channel to be ready for capture once you chk-OK and you will see the *Recording* window and the *Video In* window come up. The computer will power off after completion if the *Poweroff* checkbox is selected (not default). The job line of the active capture in the batch bay will flash continuously once the recording is started.

In the Channel Info window, lower left hand corner above, there is a `Find` button which brings up a textbox so you can type in a TV program. When you double click on that program name, the cursor moves to the center of that program name in the list of programs in the window, highlights that program, and the characters that were white on black, turn yellow.

13.2.5 Recording Immediately to Capture a current TV program

If you have setup for DVB capture as outlined in the first section above, just start here to record a TV program that is currently playing and stop it whenever you want. Media files can be directly captured from the *Recording* window to include *Start*, *Stop*, and then loaded via chk-OK.

- Bring up the *Recording* window via the main window `File` → `Record` (lower case "r").

- Set the `Path` to the write file Path; be sure there is enough disk space ($9.5Gb/hr$).
- Set *Mode* pulldown to *Untimed* found in the upper left corner area of the window.
- Press transport *start recording* button (red round button).
- Path will flash and update in batch window and remain flashing while recording.
- Press transport *stop recording* button (white square button).
- Before clicking chk-OK, be sure to set the *Insertion strategy*.
- Pressing chk-OK attempts to load any newly recorded media data for review.
- Or press *X-cancel* to terminate and dismiss the *Recording* application.

Figure 13.8 shows the recording of a currently running broadcast TV program news report.

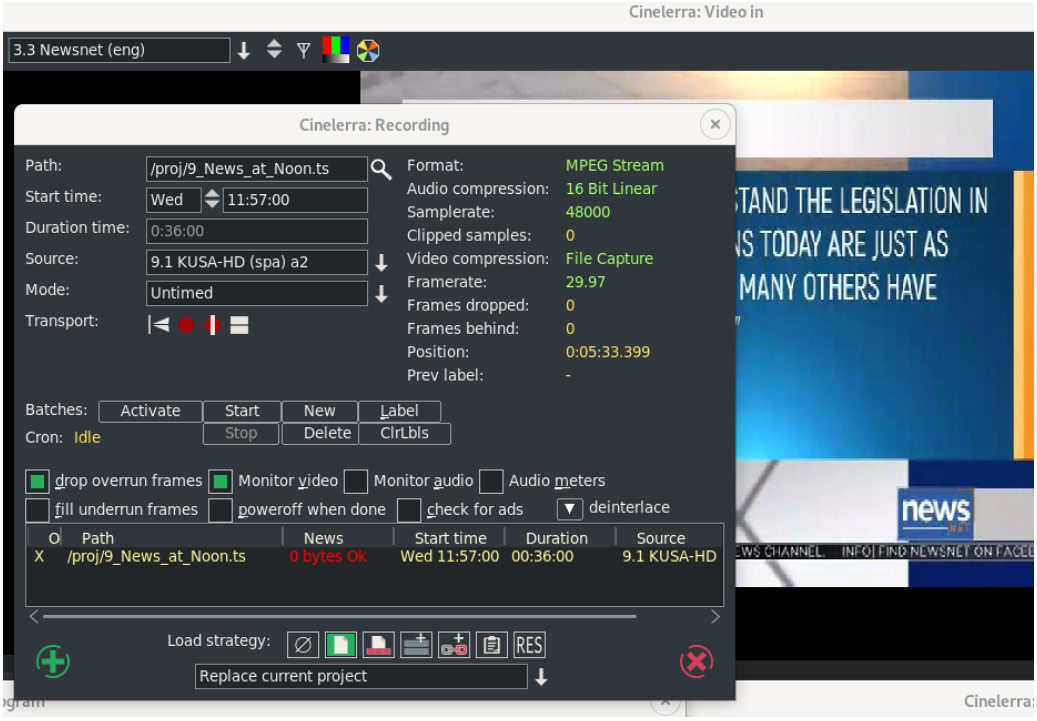


Figure 13.8: *Recording in real-time a TV news report*

13.2.6 Record “on the fly” Table of Contents (TOC)

If the `Settings` → `Preferences` → `Recording` tab is selected, and the *video Record Driver* is changed to *DVB*, the file format is switched to *MPEG Stream* automatically (note that the *audio Record Driver* must also be set to *DVB* for broadcast recording). This kind of record stream can be captured by simply writing the hardware capture stream directly to the record capture asset file path. But by checking the *Realtime TOC* in the `Preferences`

→ `Recording` window, it is possible to scan the stream data *on the fly* on its way to being written while the asset is being captured . The scan is used to generate a TOC – Table of Contents file in the `$HOME/.bcast5` directory. This file is used by the mpeg library to view the stream. This avoids the need to scan the file on first open since the TOC builder can run in realtime.

Figure 13.9 shows the red-checked *Realtime TOC* for the *Record Driver* of DVB.

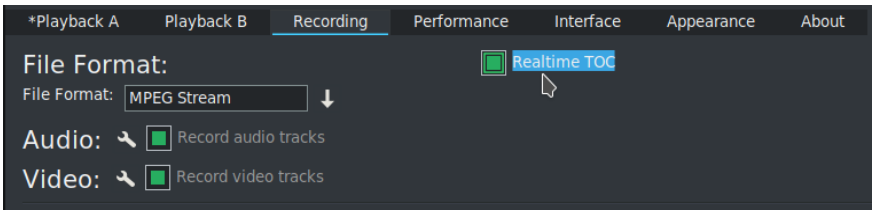


Figure 13.9: *Realtime TOC check red button*

13.2.7 Watching a Previously Recorded TV program

If you have setup for DVB capture as outlined in the section above, just start here to watch a previously recorded TV program. Use the following steps.

- In the main window choose `File` → `Load` files and select the desired media file to watch.
- Broadcast transport streams may consist of multiple programs that all run together on one channel (for example 9−1, 9−2, and 9−3 are all on channel 9). Therefore when you play the recorded program back all of these programs will play simultaneously and that is a jumble. In order to select just the one you want, press Alt-1 in the main window to select *Asset Program 1* (or "Alt-1...8" to select the desired asset program number) which removes all of the unrelated audio/video tracks except for the selected program. It also resets the session to match the media.
- Move timeline to where you would like to start watching.
- Be sure the Compositor window is up and you will see the pre-recorded program there. Use either the keyboard, a remote control that has been set up, or Android device for controls. To toggle between fullscreen or original size, press "f" in the Compositor window.

To watch on a large TV screen or monitor which you have connected, you will have to setup for *Dual Screen Mode* as explained in the next section. In addition, you can either use a Remote Control hardware device or special key setup for manipulating the playback.

13.2.8 Dual Screen Mode

Dual screen mode is useful for displaying CINELERRA-GG video/project on a large screen digital Television set or for watching a previously recorded broadcast TV program via CINELERRA-GG. The physical configuration results in one screen (an *Xwindow*) on one monitor and the other screen on the other monitor or TV screen. Before the mode is usable for CINELERRA-GG, X has to be configured in *dual screen mode* — that is, you must first split the configuration into 2 displays to operate Dual Screen Mode. On some systems, you may be able to get to the dual screen mode by using *Xinerama* allowing for use of 2 physical displays to act as 1 virtual display.

The main menu pulldown, `Settings` → `Preferences` → `Playback A` and `B` Tabs, must be properly configured in order to take advantage of the Dual Screen Mode capability.

In the main menu *Window* pulldown there are *Tile left* and *Tile right* options for the 2 monitors. If you have only 1 monitor, you will see no change upon activating these options. Tiling is possible due to the existence of one long horizontal screen display buffer. Tile left takes CINELERRA-GG and moves all of its windows to the 2^{nd} monitor/TV and Tile right puts it back to the 1^{st} monitor. The left and right monitors are the left part or right part of a Playback A/B Xwindow host referenced in the Video Config Playback A/B for separate X screens. Shortcuts for Tile left and Tile rights are the letters `a` and `b` on the keyboard once you have set *Remote Control mode*. Below is a summary of the configuration letters (only when in Remote Control mode):

a select *playback a* and tile windows left (all on monitor or laptop)

b select *playback b* and tile windows right (all on 2^{nd} monitor or TV)

c select *playback c* and tile windows left, but composer right (TV mode) This option has the effect of taking the Compositor window and moving it to the 2^{nd} monitor/TV without the surrounding borders interfering with the video.

Since Dual Screen Mode is most widely used for either a demonstration or watching TV, you will want to take advantage of Remote Control mode as described in a following section. In the camera shots below, note the square red box that shows up in the upper left hand corner above the Viewer window, when visible, which denotes that the Application/Menu key was used to get into Remote Control mode.

Figure 13.10 shows 2 monitors (could just as well be a big screen TV for the second monitor) with Tile left where all of the CINELERRA-GG windows are on the left most monitor. Note the little red square box in the far upper left corner indicating that Remote Control mode is in effect.

Figure 13.11 shows 2 monitors with Tile right (b key) where all of the CINELERRA-GG windows are on the right monitor. The partial view of the left monitor shows 2 xterm windows with landscape wallpaper background.

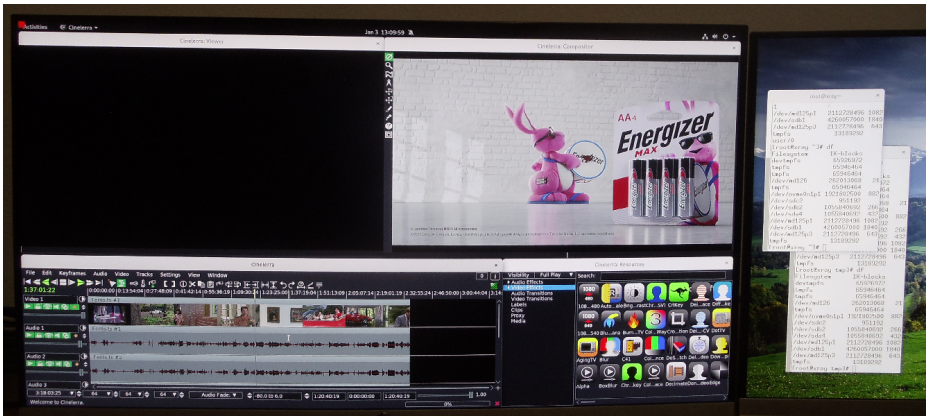


Figure 13.10: *Dual screen - initial view on the left monitor and showing part of the right monitor with 2 terminal windows*

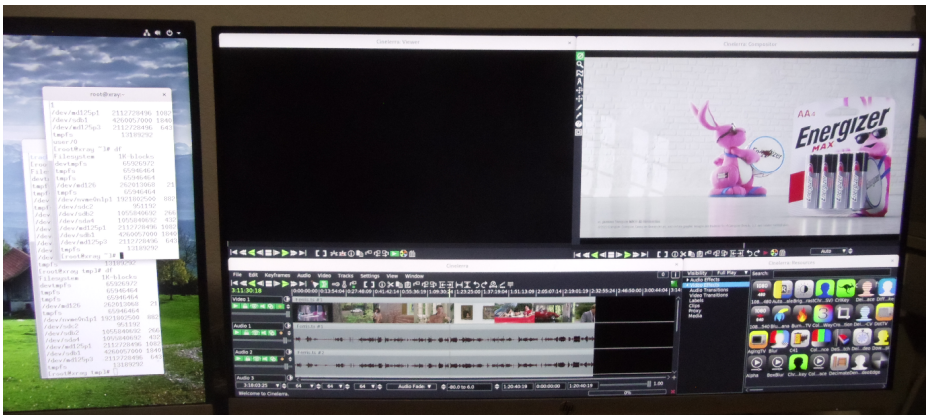


Figure 13.11: *Dual screen -* CINELERRA-GG *and desktop on different screens*

Figure 13.12 shows 2 monitors with Compositor window on the 2nd monitor which was obtained through use of the “c” keypress available when in Remote Control mode.

Figure 13.13 shows 2 monitors with Compositor window on the 2nd monitor in Fullscreen mode so that no CINELERRA-GG borders are visible to distract from the picture.

13.2.9 Remote Control for DVB

CINELERRA-GG DVB recording/playback can now easily be done in *couch potato* mode using a remote control. This comes in handy when you want to playback on a big TV screen for multiple person viewing. You can use the Application/Menu key on the keyboard to toggle between standard CINELERRA-GG usage or a Dispatcher methodology through use of a remote control, an Android programmed device such as a tablet, or keyboard. The Application/Menu key on most keyboards is between the Alt and Ctrl keys on the right hand side bottom and depicts a menu on it either with or without a pointer (see image below). THe Menu key is keycode 135. An ati-x10 Remote Control device (figure 13.14) is

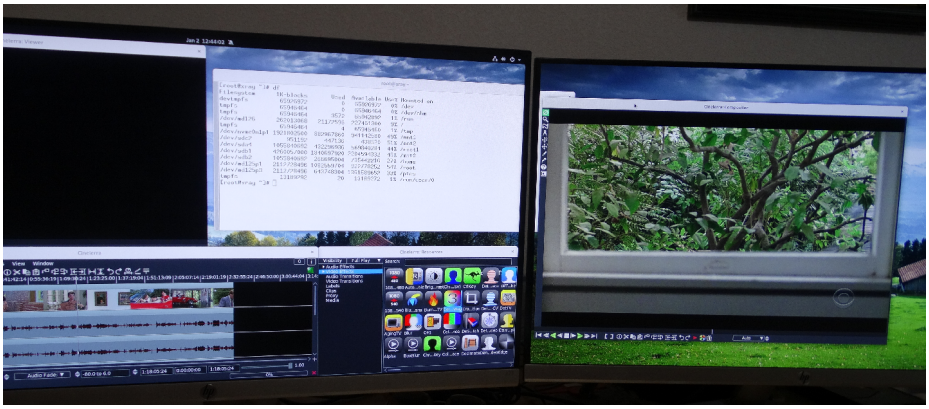


Figure 13.12: *Dual screen - CINELERRA-GG compositor on second monitor*

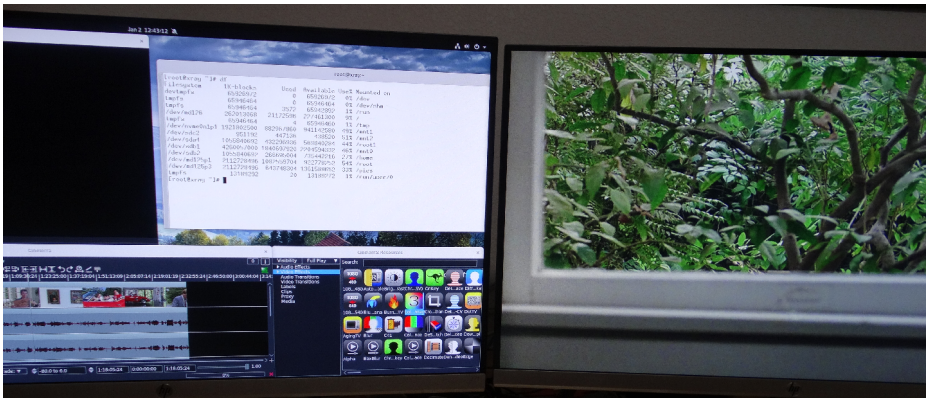


Figure 13.13: *Dual screen - CINELERRA-GG compositor on second monitor in full screen mode*

currently working with CINELERRA-GG and other remote controls may work but have never been tried. The red box on the top left main screen of CINELERRA-GG indicates Dispatcher Method access is activated which allows for keyboard grab which routes all key strokes until toggled off. Below are the currently defined operations.

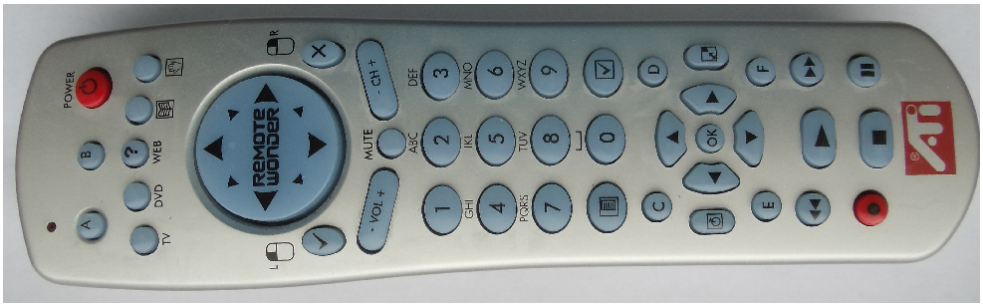


Figure 13.14: *Ati-x10 Remote*

Remote Control Keys (Application/Menu key toggle for ati-x10 remote)

up arrow	=	forward 1 minute
down arrow	=	forward single frame
left arrow	=	forward slow playback
right arrow	=	forward normal playback
d	=	toggle channel scan menu
f	=	toggle full screen
a	=	select playback a and tile windows left (all on laptop)
b	=	select playback b and tile windows right (all on TV)
c	=	select playback c and tile windows left, but composer right (TV mode)
0–9	=	select timeline by 1/10s from beginning to end
a	=	toggle audio mute
m	=	toggle audio meters
keypad /	=	display mpeg ts stream program data
keypad *	=	display mpeg ts stream channel schedule date
keypad +	=	change channel, next
keypad -	=	change channel, prev
keypad .	=	start channel change by numeric entry
keypad ..	=	clear channel entry
book key	=	toggles channel scan (always available)
hand key	=	terminates CINELERRA-GG (always available)

Plus usual Transport keys:

e	=	??
f	=	toggle full screen
«	=	fast reverse
>	=	toggles normal playback
»	=	fast forward
red o	=	slow reverse
square	=	stop
2 lines	=	fast reverse

The Application/Menu key

is used to get into remote control mode, even if there is no remote control hardware device in use, allowing for use of the same defined remote control keys to work on the keyboard. Once the Application/Menu key enables remote mode, the remote control keys don't allow the standard CINELERRA-GG keys to work.

13.2.10 Android Remote Control for DVB

CINELERRA-GG INFINITY has Android remote interface code (figure 13.15). Any device, such as a tablet or a smart phone, can be used as long as it is running the Android operating system. Programming an Android Remote Control for DVB is a bit complicated at first, but becomes pretty simple after a bit.

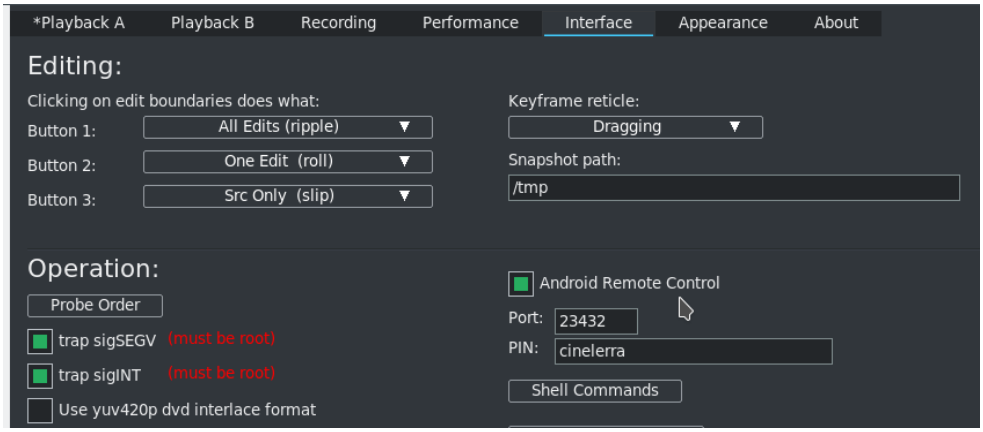


Figure 13.15: *Interface tab shows Android Remote Control*

1. The Android Remote Control requires you to download and install the Android *CineRmt* app package (apk). To install it simply download it (any way you can) to your phone or tablet and click it to install, if it doesn't run automatically.

2. In CINELERRA-GG, open `Settings` → `Preferences` → `Interface` tab, and enable *Android Remote Control* in the Operation section using the checkbox. The default port 23432 and default PIN value *cinelerra* are OK.

3. After you install the new phone/tablet app in step 1, start it up from your apps menu.

Before you can use it, the Android device must have the ip address of your computer entered into the configuration menu. The PC you are running CINELERRA-GG on and the Android device have to be on the same network. You can tell it is in communication by tapping the `Power` (menu) button. When the remote is operated with the *Power* (menu) button, there is a little red box on the upper left corner of the display to indicate the remote state is active. If you do not see the little red box when the menu key is tapped, the communication is broken and must be fixed before proceeding. Follow the directions in the paragraph below.

BIG NOTICE: the firewalls in your computer and wifi router can stop this thing dead. Reconfiguring a firewall is tricky and varies from distro to distro. If possible, during setup it is suggested that you disable the firewall temporarily. For those with advanced skills, use tcpdump or wireshark to look for udp messages from the IP address of the device.

SECTION NOT COMPLETE – (manualandroid)

Figure 13.16 shows an Android Smartphone that can be used as a Remote Control for CINELERRA-GG Recording and Playback.

Figure 13.16: *A Smart Phone with Android Remote Control*

13.2.11 Setup a Remote Control Device

SECTION NOT DONE !! This section explains how you can take your own brand of remote control and set it up to easily work the way you want with CINELERRA-GG.

13.3 The commercial DB

While watching broadcast television, it is possible to use CINELERRA-GG to automatically mute commercials being aired during a broadcast capture. Visual muting as well as sound muting is done! You can NOT use ffmpeg for the format when working with the *commercial DB*. This database that was added is a *novel architecture* shared memory database, a *Traveling Data Base* (TDB). With TDB you have direct access to the shared database presence (instead of having to go to the server).

Basically it works as follows, but there are a few differences in just how it goes about it:

1. Identify a zone in a timeline as a commercial.
2. Characterize it concisely, in the audio and video timeline.
3. Store the characterization in a database.

Then, when capturing media or building a table of contents for an mpeg transport stream (broadcast capture DVB):

1. Characterize each I-Frame (a few frames per sec).
2. Search the database in real-time to identify the image.
3. If a match is identified, automatically press the mute button.
4. Unmute if the pattern match fails, or the commercial ends.

It mostly works, but it takes more time to setup and maintain the database as the advertisements change frequently or vary by a small amount. It would take a small set of personnel to keep the database current and some finessing of the code.

Currently, to activate the commercial database (db) methods you must do the following as root:

```
cd <cin-path>/bin
mkdir /cinelerra #yes, currently only in the / filesystem
./cin_db /cinelerra/media.db #to create a new empty commercial capture database
export BC_USE_COMMERCIALS=1 #in the command shell that is starting \CGG{}
```

This activates the gui tools and database access. As of yet, all of the requisite permission changes necessary to run as non-root have not been worked out (initially just easier to run as root).

When a commercial is *cut*, the following occurs:

1. transmits the capture timeline fragment to `/tmp`
2. scans the snip to characterize it statistically
3. creates a miniature b/w movie of the beginning and end
4. stores all of this into the `media.db`

There are 2 ways currently to identify a commercial:

1. Load mpeg3 transport media, highlight select the commercial in the timeline, and press *cut commercial* on the toolbar.
2. Use the remote control interface to indicate the boundary in/out cut points in the realtime capture timeline, and operate a cut command.

The cut command creates a popup to let you know that it is processing the captured data. This can take several seconds depending on the size and media format. The results from the scan are stored in a new record in the `media.db`.

The stats are somewhat disappointing. Image recognition is way past zero, it is not 100 percent. Broadcast streams have tons of minor variances which hamper digitally perfect reproduction from instance to instance of commercial presentation. This puzzle remains to be solved.

To review media stored in the `media.db`, and delete old expired commercial data records, press (Shift-M) from the main window. This will create a media-db window which shows the various captured samples stored there.

Currently, only a customized *ati-x10* (kernel mod required) radio remote control works. But this can quite readily be modified for other remote control models.

13.4 Transfer VHS/DVD Media or Video8/Hi8 Tapes into CINELERRA-GG

If you want to transfer the data on a VHS tape, DVD disc, or Video8/Hi8 Tapes into CINELERRA-GG, you can do so by playing the media and recording while playing. This captures the media, which you are able to view as it plays, and writes it to a file so it can be played directly or edited. Because there is so much variation in hardware on both the Computer side and the Media Player side, these *how to* directions are only a guideline. You will most likely still have to do some experimentation and testing to see what works for you.

The basic strategy consists of 3 operations: connect the relevant outputs to representative inputs, configure the capture card to match the video, and start the Record monitor. Each of these may have to be experimented with to get them correct for your hardware setup.

Step 1: Connect both audio and video from your player hardware to the sound board and/or the capture card of your computer. Sometimes the capture card handles both audio and video but oftentimes audio and video are hooked up separately.

For example in a test case using the *CX23418* capture card (an older/unusual single chip MPEG-2 encoder with integrated analog video/broadcast audio decoder), one end of an S-Video cable was plugged into the back of a DVD player and the other end plugged into this computer's Capture card; the audio was plugged into the *line in* on the computer sound card.

Step 2: Setup CINELERRA-GG and it might be best the first time if you start with a default `.bcast5` file by using a command line from a window of
`CIN_CONFIG=$HOME/bcast6 cinelerra_path/bin/cin`

- Choose `Settings` → `Preferences` → `Recording` tab in the Preferences menu (figure 13.17).
- Choose a *File Format* – a common suggestion is *ffmpeg* with a *file type* of mp4. In some cases depending on your capture card, the File Format will be set to *MPEG stream* capture. Some formats support MPEG stream capture technique, otherwise the data has to be decoded first and then encoded.

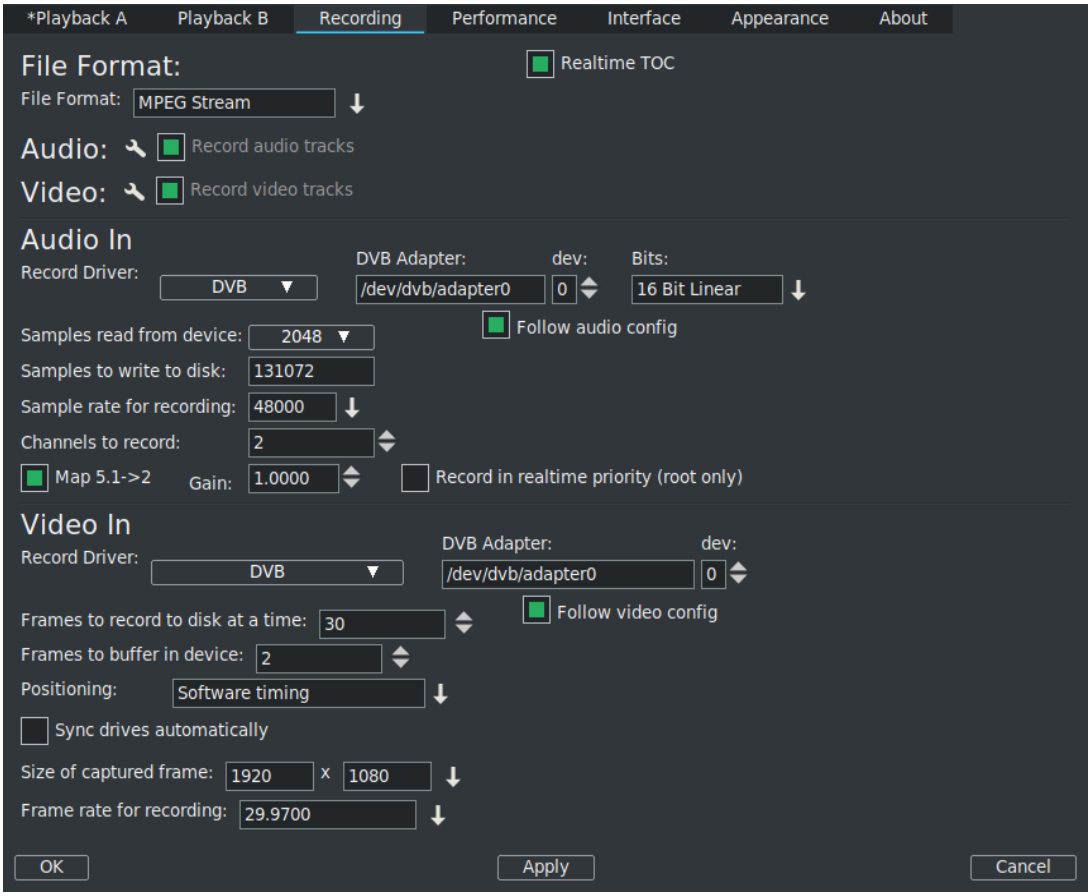


Figure 13.17: *Example of the Preferences menu with the Recording tab highlighted*

- Check that the *Audio In* section has the *Record Driver* set to *Alsa* with *Device* set to *default* as the Capture Source. If you are using something besides Alsa, change that selection accordingly. This may not always be default and you may have to change to something else based on your specific hardware. It has to match up what you see in our Control Panel on your computer. So for example, if your device is HDMI, than the *default* could be *HDMI*...instead.
- In the *Video In* section, change the *Record Driver* to whatever makes sense for your capture card. The test case for the CX23418 would use Video4Linux2 MPEG. A test case for an EasyCap Model #DC60 would use Video4Linux2 JPEG. If unknown what your card uses, you should experiment with Video4Linux2 or Screencapture as good possibilities.
- Also, *Device path* should be verified to be correct in case you have more than 1 device. In the 2 previously mentioned devices, it should proabably be set to /dev/video0.

Step 3: Next use the *File* pulldown to choose *Record* and then the Recording menu will popup (figure 13.18).

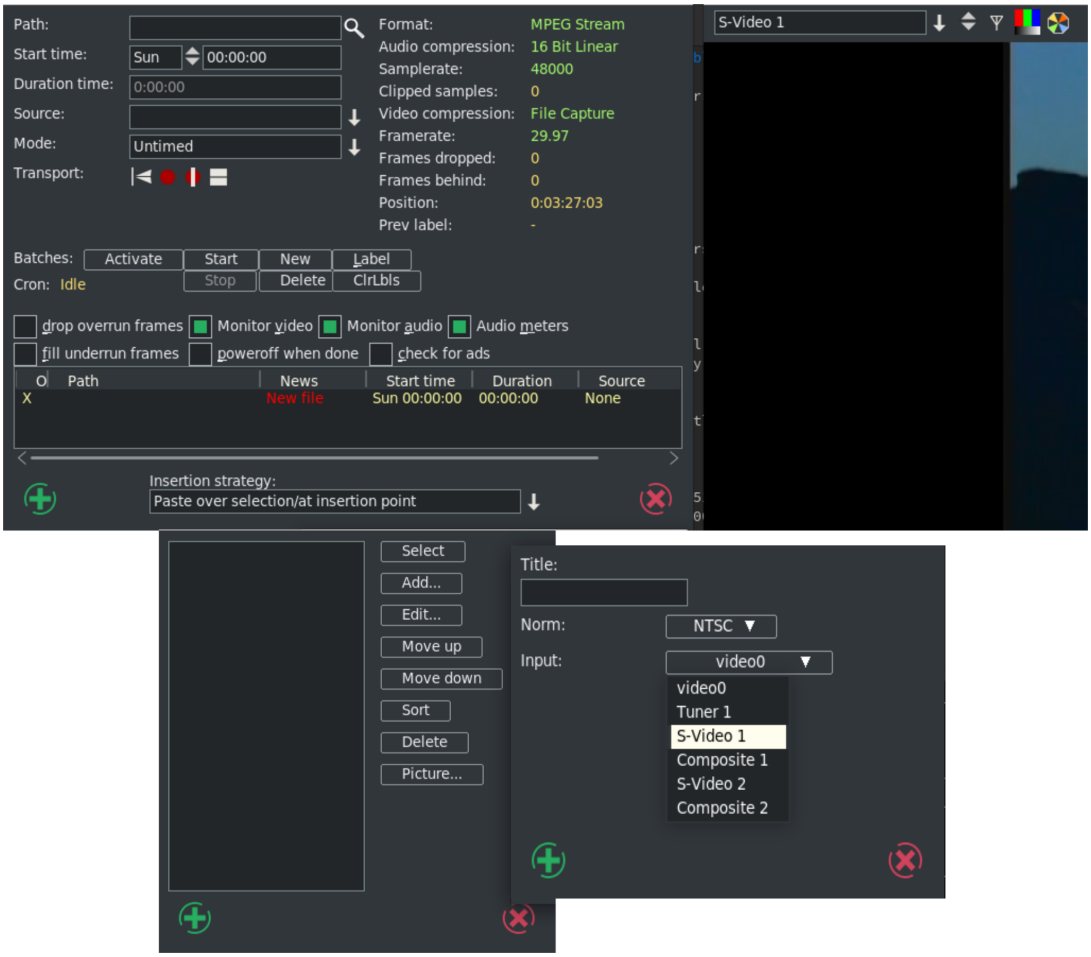


Figure 13.18: *Example of the Recording, Video In, and Channels, along with the Add option, menus.*

- Click on the Antenna icon (3^{rd} icon from the left on the left side of the right hand screen window).
- In the popup *Channels* menu, click on `Add` which brings up the *Edit Channel* menu.
- Use the down arrow in the *Input* box to display the available options. In the test case, and very likely your case too, *S-Video 1* should be picked.
- In the Recording menu, be sure to designate a complete path and filename in the *Path:* textbox which will be used to record/write the playing media. Remember you must have the media on a file in order for CINELERRA-GG to load it on the timeline for editing purposes.
- Then highlight *S-Video* in the *Channels* Menu. Again, the choice in the Channels Menu might not be S-Video – choices could vary widely. Start recording in your Recording menu by clicking on the second icon from the left, next to the *Transport* label (it is usually a round red colored button).

- Next press *play* on your media hardware device. If on the *Video In* side window you see only noise, then S-Video or something else was an incorrect choice and you will have to perform some tests to find correct choices.

There are many more parameters that you may want to vary in the Recording menu or for more details on various items, please refer to the section Record Web Media in real-time. The *Transport* buttons are well defined there also.

Step 4: When the media has finished playing, use the `Stop` icon on the Recording menu – the fourth icon next to the *Transport* label - to stop the recording (figure 13.19).

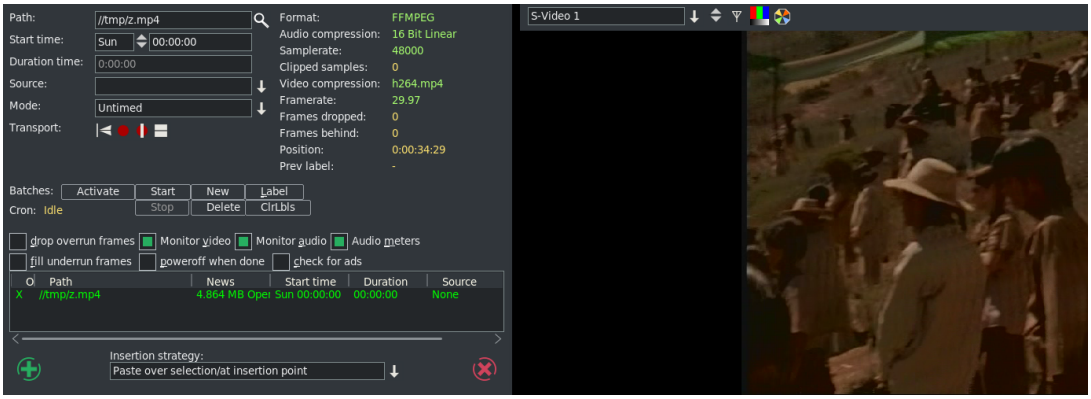


Figure 13.19: *Recording menu and Video In screen while capturing media. Note the Transport stop option.*

Now you can *Load* the file that you played and recorded via the *Insertion strategy* choices on the bottom of the Recording menu and edit the file as usual. Note, that the audio and video may have started at slightly different times so you might have to initially line them up at the beginning using `nudge`. Generally once they are correctly aligned they will stay that way. Nudge is in the expanded patchbay. The nudge value is the amount the track is shifted left or right during playback. The track is not displayed shifted on the timeline, but it is shifted when it is played back.

13.4.1 Use Case #1 – EasyCap Model # DC60

A very specific case using an Easy CAPture USB 2.0 Video Adapter with Audio, Model #DC60 (supports NTSC and PAL) is shown here next. The setup for this device is seen in figure 13.20. A somewhat unusual choice to make note of in this image is the setting for the Video In, Record Driver of Video4Linux2 JPEG.

Output on startup window while playing:

```
DeviceV4L2Base::v4l2_open pixels="MJPG"; res="480x320 640x480 720x480"
DeviceV4L2Base::v4l2_open pixels="YUYV"; res="480x320"
```

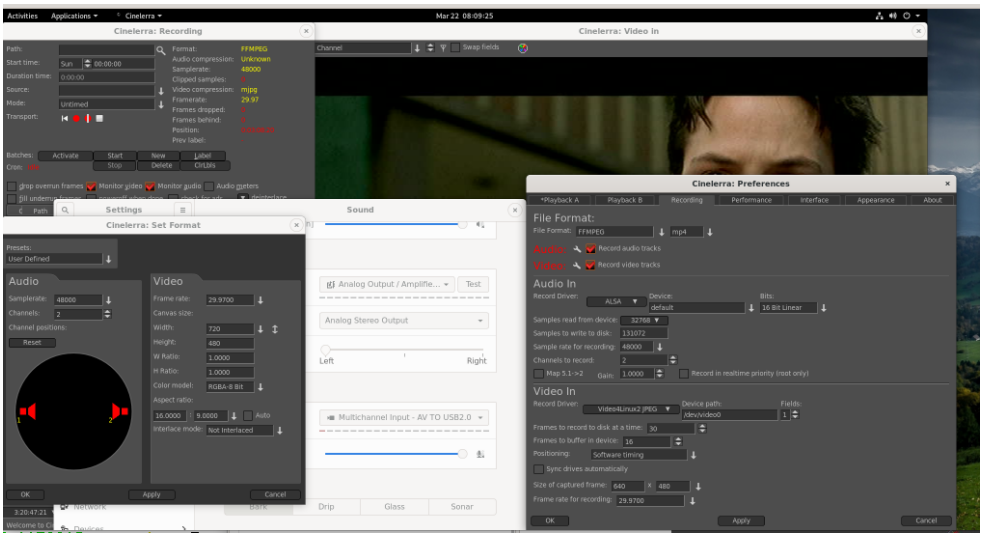


Figure 13.20: *Recording setup for EasyCap device*

```
v4l2 s_fmt 640x480 MJPG
DeviceV4L2Base::v4l2_open VIDIOC_S_STD: Inappropriate ioctl for device
DeviceV4L2Base::v4l2_open VIDIOC_G_JPEGCOMP: Inappropriate ioctl for device
DeviceV4L2Base::v4l2_open VIDIOC_S_JPEGCOMP: Inappropriate ioctl for device
[Detaching after vfork from child process 6539]
```

```
Output from "dmesg" for EasyCap -Model \# DC60:

[ 748.560775] usb 3-1.2: new high-speed USB device number 8 using xhci_hcd
[ 748.574296] usb 3-1.2: New USB device found, idVendor=534d, idProduct=0021,
    ↪ bcdDevice= 1.21
[ 748.574303] usb 3-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 748.574307] usb 3-1.2: Product: AV TO USB2.0
[ 748.574310] usb 3-1.2: Manufacturer: MACROSILICON
[ 748.574313] usb 3-1.2: SerialNumber: 20150130
[ 748.635600] hid-generic 0003:534D:0021.0008: hiddev97,hidraw0: USB HID v1.10
    ↪ Device [MACROSILICON AV TO USB2.0] on usb-0000:16:00.3-1.2/input4
[ 748.698491] mc: Linux media interface: v0.10
[ 748.727360] videodev: Linux video capture interface: v2.00
[ 748.796533] usbcore: registered new interface driver snd-usb-audio
[ 748.796660] uvcvideo: Found UVC 1.00 device AV TO USB2.0 (534d:0021)
[ 748.796876] uvcvideo: UVC non compliance -GET_DEF(PROBE) not supported.
    ↪ Enabling workaround.
[ 748.797204] uvcvideo 3-1.2:1.0: Entity type for entity Processing 2 was not initialized!
[ 748.797212] uvcvideo 3-1.2:1.0: Entity type for entity Camera 1 was not initialized!
[ 748.797353] usbcore: registered new interface driver uvcvideo
[ 748.797356] USB Video Class driver (1.1.1)
```

```
[ 3982.626522] retire_capture_urb: 7 callbacks suppressed
[ 3982.747515] usb 3-1.2: USB disconnect, device number 8
[ 4007.286371] usb 3-1.2: new high-speed USB device number 9 using xhci_hcd
[ 4007.302726] usb 3-1.2: New USB device found, idVendor=534d, idProduct=0021,
    ↪ bcdDevice= 1.21
[ 4007.302732] usb 3-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber
    ↪ =3
[ 4007.302736] usb 3-1.2: Product: AV TO USB2.0
[ 4007.302740] usb 3-1.2: Manufacturer: MACROSILICON
[ 4007.302743] usb 3-1.2: SerialNumber: 20150130
[ 4007.354826] uvcvideo: Found UVC 1.00 device AV TO USB2.0 (534d:0021)
[ 4007.354939] uvcvideo: UVC non compliance -GET_DEF(PROBE) not supported.
    ↪ Enabling workaround.
[ 4007.355252] uvcvideo 3-1.2:1.0: Entity type for entity Processing 2 was not
    ↪ initialized!
[ 4007.355256] uvcvideo 3-1.2:1.0: Entity type for entity Camera 1 was not initialized!
[ 4007.420828] hid-generic 0003:534D:0021.0009: hiddev97,hidraw0: USB HID v1.10
    ↪ Device [MACROSILICON AV TO USB2.0] on usb-0000:16:00.3-1.2/input4
```

13.4.2 Use Case #2 – Hauppauge WinTV-HVR

In the case of the Hauppauge 9500 WinTV-HVR Hybrid TV stick, the setup is similar as in case #1 with the major exception being that the Video In, Record Driver should be Video4Linux2 MPEG.

Figure 13.21: *Two possible capture devices using USB*

Chapter 14

DVD and Bluray Creation

This section describes how to create a blu-ray DVD, or BD referring to a Blu-ray DVD, and a regular DVD, or SD referring to a Standard DVD. The DVDs (plural usage of DVD means either BD or SD) created are unencrypted, unlike commercially available movie DVDs. This CINELERRA-GG version conceivably can create different variations of DVD/Blu-ray media but for the casual user the most standard usages are readily usable and will be described here. In CINELERRA-GG DVD and Bluray creation is a convenience for users who want to edit and then just make basic media without having to work too hard, therefore all features will not be available.

Some preliminary information follows. For NTSC, SD media is almost always 720 × 480 interlaced (the format in the United States, US). For PAL, SD media is almost always 720 × 576 interlaced (Europe, EU, and most of the world). An SD can conceivably be created with a lower resolution – for example 352 × 240 MPEG-1 – but it is not useful. Aspect ratio for either NTSC or PAL can be 4 : 3 or 16 : 9.

Frames per seconds is usually 29.97 for NTSC and 25 fps for PAL. The standard SD dvd generally uses the MPEG-2 program stream with a filename extension of `m2v`. BD blu-ray media is not normally interlaced, but you can leave it interlaced. Blu-ray uses the MPEG transport stream which contains 1 or more program streams with a default filename extension of `m2ts`.

Requirements for creating DVDs is the hardware device to write the media on and, obviously, the blank media for either BD or SD. When generating SD media, you will have to install *dvdauthor* and for BD media, install *udftools* if they are not on your system. Also, keep in mind that to mount filesystems for creating files and to burn DVDs, you will have to be root since you have to have privileges, unless special permissions have been provided for a non-root user. It is also highly recommended to run the CINELERRA-GG startup from a terminal window, instead of the icon, in order to see informative messages of how to actually write the output (at prompt: keyin `<install_directory_path/bin/cin` or `<install_directory_path/<your_version>.AppImage`).

A warning here – writing blu-ray BDs and regular SDs can take a large amount of clock time. Keep in mind that a blu-ray can contain 25 hours of viewing, so would take multiple hours to just write one.

The amount of data you have to render determines the amount of disk space you will need for working. The max disk space needed for a completely full media disc is a little over 100*GB* for 50*GB* Double Layer (DL) media or 50*GB* for a single layer BD blu-ray. A standard SD of 4.7*GB* needs about 10*GB* disk space. You will need much less if the render for blu-ray is less than 25 hours of media. Generally you will need twice as much disk space as will be written on the media to ensure you have sufficient space for working and copying.

The most important thing you need to know about in order to get started is *Format* and *Scale* in the *Create dvd or bd* window. Format settings shown in the `Set Format` window are set accordingly to an algorithm. Basically, it will take whatever you say in the asset format. It matches that against the known Presets available so that is what will be shown. If the asset format doesn't match any of the Presets default formats, then that will be shown as the *User Defined* format. In addition when you load media, the format is initially set to *same as source* so matches the source input and if that matches a known preset, then that is what is shown. The PAL versus NTSC only comes into play when there is no known correct matching format when you attempt to create a DVD Render batch job. It is applied when you click OK. For example, if you load up a YouTube video, it will not match any known format and will choose PAL or NTSC based on time zone.

Figure 14.1 shows pulldown *Presets* in *Set Format* window:

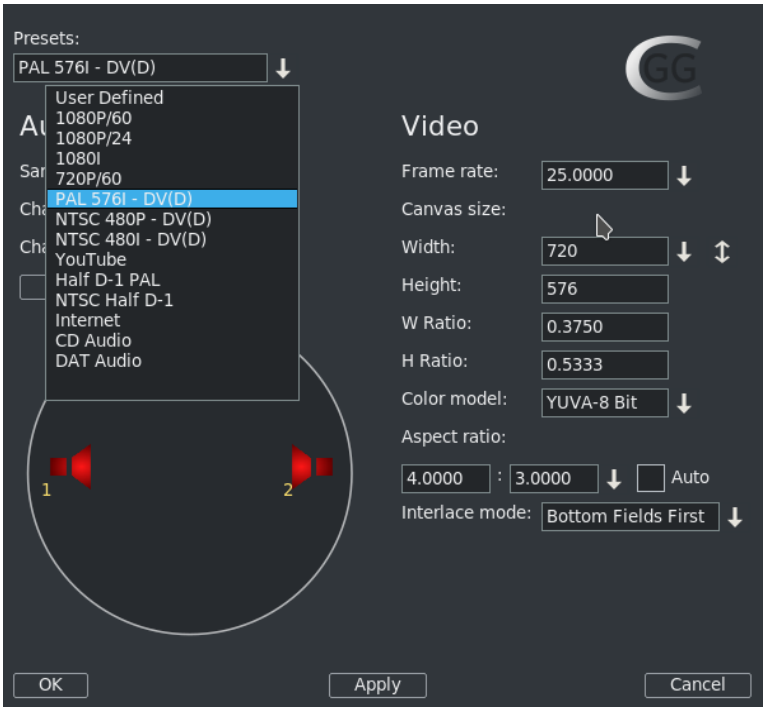


Figure 14.1: *When choose PAL, values get changed in window to reflect PAL*

A quick set of basic steps to create DVDs is immediately below and usually just using the defaults will get you something. However there is a serious issue with interaction between

the Operating System and bdwrite when creating a BD/blu-ray that requires automount to be turned off. For more details on using automount see: 14.8.

1. If not logged in as root, you will get an error message in order to avoid doing a lot of work and then failing out because root is required for automount and to write on DVD hardware.
2. Load your input source media via: `File` → `Load files`.
3. Choose PAL or NTSC for SD/dvd or 1080P/24 for blu-ray in `Settings` → `Format`.
4. If DVD "Interlace Mode" of Bottom Field First is wanted, you MUST set that in `Settings` → `Format`. If you do, then also remember to change it when you no longer want BFF.
5. For blu-ray, choose BD Render or for PAL/NTSC, choose DVD *Render* in *File* menu.
6. Designate a *work path* with sufficient disk space and then Chk-OK.
7. When the Batch Render window comes up, click on *Start* and the batch jobs will run.
8. Read the final messages echoed to the screen to see the command for burning... OR:
9. Use the provided directory name to:
   `cd /<target directory>bd_(or dvd_)<date-time>`
10. Load your media, format if needed, note device name to substitute for *<bd>* or *<dvd>*
    - If rewritable blu-ray, use
      `dd if=./bd.udfs of=/dev/<bd> bs=2048000`
      Note: unwritten media must be quickly formatted one time using:
      `dvd+rw-format /dev/bd`
    - If write-once blu-ray, use
      `growisofs -dvd-compat -Z /dev/<bd>=./bd.udfs`
    - If any DVD media, use
      `growisofs -dvd-compat -Z /dev/<dvd> -dvd-video ./iso`

Any problems encountered will require that you read more information in this section to include specific details, helpful hints, and problem resolution.

Details and specific explanations to create blu-ray or regular DVD are provided here. It is very advantageous to startup CINELERRA-GG from the command line prompt instead of the icon. Also, please be root or your hard work will be lost when the automount is issued and fails for bluray udfs mounting.

The general design of the DVD/blu-ray generation operations is to first render media using batch rendering and then terminate CINELERRA-GG to start a script which creates the target device filesystem data. These scripts are the `dvd.sh` and `bd.sh` scripts written into the target directory. For DVD, the general plan is to write a directory `<target>/iso` with the dvd filesystem via *dvdauthor* and then generate an iso9660 filesystem and write it to a dvd via *growisofs*.

For blu-ray, the filesystem generation is slightly harder. First, it creates an empty filesystem image `<target>/bd.udfs` using *mkudffs* which makes a big hole for the filesystem data. The hole is made just a little bigger than the data written by *bdwrite* so that you don't have to write an entire 25*GB* or 50*GB* disc even if no data exists. The actual calculation for the mkudffs size is yourfile.m2ts size-in-bytes/2048 + 4096. This empty filesystem is loopback mounted to make it writable, and the linux kernel manages the filesystem image. The bdwrite program applies the blu-ray structure to the UDF filesystem by creating the needed BDMV blu-ray filesystem, which the kernel stores onto the image file `bd.udfs`. When udfs is unmounted, the kernel finalizes the disk image on bd.udfs. The bd.udfs image can be written directly to a blu-ray disk via *dd* or *growisofs*.

NOTE of IMPORTANCE: there is a serious situation with the interaction between the Operating System (OS) and bdwrite when creating blu-ray, that requires automount to be turned off. The blu-ray automatic script unmounts the blu-ray/UDF filesystem but the system has not finalized the directories so the OS creates a new loop file device and the data is loaded and cached for use by the new loop but it is stale. Consequences is that not all of the data is written where it should be. The solution is for the OS not to mount this second mount so we have to make sure it doesn't. There are 2 methods to fix this. The first and easiest is by using the following command to disable automount:

```
gsettings set org.gnome.desktop.media-handling automount false
```

This can be reversed when you have completed the blu-ray generation via:

```
gsettings set org.gnome.desktop.media-handling automount true
```

A different and more complicated method you can use to turn off automount is to download and install the *dconf-editor*. Automount is a system parameter and only needs to be done once unless you do not want automounts to always be disabled.

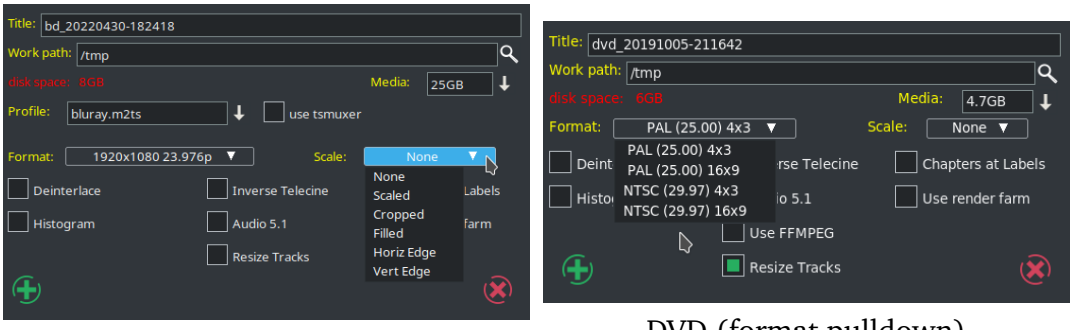
1. run: *dconf-editor*
2. select: `org` → `gnome` → `desktop` → `applications` → `media-handling`
3. uncheck: *automount*
4. close *dconf-editor* window

Immediately below are the detailed steps with explanations for creating SD or BD media.

Step 1: Construct a session with the desired presentation:

- Format frame rates choices are 29.97 or 25 for SD, based on the user's timezone, with NTSC 29.97/US or PAL 25 /EU. For BD, the media input will be analyzed to automatically pick the default format or if unknown, the user's timezone will be used to default to 1920/29.97*i* for US or 1920/25*i* for EU. Be sure to set the rendering parameters in the `settings` → `format` menu.
- Choose audio stereo or 5.1, again depending on your media.
- Target Geometry will be 720 × 480 (US) or 720 × 576 (EU) for SD.

Step 2: From the main window, select `file` → `BD Render` or select `file` → `DVD Render` (figure 14.2). Then:



BluRay (scale pulldown)

DVD (format pulldown)

Figure 14.2: *BD Render and DVD Render*

(Note: both the BD and the DVD windows above, show insufficient space, *disk space:* in red)

- Select desired features, check/un-check as appropriate.
- Click OK check button. It is very important to realize that when you check OK, the EDL will be saved and that will be used for batch job rendering. If you bring

up the Batch Render and then change some parameters, they will not take effect UNLESS you remember to either check *Save to EDL Path* or *Use Current EDL* in the Batch Render window. You will get a reminder automatically if *warn if jobs/session mismatched* is checked.

Explanation of the choice boxes as seen in figure 14.2 for both SD and BD menus is given below. Many of them are plugins which allow you to further manipulate the settings for best results. They are just suggestions set by the program automatically based on your input media, and can be reset to suit your needs. These are listed in the next 4 points.

1. If the media does not match the DVD target geometry, and the scale plugin is not already in use, then the *scale* plugin is applied with scaling set to fit the media dimensions to the DVD format target geometry.
2. If the video height is at least twice the DVD height and the input media is interlaced, then the *deinterlace* plugin is applied with odd line sampling.
3. *Audio 5.1* will automatically be set to the wide-audio if you have 6 tracks of audio.
4. To allow video data to be accessible and overlay properly, the track buffers are *resized* to the largest track frame size in use (*Resize Tracks*). The theory behind this is to make sure to have enough memory to cover the entire presentation for transcoding.

All of the current choice boxes are further defined immediately following.

Deinterlace remove the interlace. Interlacing is a video scanning system in which alternating lines are transmitted so that half a picture is displayed each time the scanning beam moves down the screen. You lose a lot and the quality is bad when you view interlacing on a progressive TV. You might not really want to use deinterlace, because if you deinterlace non-interlaced media, it will look awful. In the case of DVD only, for interlaced media when you do not check the Deinterlace option, it will always be done as Top Field First. To get DVD media to be Bottom Field First, you MUST set the "Interlace Mode" in `Settings → Format`.

Scale alter the spatial mapping of an image to increase or reduce the size; modifies the picture. When some programs scale from 4 : 3 to 16 : 9 they will automatically cut off the appropriate section of the image for you. It is necessary to keep in mind, that *square pixels* is the true end goal of scaling, not the aspect ratio which could result in squished or stretched output. More information about scaling will be provided on a subsequent page with usage of the *Scale Ratio* plugin.

Profile for bluray only, use the default bluray.m2ts or choose bluray_lpcm.m2ts for linear pulse code modulation for digitally encoding uncompressed audio information.

use tsmuxer for bluray only, to use tsMuxer as a transport stream muxer for remuxing/-muxing elementary streams, check this box. Usage of tsmuxer requires that your computer have the tsmuxer software installed along with any of its prerequisites. The default is do not use tsmuxer.

Histogram remaps the color space. The color space ranges from $0-255$ for 8-bit color values. You can use this tool to remap the color space to use the entire space or for stretching the contrast. Also, it lets you perform global color-correction on the image. You can use this to correct for color screens that are *too blue*, or for color Televisions that produce *brownish* output, or whatever. In addition to color-correction, you can use the RGB modification tool to add color to images that didn't have color to begin with. For instance, you can *pseudo-color* greyscale media.

Inverse Telecine the reverse of $3:2$ pulldown where frames, which were duplicated to create 60-fields/second video from 24-frames/second film, are removed. MPEG-2 video encoders usually apply an inverse Telecine process to convert 60-fields/second video into 24-frames/second encoded video. The encoder adds information enabling the decoder to recreate the 60-fields/second display rate. *Telecine*, i.e. $3:2$ pulldown, is used to transfer film to video. That's where the $3:2$ ratio comes in. To ensure that there will consistently be 60 frames per second, the first frame is displayed on the TV screen 3 times and the second frame is displayed 2 times. The following frame is repeated 3 times, the next one 2 times, etc. throughout the film. For inverse telecine, you show 2 of the film frames for 3 of output frames. Only check *Inverse Telecine* if you have film or something that is $24fps$ and want to project to $30fps$ (most likely this will never be necessary).

Audio 5.1 6 channel surround sound. For most home systems, uses five full bandwidth channels and one low-frequency effects channel. Could automatically get set as explained previously.

Aspect Ratio aspect ratio may be automatically set to $4:3$ or $16:9$. Aspect ratio would better be defined as the size of the display, monitor, or TV which will be used to view the output. If you measure your old TV, which supposedly is $4:3$ and your latest digital TV, which is supposedly $16:9$, you will see that those ratios aren't always correct anyway. Then measure your laptop monitor, your desktop monitor, and your neighbor's, and lo and behold, the ratios don't fit either of the purported *standard* aspect ratio. Maintaining square pixels via scaling is more important in the long run.

Use FFMPEG this is user's choice; it is recommended and faster but more difficult to modify due to numerous options. For blu-ray, ffmpeg must be used and is not an available option.

Resize Tracks change track width and height as explained previously. The size is adjusted to the largest frame size needed.

Chapters at Labels for DVDs without this checked, chapter markers are automatically inserted every 5 minutes. The chapter labels can then be *skipped to* when playing the DVD. If instead, you want to put labels in at opportune times, you will have

to run dvdauthor outside of CINELERRA-GG and mark the chapter labels yourself by hand. In that case, you should check the box *Chapters at Labels* so that the automatic 5-minute labels are not created. This checkbox is not currently available for blu-ray, but you can easily create chapters on blu-ray by adding '-c timeinterval in ticks' as the first argument to bdwrite in the bd.sh job generated by "BD Render" and then executing that job manually. On 1 test computer 100 ticks equalled 20 seconds. When "use tsmuxer" option is included, there is by default a chapter every 5 minutes. This can be customized by modifying the bd.meta file associated with your current job, changing the '–auto-chapters=5' to your choice in minutes instead of 5, and then rerunning the pertinent lines seen in your window.

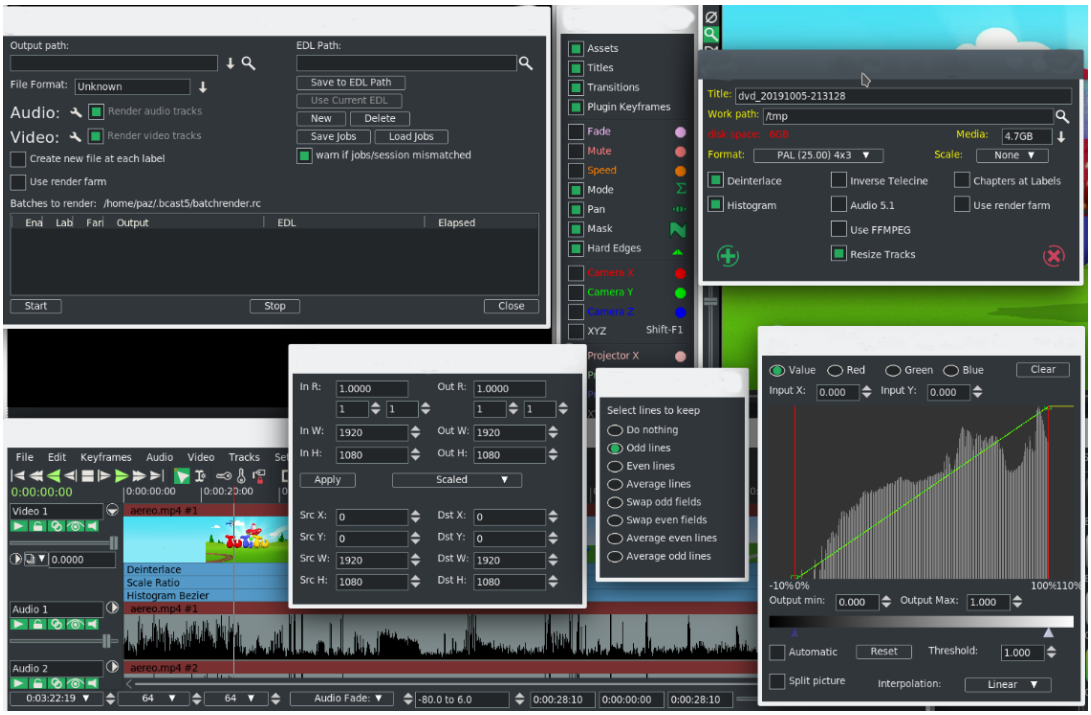


Figure 14.3: *check Deinterlace; Histogram and Resize Tracks*

Figure 14.3 shows on the upper right side the *Create DVD* window with Deinterlace, Histogram and Resize Tracks checked. Also Scale is set to Scaled. Once the green checkmark is clicked, the *Batch Render* window comes up and in the main window you will see the 3 plugins below the video track on the main window. By clicking on the magnifying glass that appears on the rightmost side, the controls for each will popup and you can make any necessary adjustments. Note the numerous choices for Deinterlace; the Value, Red, Green, and Blue for color adjustments in the Histogram window; and Scale Ratio popup menu for numerical settings control.

The `Scale` parameter gives you a lot of flexibility. A default based on your input media is provided for you but possible choices are None, Scaled, Cropped, Filled, Horiz Edge, and Vert Edge. You will have the opportunity to manipulate the desired results in the *Scale*

Ratio window. Values for W (width), H (height), and X/Y coordinates are the number of pixels. For example, if video is 720 × 432, that is obviously 720 pixels by 432 pixels and this would be the values for Dst W and for Dst H. So if you have some media that is off center you can crop by changing the SRC Y value AND then change DST X/Y to non-zero. It will become the output origin. To see what it does, change them from 0.0 to 400.0 and you will see big changes in the compositor window.

For example, if you have the Cropped choice for the Scale, you will want to manipulate the *ScaleRatio* plugin (the magnifying glass on the main window video track) which brings up the Scale window. For cropped top instead of crop both top and bottom, modify the *Src Y*. As you change the Y scale, you will see the cropping take place in the Compositor.

Scaling options are provided in order to preserve image aspect ratio. To determine which scaling option to use, it is important to correctly identify your source/destination video aspect ratios. Next is a short explanation of possible options.

None do not do any scaling. The destination output matches the source input. There is no resizing.

Scaled do not use uniform scaling on X and Y. Just make it fit and you will might end up with squishing/stretching in one direction or the other in order to do that. This happens when the input aspect ratio is different than that displayed on the output.

Cropped removes outer edges of a source image in order to fit the image on the output display. This is done in order to maintain uniform geometry scale on the destination display without being affected by the particular factor of original media aspect ratio. Since cropping omits image area, and the areas which you wish to view may not be viewed when the image is centered, you can pan the image source using the Src X/Y to modify which area is in view. You can also think of this as *scaled up*.

Filled the entire output display will be filled with image content but in order to do so, some of the image may have had to be cut. Any mismatch between the two will be filled with black. Centering of the image can be modified by using the Dest X/Y variables in the Scale Ratio controls. You can also think of this as *scaled down*.

Horiz Edge this indicates preserving of the Horizontal edge.

Vert Edge this option preserves the Vertical edge.

Horizontal and Vertical are duplicates or restatement of same functionality as Cropped or Filled but are provided as options to accommodate different ways of thinking. In any case, you can choose which outer edges of the image to crop by using the *Show Controls* of the Scale Ratio plugin. For example, you can ensure that no *action* is lost by displaying the center of the screen only or making sure that any textual information on the bottom is not lost by cutting only the top off.

Step 3: Batch render menu appears with `m2ts` format selected for blu-ray or `dvd` format selected for regular/standard DVD when File Format selected is ffmpeg in the previous *Create DVD* menu. It will work just fine without selecting ffmpeg for DVD and may be advantageous not to. Using the audio/video wrench tools (you will have to have the video batch job highlighted to manipulate the video or audio batch job for audio).

- Setup the audio bitrate (192000 recommended). Data rate is $192K$ default.
- Setup the video bitrate ($6000000 - 12000000$ recommended). $10Mb/sec$ is the current default.
- Click OK check button.

The default bitrate is the largest value possible. The actual *target* bitrate is calculated based on a formula from the blu-ray/DVD code. It divides the media size (in bits) by the video time (in seconds) to find the bitrate that will *just fit* on the target media. This could create a weird bitrate if the media is large, and the video time is small, so the default/target bitrate is limited to $10Mb/s$.Batch jobs are then built and appended to the job list. Once these batch jobs are built, if you make any changes, you have to start over. You will see listed the batch jobs that are created to perform the rendering/tasks – 2 jobs for blu-ray, one for audio/video processing and one for the scripts. There will be 3 batch jobs created for DVD when not using ffmpeg to include one each for audio and video and then one more for the scripts. When you click on `start`, it fires off those jobs and you will see the rendering main window progress bar in the bottom corner reflecting the fact that it is currently processing. Be aware that the CINELERRA-GG program will be totally shutdown AFTER the batch jobs finish.

Figure 14.4 for DVD and Figure 14.5 for BD show creation renders with batch jobs queued up and ready to be started. Note that because it is not ffmpeg, the processing of video is done separately from audio. If you need to modify the video tracks, which you can see is ghosted out, you need to highlight the first batch job listed. Be sure to highlight the first batch job before pressing Start so it runs all of the jobs and be sure the X for job enabled is set.

When you click on *start*, it fires off those jobs and you will see the rendering main window progress bar in the bottom corner reflecting the fact that it is currently processing. Be aware that the CINELERRA-GG program will be **totally shutdown after the batch jobs finish** and you will be at the command line prompt.

This will produce a new directory in your target path which contains a filesystem image file. For example:
`/<target directory>/bd_(or dvd_)<date-time>/`
Directory and file names should not be changed at this time because the scripts and programs rely on the given names in order to proceed. You can change them later for your own purposes.

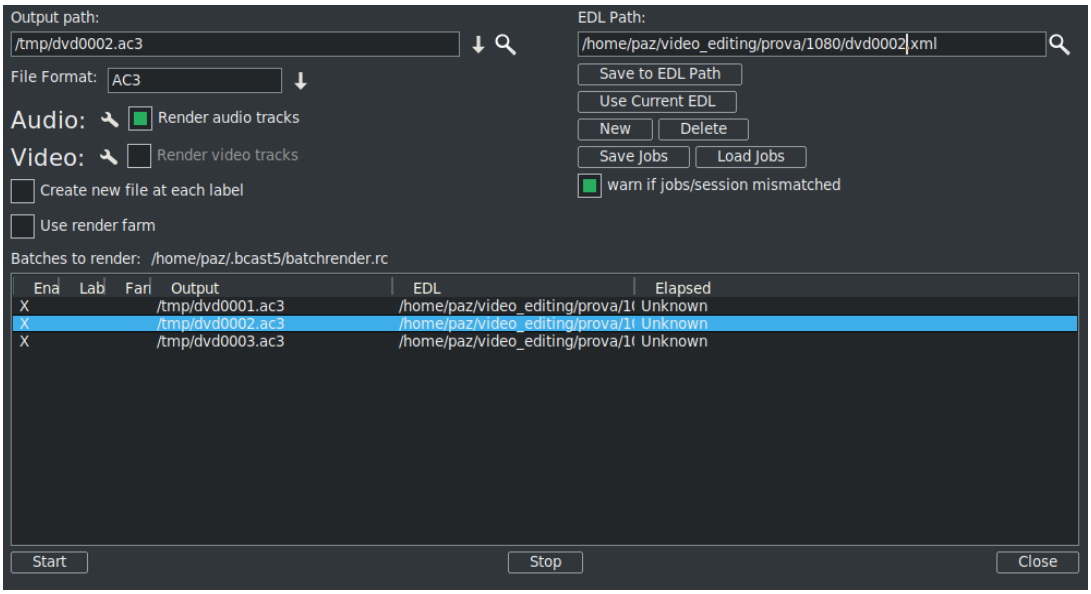


Figure 14.4: *Batch render for DVD creation*

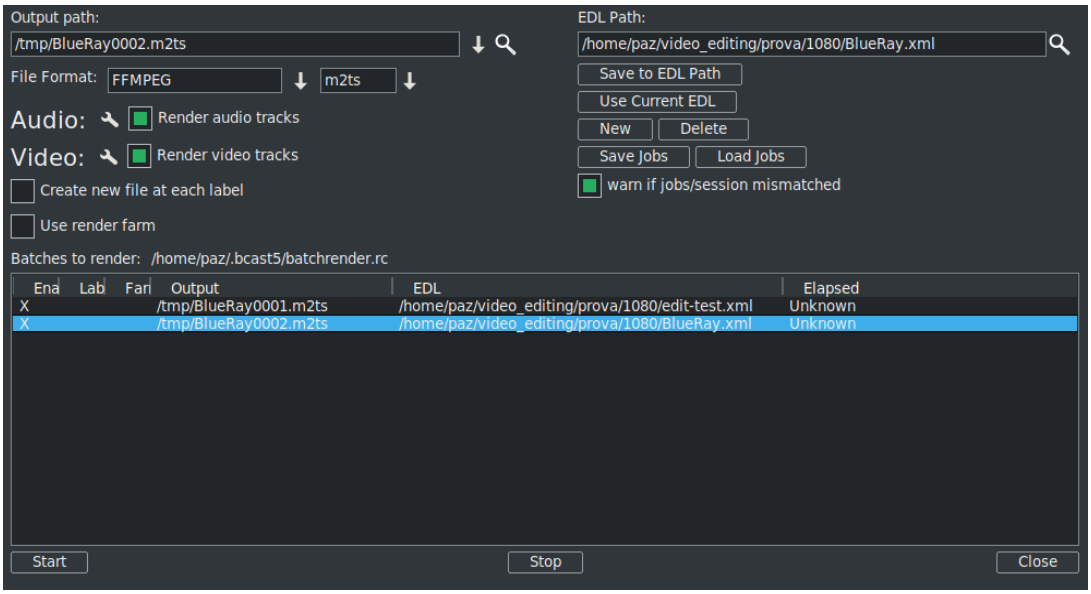


Figure 14.5: *Batch render for BD creation*

If bluray to test the filesystem you just created, use the command line interface; loopback mount the filesystem image which was generated in the target directory. For example if blu-ray:

```
cd /<target directory>/bd_(or dvd_)<date-time>/
mount -o loop,ro ./bd.udfs ./udfs
#check the media using a compatible media rendering tool like ffplay
umount ./udfs...
```

To burn blu-ray media you will need to run from the command line interface. In the examples below, `/dev/bd` represents your blu-ray writer device (for example: `/dev/sr1`) and `/dev/dvd` represents your dvd writer device (for example: `/dev/sr0`).

Blu-ray Media

For rewritable blu-ray (recommended) (BD-RE):

Note: new or previously unwritten media must be formatted first using:

```
dvd+rw-format /dev/bd #only done once and does not take very long
```

To write or rewrite rewritable blu-ray media:

```
cd /<target directory>/bd_(or dvd_)<date-time>/
dd if=./bd.udfs of=/dev/bd bs=2048000 #the growisofs command below also works
```

To write blu-ray write-once media: (BD-R) (no pre-formatting needed):

```
cd /<target directory>/bd_(or dvd_)<date-time>/
growisofs -dvd-compat -Z /dev/bd=./bd.udfs
```

DVD Media

For rewritable DVD (DVD+RW):

Note: unwritten media must be formatted first using:

```
dvd+rw-format /dev/dvd #only done once and does not take very long
```

To write a DVD, load blank media and run the following from the command line (requires dvdauthor):

```
cd /<target directory>/bd_(or dvd_)<date-time>/
growisofs -dvd-compat -Z /dev/dvd -dvd-video ./iso
```

Figure 14.6 shows the availability of 4:2 :2 for a Batch Render seen by clicking on wrench icon.

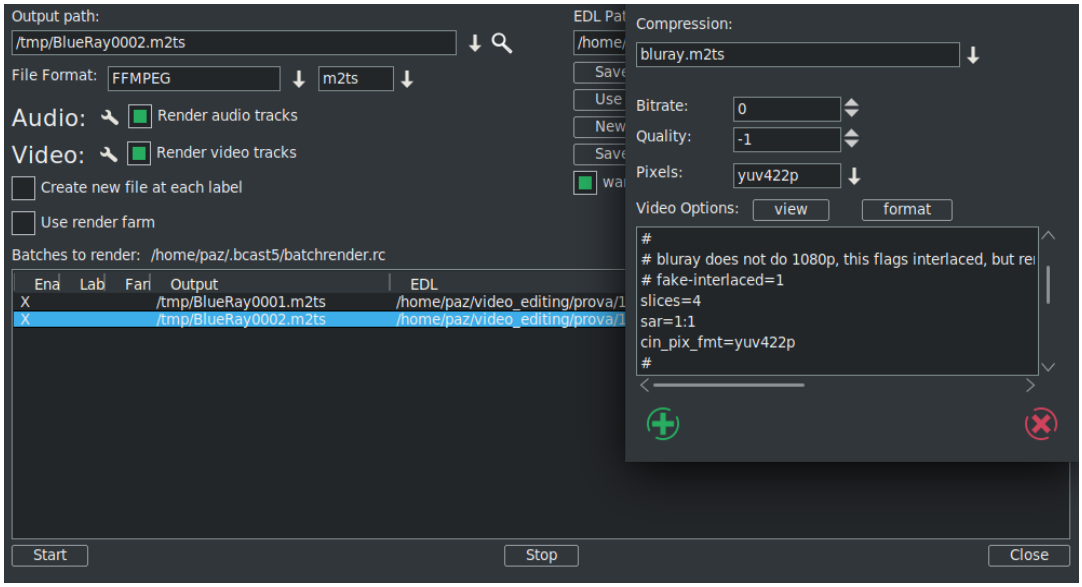


Figure 14.6: *Video options for bluray yuv422p*

Figure 14.7 shows the availability of 10-bit high quality 4:2 :2 for a Batch Render seen by clicking on wrench icon. You need specially compiled CINELERRA-GG in order to use the x265 10-bit as opposed to 8-bit.

14.1 Output Terminal Messages from Creating DVDs

Below are examples of what the batch jobs generate and you will see on the terminal screen if you started the CINELERRA-GG program in the recommended manner from a terminal window. It is just informational but will let you know if errors. In looking at any of the output, you can safely ignore the errors that read *Unsupported codec with id 100357 for input stream 0* – this comes from *nav-data* (navigation data). The first 2 examples are seen from running the batch jobs; the last 2 are from the single line execution which records the media output to the DVD hardware.

SD Example: Partial Output during CINELERRA-GG run

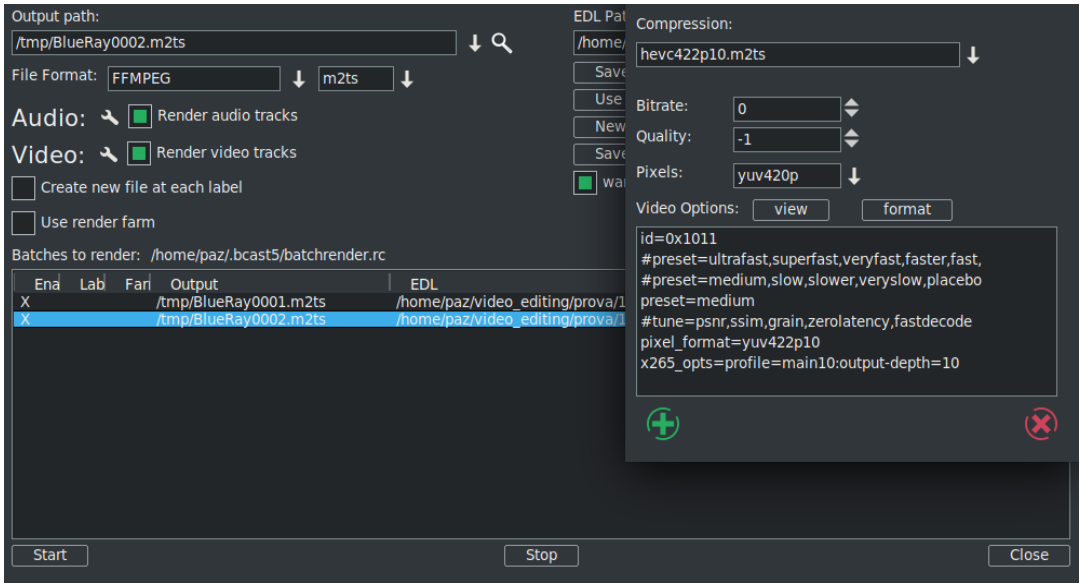


Figure 14.7: *Video options for bluray yuv422p10*

```
...
FFMPEG::open_decoder: some stream times estimated
Render::render_single: Session finished.
FFMPEG::open_decoder: some stream times estimated
Render::render_single: Session finished.
running /tmp/dvd_20160407-113530/dvd.sh 1 /tmp/dvd_20160407-113530
INFO: [mplex] mplex version 2.1.0 (2.2.7 $Date: 2012/11/17 01:55:16 $)
INFO: [mplex] File /tmp/dvd_20160407-113530/dvd.m2v looks like an MPEG Video
    ↪ stream.
. . .
INFO: [mplex] MUX STATUS: no under-runs detected.
DVDAuthor::dvdauthor, version 0.7.1.
Build options: gnugetopt graphicsmagick iconv freetype fribidi fontconfig
Send bug reports to <dvdauthor-users@lists.sourceforge.net>

INFO: default video format is NTSC
INFO: dvdauthor creating VTS
STAT: Picking VTS 01

STAT: Processing /tmp/dvd_20160407-113530/dvd.mpg...
STAT: VOBU 32 at 15MB, 1 PGCs
INFO: Video pts = 0.133 .. 22.789
INFO: Audio[0] pts = 0.133 .. 22.789
STAT: VOBU 46 at 21MB, 1 PGCs
CHAPTERS: VTS[1/1] 0.000
INFO: Generating VTS with the following video attributes:
```

```
INFO: MPEG version: mpeg2
INFO: TV standard: ntsc
INFO: Aspect ratio: 16:9
INFO: Resolution: 720x480
INFO: Audio ch 0 format: ac3/6ch, 48khz drc, 'en'

STAT: fixed 46 VOBUs
INFO: dvdauthor creating table of contents
INFO: Scanning /tmp/dvd_20160407-113530/iso/VIDEO_TS/VTS_01_0.IFO
To burn dvd, load blank media and run:
growisofs -dvd-compat -Z /dev/dvd -dvd-video /tmp/dvd_20160407-113530/iso
```

BD Example: Partial Output during CINELERRA-GG run

```
...
FFMPEG::open_decoder: some stream times estimated
Render::render_single: Session finished.
+ PATH=/usr/lib64/ccache:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/
    ↪ bin:/mnt0/build5/cinelerra-5.1/bin
+ mkdir -p /tmp/bd_20161224-162059/udfs
++ du -sb /tmp/bd_20161224-162059/bd.m2ts
++ sed -e 's/[ ].*//'
+ sz=19811904
+ blks=13769
+ mkudffs /tmp/bd_20161224-162059/bd.udfs 13769
start=0, blocks=16, type=RESERVED
start=16, blocks=3, type=VRS
start=19, blocks=237, type=USPACE
start=256, blocks=1, type=ANCHOR
start=257, blocks=16, type=PVDS
start=273, blocks=1, type=LVID
start=274, blocks=13238, type=PSPACE
start=13512, blocks=1, type=ANCHOR
start=13513, blocks=239, type=USPACE
start=13752, blocks=16, type=RVDS
start=13768, blocks=1, type=ANCHOR
+ mount -o loop /tmp/bd_20161224-162059/bd.udfs /tmp/bd_20161224-162059/udfs
+ bdwrite /tmp/bd_20161224-162059/udfs /tmp/bd_20161224-162059/bd.m2ts
+ umount /tmp/bd_20161224-162059/udfs
+ echo To burn bluray, load writable media and run:
To burn bluray, load writable media and run:
+ echo for WORM: growisofs -dvd-compat -Z /dev/bd=/tmp/bd_20161224-162059/bd.
    ↪ udfs
for WORM: growisofs -dvd-compat -Z /dev/bd=/tmp/bd_20161224-162059/bd.udfs
+ echo for RW: dd if=/tmp/bd_20161224-162059/bd.udfs of=/dev/bd bs=2048000
```

```
for RW: dd if=/tmp/bd_20161224-162059/bd.udfs of=/dev/bd bs=2048000
```

SD Example – Partial Output during writing disc media

```
growisofs -dvd-compat -Z /dev/sr0 -dvd-video /tmp/dvd_20161224-160756/iso
WARNING: /dev/sr0 already carries isofs!
About to execute 'mkisofs -dvd-video /tmp/dvd_20161224-160756/iso | builtin_dd of
    ↪ =/dev/sr0 obs=32k seek=0'
I: -input-charset not specified, using utf-8 (detected in locale settings)
75.62% done, estimate finish Sat Dec 24 16:09:51 2016
Total translation table size: 0
Total rockridge attributes bytes: 0
Total directory bytes: 4096
Path table size(bytes): 42
Max brk space used 1a000
6624 extents written (12 MB)
/dev/sr0: "Current Write Speed" is 4.1x1352KBps.
builtin_dd: 6624*2KB out @ average 0.7x1352KBps
/dev/sr0: flushing cache
#
```

BD Example – Partial Output during writing disc media

```
growisofs -dvd-compat -Z /dev/sr0=/tmp/bd_20161224-155658/bd.udfs
Executing 'builtin_dd if=/tmp/bd_20161224-155658/bd.udfs of=/dev/sr0 obs=32k
    ↪ seek=0'
/dev/sr0: "Current Write Speed" is 2.0x4390KBps.
1605632/24524800 ( 6.5%) @0.0x, remaining 1:39 RBU 100.0% UBU 0.0%
1605632/24524800 ( 6.5%) @0.0x, remaining 2:22 RBU 100.0% UBU 100.0%
1605632/24524800 ( 6.5%) @0.0x, remaining 3:05 RBU 100.0% UBU 100.0%
5865472/24524800 (23.9%) @0.3x, remaining 0:54 RBU 100.0% UBU 33.3%
11829248/24524800 (48.2%) @0.4x, remaining 0:21 RBU 75.8% UBU 37.5%
17858560/24524800 (72.8%) @0.4x, remaining 0:08 RBU 39.8% UBU 79.2%
23789568/24524800 (97.0%) @0.4x, remaining 0:00 RBU 4.5% UBU 4.2%
builtin_dd: 11984*2KB out @ average 0.2x4390KBps
/dev/sr0: flushing cache
```

14.2 Debugging DVDs Creation

This section contains helpful hints, how to initially check the results, and some information on determining what might have gone wrong and how to address it.

1. For first time users, taking the defaults seem to work very well when running as root.
2. You may want to use rewritable media to see how it goes before using permanent media.
3. Until you are familiar with the procedure, start with shorter input in order not to waste time.
4. Test the generated output with a compatible media rendering tool before burning DVDs.
5. Check the list of files and file sizes after the batch jobs are complete before burning DVDs.

For blu-ray creation, `cd /workpath/bd_date-time` directory and look for similar files:

`bd.jobs  bd.m2ts  bd.sh  bd.udfs  bd.xml`

`udfs` directory which is used as a loopback mount point

Note that the size of *bd.udfs* should be larger than `bd.m2ts` because this is the final file which is actually going to be written to the disc media. It contains contents of `bd.m2ts` and all of the required disc structure.

For DVD creation, `cd /workpath/dvd_date-time` directory and look for similar files:

`dvd.ac3  dvd.jobs  dvd.m2v  dvd.mpg  dvd.sh  dvd.xml`
`iso` directory with VIDEO_TS and AUDIO_TS subdirectories of non-zero size.

Note that there will be no files in the actual AUDIO_TS directory.

6. The `bd.sh` and `dvd.sh` files are script files that you can carefully run manually from some start point to determine where the failure occurred. You must BE CAREFUL and know what you are doing and what directory you are in because dvd.sh contains an `rm` command and will delete files. The script takes a command line parameter of the directory where the file was rendered to and which is usually the directory where dvd.sh or bd.sh was created.
7. There is also a file in the same directory, called bd.jobs. It was the information that was used in creating the batch jobs and may be helpful in determining what parameters were actually used if there are any resulting problems. With enough background knowledge, you can make changes and rerun.
8. For blu-ray check to make sure you do not have any spurious loopback disks mounted that may interfere with the correct generation. Use the df command to check this

and then the umount command to unmount these. Also, check to make sure you have used the gsettings command to disable automount.

9. For blu-ray loopback mount the `<target>/bd.udfs` image, and see if it has the BDMV filesystem written to it, and in particular a subdirectory named STREAM. Look at the results in `./udfs` and check for the stream file which should exist in `./udfs/BDMV/STREAM/00000.m2ts` and should have the same size as `./bd.m2ts`.

```
mount -o loop <target>/bd.udfs <target>/udfs
ls -lR <target>/udfs
du -sc <target>/udfs
umount <target>/udfs
```

Checklist for Troubleshooting

- Are you logged in as root? This is required in order to loopback mount files for bluray and to write media on `/dev/hardware`. See section 13.7 for a workaround for normal user mode.
- Did you startup CINELERRA-GG from a terminal window so you can see informative messages?
- Is udftools installed for BD and dvdauthor installed for SD?
- Do you have loopback not enabled for bluray? At least temporarily, disable automount via:

```
gsettings set org.gnome.desktop.media-handling automount false
```

- Did you have sufficient disk space for working/writing files? In the *Create* window, the *disk space* will be displayed in green if sufficient, but in red if less than what fits on the disc media.
- Did you use `/tmp` as the work device, then rebooted the computer, which deleted files on `/tmp`?
- If the input media is interlaced, did you check the Deinterlace option to eliminate interlacing?
- Did you change the output name in the *Batch Render* window after the batch jobs were already created? These filenames have already been written to disk. If you want to change either the Title or the *Work_path*, you have to start over.
- Have you selected a Title in the Create window that is a directory that already exists? The program attempts to create that directory and will give you an error message if it exists.

- Did you replace the `/dev/bd` or `/dev/dvd` on the command line with your hardware device name?
- If a warning was issued in the Create BD/SD window of * *non-standard format* and your bluray reader could not play the disc, did you change to a standard format instead?
- Did you correctly interpret the frame rate if using interlaced format to be $\frac{1}{2}$ due to interlacing?

14.3 DVD-Subtitles

CINELERRA-GG supports the .udvd format of DVDs, so you can create subtitles on the timeline and then save them as .udvd files. For how the subtitle tool works see: Subtitles The subtitle format for Blu-Ray is PGS with a .sup extension which CinGG does not support. It is an image format (also 3D) so it can be created, converted and extracted only via specific OCR. You can use external programs like ffmpeg; SubTitle Editor; MKVToolNix; etc.

14.4 Dvd Interlaced Chroma

CINELERRA-GG uses 4:4:4 colorspace to edit, so it is necessary to convert interlaced 4:2:0 video to 4:4:4. But you can run into problems, referred to as the *chroma bug*, which you see in DVD media displayed on higher resolution monitors – streaks or spiky horizontal lines are visible in the chroma channel, especially on diagonal edges. The Chroma Bug is specific to MPEG and 4:2:0 encoding.

Now you can use the *YUV420P DVD Interlace Mode* when rendering DV directly to mpeg2 through a yuv4mpeg stream and when using video effects on HDV video.

With this option enabled, improved chroma results will be obtained from your DV or HDV source. Editing DV or HDV and rendering it back to the same format does not require any special handling. In order to perform colorspace conversions correctly in CINELERRA-GG and avoid Chroma errors for interlaced 4:2:0 video, check the box as follows:

`Settings → Preferences, Interface tab, Operation section, Use yuv420p dvd interlace format`

This option maintains the interlacing in Chroma sample addressing, which ordinarily would be deleted because the upsampling of interlaced chroma fields is normally done using a progressive algorithm. With this mode enabled, the MPEG decoder uses a different algorithm for interlaced frames so that the 4:2:0 format chroma interlacing is preserved.

14.5 DVD with LPCM or MP2 audio

By default, the audio when creating a DVD is always *AC3*. However you can switch to *PCM* (Pulse Code Modulator) or MP2 with just a few additional steps as outlined below. Note that Audio must be 48*Khz* or 96*Khz*, nothing else is supported, even by ffmpeg's dvd pcm encoder.

1. Start CINELERRA-GG from a terminal window so you will be able to see what is happening. Only the final step when you actually want to write to a DVD media writer, requires privileges of either root or the system has granted the user this privilege.
2. Make sure you have your video with audio loaded and edited to your satisfaction and you are positioned at the start of the video.
3. Use the `File` pulldown and select the *DVD Render* option.
4. In the*Create DVD* window, accept the defaults or select different values and then click `OK`. Do not check "Use FFMPEG" as that may not work.
5. When the *Batch Render* window pops up, in the big box towards the bottom will be 2 lines with the first line for *Video* already highlighted. Instead, click on the second line, which is for *Audio*, so that it is highlighted.
6. On the top left, you will see the *File Format* set as AC3. Use the down arrow next to the box and change it to *Raw PCM* or *MPEG Audio* by clicking on it.
7. When you switch to Raw PCM or MPEG Audio, you see the extension in the `Output path` above change to pcm or mp3 instead of ac3. Now just reset the extension from pcm to lpcm or mp2 as that is required. In most cases if you click on the `Audio` wrench to see the settings, you will find that the standard settings of 16 bit Linear / Signed / Hi Lo work for Raw PCM. For mp2, you will have to click on the Audio wrench and change the default Layer III to Layer II.
8. Now click on the `Start` box in the bottom left hand corner and CINELERRA-GG will process what it can of the job and put you back at your terminal startup window.
9. You will see a few lines of output, some of which are shown below, to include the ERROR:

```
running /dev/shm//dvd_20240116-182336/dvd.sh
INFO: [mplex] mplex version 2.1.0 (2.2.7 $Date: 2012/11/17 01:55:16 $)
**ERROR: [mplex] Unable to open file
/dev/shm/dvd_20240116-182336/dvd.ac3 for reading.
```

10. Change directory to the location as shown on the terminal window of `dvd.sh`.

11. Using an editor, modify the line in dvd.sh to change `dvd.ac3` to `dvd.lpcm` or to `dvd.mp2` for mp2. In addition ONLY for Raw PCM you have to change the mplex parameter to include:

    ```
    -L 48000:2:16
    ```

    The full line will look like this:

    ```
    mplex -f 8 -L 48000:2:16 -o $dir/dvd.mpg $dir/dvd.m2v $dir/dvd.lpcm
    ```

12. Now the script is ready to run in the same manner it would have had it been ac3. That is just run via:

    ```
    ./dvd.sh
    ```

13. Check to make sure there are no errors in the output shown on the window and proceed as usual.

14.6 MPEG utility programs

There are 2 utility programs that come in handy when dealing with DVD media for creating or reading previously written DVDs.

zmpeg3cc2txt convert closed captioning data to subtitle data.

This program can be used to scan captured broadcast data streams and convert the closed captioning text data into subtitle track data. The result can be used to add subtitles to DVDs created from the captured media using the text collected from the closed captioning.

usage:

```
./zmpeg3cc2txt [-c cc_service ] [-s start:length, ...] [ -t track ] [-v verbose ] [-w wdw
    ↪ mask ] [-x file.xml ] [-o file.udvd ] file.ts
```

cc_service	=	closed caption service id
start:length	=	start:length frames (comma separated list)
track	=	video track number
verbose	=	verbose level
wdw mask	=	bit mask for windows (-1 = all)
file.xml	=	filename for edl xml format subtitle data
file.udvd	=	filename for udvd format subtitle data
file.ts	=	filename for transport stream

To use this program, the input file must be a transport stream (broadcast video) which contains closed captioning services. The service id defaults to one, and the default video track is zero. Either `-o` or `-x` must be specified to indicate the output file format desired. If the output file name is a '-' then stdout is selected as the output file. For example:

```
zmpeg3cc2txt -o -/dvb_data/channel5.ts
```

zmpeg3ifochk check DVD `ifo` file for usable features.

For some time, DVD manufacturers have been employing a variety of measures to make reading a DVD difficult on a computer. One technique which is widely deployed is to add a bunch of extra program data, so that correct playback is only likely if you can read the DVD virtual machine data and decode a maze of program data to find the undamaged stream definitions. Only a few streams are created which are machine usable, and dozens are created as decoy streams. The decoy streams fail or introduce errors. This program scans the IFO (info file) playlists and verifies the contents of the stream that does not contain obvious damage. The result is a list of program ids which can be entered into the playback preferences to select a program which qualifies.

14.7 Creating Blu-ray Without Re-encoding

If you have video that is already in Blu-ray format AND you do not want to use CINELERRA-GG to do any editing, you can create blu-ray video media using Cinelerra's *bdwrite* program. This could be the case if you are working with media from a modern Digital Camcorder. Camcorders usually record in a legacy tape-based format or the current file-based format. The most common digital camcorder formats and whether or not they need to be re-encoded are:

- DV (1995) to SD-blu-ray video - process in CINELERRA-GG using the File pulldown of BD Render as previously described or use programs documented in their manuals to re-encode video to mpeg2 and then re-encode DV LPCM audio to Blu-ray LPCM.

- HDV (2004) to blu-ray video - process in CINELERRA-GG using the File pulldown of BD Render as previously described or use programs documented in their manuals to copy mpeg2 video and then re-encode MP2 audio to Blu-ray LPCM or AC3.
- AVCHD (2006-current) to blu-ray video; these are the types of videos that do not have to be re-encoded if you do not want to edit anything.

For an explanation of the differences between HDV vs AVCHD see: http://www.differencebetween.net/technology/difference-between-hdv-and-avchd

Besides digital camcorder file-based format of AVCHD, there are video files in MTS format that are also candidates for burning to blu-ray video media without re-encoding. An MTS file is a video file saved in the high-definition (HD) MPEG Transport Stream video format/H264, commonly called *AVCHD*. It contains HD video compatible with Blu-ray video disc format and is based on the MPEG-2 transport stream. MTS files are often used by Sony, Panasonic, Canon and other HD camcorders. Legal input for Video – MPEG1VIDEO, MPEG2VIDEO, H264; Audio – MP1, MP2, AC3, AC3PLUS, DTS, TRUHD.

For creating a blu-ray video disc, if you have these HD MPEG-2 media types that are in blu-ray video format, you can save the original quality of your work, rather than rendering it to another format. Note that for older media or media which used MP1 or MP2 audio codecs which will not work on blu-ray video discs, it may be necessary to first transcode the audio while leaving video intact. An example of the transcode line to use if using ffmpeg version 5.1 or higher is shown next. If using a lower ffmpeg version substitute ac3 for pcm_bluray:

```
ffmpeg -i hdv -c:v copy -c:a pcm_bluray -mpegts_m2ts_mode 1 new_hdv.mts
```

Follow the next 2 sets of steps to create the blu-ray video disc.

```
du -sb /yourHDVfile.MTS # Determine the size of your file in bytes.
blocks=((size-in-bytes/2048 + 4096)) # Convert bytes into blocks + a little more.
mkudffs -b 2048 /tmp/newfilename.udfs blocks # Create a file with that \# of blocks +
    ↪ some extra.
mount -o loop /tmp/newfilename.udfs /mntX # Use a mount point like mntX that is not in
    ↪ use.
```

In order to create the udfs file, you will need access to Cinelerra's *bdwrite* program. If you do not do your own build and compile, but use the AppImage instead, you will have to first extract *bdwrite* from the AppImage as described in the NOTE at B.10.1.

```
/<cinelerra_installed_path>/bin/bdwrite /mntX /tmp/yourHDVfile.MTS # Substitute \
    ↪ CGG{} path.
umount /mntX # You must unmount the udfs filesystem
growisofs -Z /dev/bd=/tmp/newfilename.udfs # Replace /dev/bd with your bluray
    ↪ hardware device.
OR dd if=/tmp/newfilename.udfs of=/dev/bd bs=2048000 # if using rewritable blu-ray;
    ↪ replace bd.
```

14.8 Blu-ray Workaround for Mount/Umount

Creating BD images to be written to media requires usage of *mount* and *umount* which typically can only be done by the root user due to security. If you want to avoid running CINELERRA-GG as root, you can implement a workaround by adding a line in `/etc/fstab` (must be root to edit the file initially) and by creating a directory in your home area, called *bluray*. You only have to do this once unless you upgrade the Operating System and it wipes out the line in `/etc/fstab`. Now the CINELERRA-GG program will automatically do the mount and umount for you each time you execute BD Render and you can run as an ordinary user.

The line to add to `/etc/fstab` will look something like the following, assuming your username is *name* and your groupid may be *users* or *name*. If you do an `ls -l` in your home directory, the 3^{rd} and 4^{th} fields shown will be your uid or name and gid or groupid which you must substitute in the line below.

```
/home/name/image /home/name/bluray udf noauto,loop,rw,user,uid=name,gid=
    ↪ groupid 0 0
```

Also, be sure to do a `mkdir bluray` in your `/home/name` directory as this is a requirement (owned by you; uid=gid=name). When the actual image to be written to disc media is created, it will first d any current `/home/name/image` file. Warning – make sure you do not already have a file called *image* that you want to save as it will be automatically deleted every time you initiate a BD Render. So you will want to burn a bluray disc after CINELERRA-GG creates the *image* since it will written over on the next rendition. The actual writing to your bluray burner (something like `/dev/sr0`) is done outside of CINELERRA-GG at a terminal prompt and requires root privilege usually. You can either use *sudo* for 1 line or create user wheel group to get around this.

14.9 Blu-ray from Multiple CINELERRA-GG Output

Writing prepared multiple CINELERRA-GG output files, `bd.m2ts`, to a single bluray video disc is relatively easy to do but is not done automatically. You can render all of the desired

files via the Create BD menu, save each individual `bd.m2ts` file with a unique name, construct a Menu Title that reflects the contents of each of these files, then manually use a few commands to create a udfs file to be written to blu-ray video media. If using AppImage, you will have to first extract the *bdwrite* program as explained in the NOTE at B.10.1. You can use any other software program that works instead.

Usage of the final preparation taken from the bdwrite program comments:

```
./bdwrite <tgt_dir_path> <playlist-0> <sep> <playlistp1> <sep> ... <sep> <playlist
    ↪ -n>

<sep> == -<pgm_pid> | --<pgm_pid> | ---<pgm_pid>
<pgm_pid> may be empty string, or a numeric pgm_pid for current title clip
<pgm_pid> defaults to first pgm probed
<playlist-x> == <clip-0.m2ts> <clip-1.m2ts> ... <clip-n.m2ts>
```

One title is built for each playlist; playlist-0 is used as first-play item. The separators (`<sep>` represented by the dash character) have unique roles. The double "——" means stop after playing, and the triple "———" means pause.

For example:

```
./bdwrite /tmp/dir /path/menu_titles.m2ts ---/path/clip0.m2ts --/path/clip1.m2ts --/
    ↪ path/clip2.m2ts
```

The basic idea is to use playlist-0 as a menu or directions to use the bluray player remote control to select the desired Title and start the play, avoiding the need for a menu system. Planning in advance to get the desired results is necessary. The following steps provide an outline to get started.

1. Create all of the `bd.m2ts` files that you want to put on the Bluray.
2. Using CINELERRA-GG, design your Title page using a few seconds of video and the *Title* plugin.
3. Use BD Create to render your short Title video.
4. Next is the most complicated part which is to run `mkudffs` with a sufficient amount of disk space to hold all of the `bd.m2ts` files *plus a little more!* To calculate this, you can record the sizes from having run BD Create mkudffs. This number is displayed on the terminal screen when using the command line interface each time and add them together. Or recalculate the size of each `bd.m2ts` using the formula below and adding them all together. This is the number of blocks used to make a bluray

image space for bdwrite to use. For many files, this could require a huge amount of space, so check first.

```
Total size = File0 size in bytes / 2048 + 4094 "+" File1 size in bytes / 2048 +
    ↪ 4094 "+" ...
```

Now create the image file via: `mkudffs image <Total size>` where image or udfs is the image name.

5. Loop mount the disk image (refer to Section 13.7).
6. Then actually write your multiple bd.m2ts type files onto the *image* where `<cin_path>` is the location of the CINELERRA-GG binary *bdwrite* file and `<path>` is your directory path. Below is a single line that wrapped around with 4 Titles.

```
<cin_path>/bin/bdwrite image /<path>menu_titles.m2ts ---/<path>/bd1.
    ↪ m2ts --/<path>/bd2.m2ts --/<path>/bd3.m2ts --/<path>bd4.m2ts
```

Note that the 3 dashes after the `menu_titles.m2ts` lets the bluray player know to *pause* after playing the few seconds video which contains the index to the rest of the files. The 2 dashes after each of the `bd.m2ts` signify *stop*. That is when you will have to use your remote to *Search Titles* in order to play the next one you want to see. In addition, if for some reason you just want to *play all*, you will have to add another line to the Title menu as a choice and list all of the 4 files in a row at the end of your bdwrite line without any dashes in between.

7. Umount the loopback disk.
8. Use your favorite *dd* or *growisofs* tool to write to a formatted bluray disc.

Figure 14.8 demonstrates an example of setting up a Title menu on a 5 second video. There is a list of 4 menu title options that can be searched via remote control using the Title search option for your player.

14.10 Use Case: DVD

Example of Video Source with 4:3 Aspect Ratio, Being Transcribed to 16:9 and Creating a DVD to be Displayed on a Digital TV. Illustrated steps to take source input with 4:3 aspect ratio and convert to 16:9, with the bottom of the image being cropped in order to preserve top of video so nobody's head gets cut off are provided here.

1. In order to write to a DVD writer hardware device, you must be *root*!!

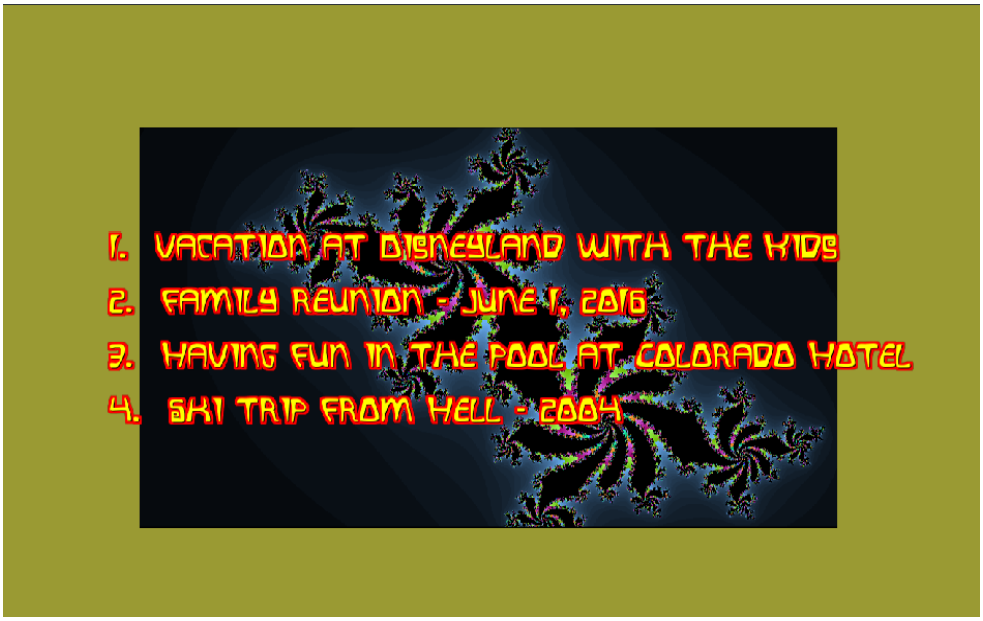


Figure 14.8: *Title menu for DVD/BD*

2. Start CINELERRA-GG to bring up the 4 usual screens with main track canvas in the lower left corner.

3. Load media via the pulldown `File` → `Load files...` by choosing the directory path with the desired file.

4. Bring up the *Create DVD* window using `File` → `DVD Render`.

5. Choose Format: PAL or NTSC with 16x9 aspect ratio for today's digital TV as in below screenshot. figure 14.9.

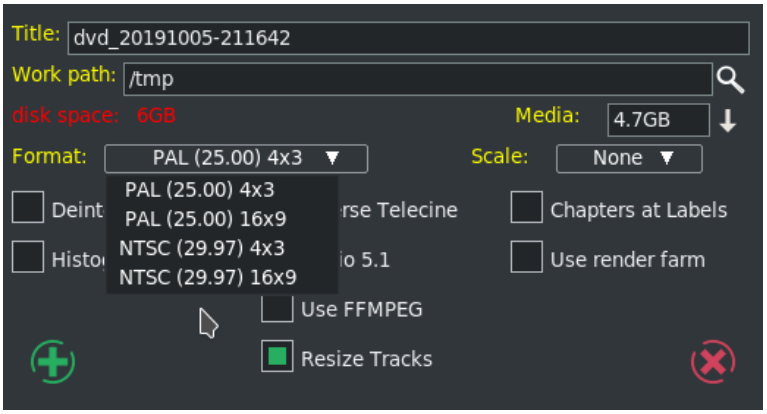


Figure 14.9: *Choose NTSC or PAL for DVD creation*

6. Modify the *Work path:* parameter to a disk that has sufficient disk space and you will see the amount of disk space in green letters (`/tmp` is default, but is often deleted so may be a bad choice).

7. Note that in the following screenshot, Scale of *Cropped* has been chosen (figure 14.10).

Figure 14.10: *Set Scale Ratio to Cropped*

8. Click the green checkmark on the bottom left side of the window to close it and proceed.

9. Now the *Batch Render* window will appear along with the Scale Ratio brown-colored line below the video in the main track canvas as seen in the screenshot below. Note that in this screenshot the top right most corner of the shot displays the bottom portion of the Compositor window.

10. Next, right mouse click the gold-colored magnifying glass, which is on the right hand side of the brown-colored line. This will bring up the Scale Ratio window that you can see below. Note in the Compositor window, the blue legs are only showing up to the knees.

With the *Scale Ratio* plugin you can manipulate your video so that it will look the way you want it on a different output Display device. In this case we are going to create a DVD for playing on a Digital TV screen.

In figure 10.64, the left side shows the Input Ratio, Width, and Height of input. The top right half shows desired output values. In this particular case, the input was a YouTube video which was not quite 4 : 3 aspect ratio.

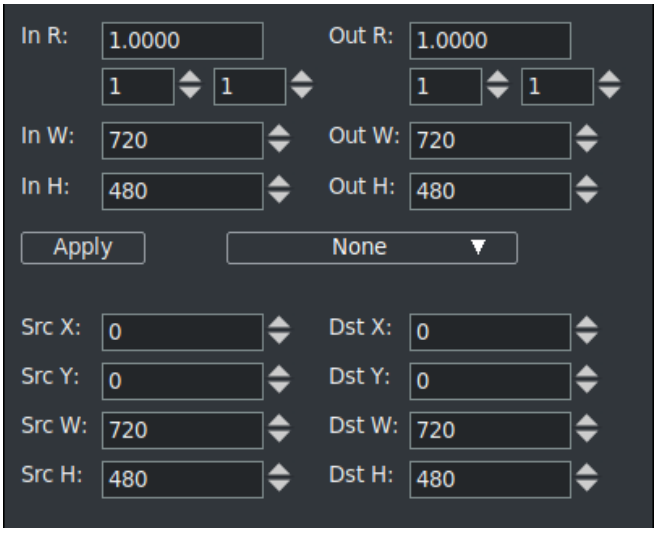


Figure 14.11: *Scale Ratio plugin*

Left and right sides of the bottom portion show the Source and the Destination X, Y, W, H values. As you change the values on the left side, you can see how this will affect the output as you observe the results in the compositor window. For example, as you change the values for SrcY in a *cropped* Scale scenario, you see up/down movement.

Keep in mind that the monitor you are using is NOT the intended output display device – your digital TV is, which most likely will have different looking aspect/pixels, etc.

11. In order to *crop* the bottom of the video in order to preserve all of the image on the top, modify the Src Y value on the bottom of the left hand side in the Scale Ratio plugin. Src Y which was 21 has now been changed to −18. You will see in the Compositor window how the bottom dark colored border is now gone so that none of the top portion which contains the Bunny's head will be chopped off. Compare figure 14.12 to figure 14.10 and note the Bunny's head.

12. Click the *Apply* box in the Scale Ratio window.

13. Click the *Save to EDL Path* in the *Batch Render* window for creating a DVD. If you do NOT do this, you will get a Warning box as seen in figure 14.13, to remind you to Save because you have changed the EDL by modifying the scaling parameters in the Scale Ratio window.

Figure 14.12: *Better scale on compositor*

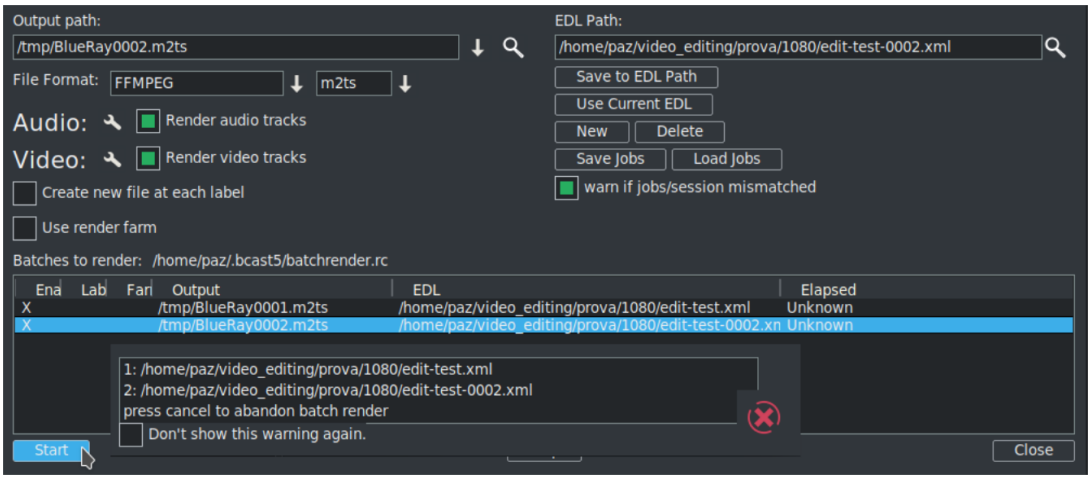


Figure 14.13: *Error in Batch Render*

13. Next, make sure you have the Timeline set in the Main window at the beginning of where you want to start rendering. Also, make sure the first line in the *Batches to render* section is highlighted as you can see above by the blue highlighting. Click on the *Start* box in the Batch Render window and you will see the video playing in the Compositor window.

14. CINELERRA-GG program will be stopped when done rendering; you will be at the terminal prompt where you will see it has printed out some informational messages (or errors if problems), the last 2 are:

```
To burn dvd, load blank media and run:
growisofs -dvd-compat -Z /dev/dvd -dvd-video /mnt0/dvd_20161027-
    ↪ 131723/iso
```

15. Load a blank or rewritable DVD into your DVD writer device, which will be similar to `/dev/dvd` as in the *growisofs* line above – something like `/dev/sr0` on your computer.

16. Keyin the *growisofs* line, substituting your actual writer device name. Again, you must be root.

17. When back at the terminal prompt, and if there are no errors, keyin `eject /dev/dvd` substituting.

18. Play it on your DVD player connected up to your Digital TV screen.

Chapter 15

Shortcuts

In Cinelerra-GG Infinity a lot of shortcuts are defined for various operations. Although it is quite possible to do everything with the mouse, the extensive use of shortcuts can make the editing process much more convenient and efficient. However, with so large a variety of shortcuts it is not uncommon that some of them may interfere with assignments of the same shortcuts in the user's desktop environment, keyboard language switcher, etc. For example in KDE, the Alt-Tab and Shift-Alt-Tab shortcuts are commonly assigned to switch between different desktop applications, and if so, these keystrokes would not be available to Cinelerra-GG Infinity. Another example, in Arch linux KDE: the combination Alt+LMB does not translate the mask, but the entire Compositor window. To translate the mask we must use Ctrl+Alt+LMB.

When having problems with some shortcut in Cinelerra-GG Infinity, it is recommended first to examine which set of shortcuts is assigned to the desktop environment and, if necessary, resolve the conflict. A special X11 application `xev` (or `xorg-xev`) can also help to test the keystrokes functionality (see `man xev`). Otherwise, Cinelerra-GG Infinity is completely desktop-neutral and has no requirements of some special window manager's support.

Here the shortcuts are listed organized by window and type. Some specific alternatives are listed in 2.2.6 in the "Key Alternatives" paragraph. Any reference to Alt or Ctrl always refers to the left one on the keyboard. In addition to this manual, you can view the shortcuts in html format via the *shell cmds* icon on the top, right corner of the main program window. There is also a **hotkey Alt/h**, that can be used just about anywhere to get help on a specific window, menu, item, tooltip, button, and other elements (see 18.1).

15.1 Main window

The Main window (also called the program window) consists of pulldown menus, buttons and keys.

Main menu pulldowns

Main window menu pulldowns

Identifier	Items	Shortcuts	Description
File	New	n	Create new project.
	Load files...	o	Load file(s).
	– load tool	Ctrl-a	Selects all files.
	– load tool	Ctrl-z	Deselects any highlighted files.
	Save	s	Save project.
	Save	Ctrl-s	Save project.
	Save As...	Shift-S	Save a project for easy moving.
	Export Project	Alt-s	Save a project for easy moving.
	Record...	r	Open record window.
	Scan...	Ctrl-Alt-s	Open dvb scan window.
	SubTitle...	Alt-y	Open subtitle script window.
	Render...	Shift-R	Open render window.
	Export EDL...	Shift-E	Export to CMX3600 format.
	Batch Render...	Shift-B	Open batch render window.
	BD Render...	Ctrl-Shift-D	Open create bluray disk window.
	DVD Render...	Alt-D	Open create dvd disk window.
	Quit	q	Quit the program.
	Dump EDL		Write current EDL to stdout.
	Dump Plugins		Write defined Plugins to stdout.
	Load Backup		Load auto-backup file “backup.xml".
	Save Backup	b	Save auto-backup file “backup.xml".
Edit	Undo clear	z	Undo from history previous operation.
	Undo clear	Ctrl-z	Undo from history previous opration.
	Redo	Shift-Z	Redo from history next opration.
	Cut	x	Copy selection to cut buffer and delete.
	Split	x	Split edit into 2 sections at hairline cursor.
	Paste	v	Insert selection from cut buffer.
	Clear	Del	Delete selection.
	Paste Silence	Shift-Space	Paste empty edit at selection or frame if none.
	Mute Selection	m	Replace selection with empty edit.

Main window menu pulldowns			
Identifier	**Items**	**Shortcuts**	**Description**
	Trim Selection		Replace track with selection.
	Select All	a	Toggle select all/none.
	Shuffle Edits		Randomly reorder track edits.
	Reverse Edits		Reverse track edits.
	Edit Length...		Change selected track given duration.
	Align Edits		Change selected corresp. track edits to start/end same.
	Trans. Length...		Change selected track transitions to given duration.
	Detach trans.		Remove selected track transitions.
	Clear labels		Delete selected timeline label markers.
	Cut ads		Remove selected region, add to commercial trking db.
	paste subttl	y	Replace selected subtitle edit text w/ subtitle script text.
Keyframes	Cut Keyframes	Shift-X	Copy selected keyframes to cut buffer and delete.
	Copy Keyframes	Shift-C	Copy selected keyframes to cut buffer.
	Paste Keyframes	Shift-V	Paste keyframes in cut buffer.
	Clear Keyframes	Shift-Del	Delete selected keyframes.
	Change to linear		Change keyframe auto curves to piecewise linear.
	Change to smooth		Change selected keyframes auto curves to bezier.
	Create curve type		Set new keyframe type: smooth,linear,tangent,disjoint.
	Copy default key	Alt-c	Copy default keyframe to clipboard.
	Paste default key	Alt-v	Paste clipboard keyframe into default keyframe.
Audio	Add track	t	Append new empty audio track.
	Default Trans.	u	Insert default audio transition at selected edit boundary.
	Map 1 : 1		Setup audio mixing output = input.

Main window menu pulldowns

Identifier	Items	Shortcuts	Description
	Map 5.1 : 2		Setup audio mixing stereo = 5.1.
	Attach Trans.		Open audio transition select for insert at edit boundary.
	Attach Effect...		Open audio selection menu for insert at edit boundary.
	Render Effect...		Open audio render select to render from select as pcm.
Video	Add track	Shift-T	Append new empty video track.
	Default Trans.	Shift-U	Insert default video transition at selected edit boundary.
	Attach Trans.		Open video transition select for insert at edit boundary.
	Attach Effect...		Open video effect selection for insert at edit boundary.
	Render Effect...		Open video render select to render from select as pcm.
Tracks	Move tracks up	Shift-Up	Swap tracks up.
	Move trks down	Shift-Down	Swap tracks down.
	Roll tracks up	Ctrl-Shift-Up	Circulate tracks up.
	Roll trks down	Ctrl-Shift-Down	Circulate tracks down.
	Delete tracks		Delete all tracks.
	Delete last track	Ctrl-d	Delete last track.
	Delete first track	Shift-D	Delete first track.
	Concatenate trks		Concatenate tracks together.
	Append to Proj.	Shift-N	Add set of tracks.
	add Subttl	Shift-Y	Add subtitle track.
Settings	Format...	Shift-F	Open render format setup menu.
	Preferences...	Shift-P	Open preferences setup menu.
	Proxy Settings	Alt-r	Bring up Proxy settings window.
	Transcode	Alt-e	Bring up the Transcode menu.
	Align cursor...	Ctrl-a	Align cursor on frames.
	– Edit labels		Toggle labels follow edits.
	– Edit effects		Toggle plugins follow edits.
	– Keyfrs fol. edits		Toggle keyframes follow edits.
	– Typeless keyfrs		Toggle typeless keyframes mode.
	Save settings now	Ctrl-Shift-S	Save CINELERRA-GG_rc.

Main window menu pulldowns

Identifier	Items	Shortcuts	Description
	Loop Playback	Shift-L	Set loop playback region to selection/all.
	Guide on timeline	Shift-L	Create a guide on timeline.
	Set bkg render	Shift-G	Toggle background rendering.
View	– Show assets	0	Toggle show asset data.
	– Show titles	1	Toggle show asset title.
	– Show trans.	2	Toggle show transition effects.
	– Fade	3	Toggle show fade auto curves.
	– Mute	4	Toggle show mute auto curves.
	– Overlay Mode	5	Toggle show mode auto curves.
	– Pan	6	Toggle show pan auto curves.
	– Plugin Autos	7	Toggle show plugin keyframes.
	– Mask	8	Toggle show mask auto keyframes.
	– Speed	9	Toggle show speed auto curves.
	– Camera X	Ctrl-Shift-X	Toggle show camera X auto curves.
	– Camera Y	Ctrl-Shift-Y	Toggle show camera Y auto curves.
	– Camera Z	Ctrl-Shift-Z	Toggle show camera Z auto curves.
	– Projector X	Alt-Shift-X	Toggle show projector X auto curves.
	– Projector Y	Alt-Shift-Y	Toggle show projector Y auto curves.
	– Projector Z	Alt-Shift-Z	Toggle show projector Z auto curves.
Window	– Show Viewer		Toggle show view window.
	– Show Resrs.		Toggle show resources window.
	– Show Compos.		Toggle show compositor window.
	– Show Overlays		Toggle show auto overlay window.
	– Show Levels		Toggle show audio levels window.
	– Split X pane	Ctrl-1	Toggle $\frac{1}{2}$ horiz track timeline window panes.
	– Split Y pane	Ctrl-2	Toggle $\frac{1}{2}$ vert track timeline window panes.

Main window menu pulldowns

Identifier	Items	Shortcuts	Description
	Mixer Viewer	Shift-M	Bring up a Mixer Viewer window.
	Drag Tile mixers	Alt-t	Create a drag box to be used to tile mixers.
	Default Positions	Ctrl-p	Reset window positions/size to defaults.
	Tile Left		Set window positions/sizes to tile left screen.
	Tile Right		Set window positions/size to tile right screen.
_-_	shell cmds	Top rt. Corner	Run scripts (setup in interface preferences).

Main menu buttons

Main window menu buttons

Identifier	Items	Shortcuts	Description
Transport	Rewind	home	Jump to beginning.
	Fast reverse	KP + / Alt-p	Double speed reverse play.
	Normal reverse	KP 6 / Alt-o	Normal speed reverse play.
	Frame reverse	KP 4 / Alt-u	Render previous frame.
	Stop	KP 0 / Alt-m	Stop playback.
	Frame forward	KP 1 / Alt-j	Render next frame.
	Normal forward	KP 3 / Alt-l	Normal speed forward play.
	Fast forward	KP Enter/A-;	Double speed forward play.
	Jump to end	end	Jump to end.
	(No button)	KP 5 / Alt-i	Slow speed reverse play.
	(No button)	KP 2 / Alt-k	Slow speed forward play.
	(No button)	Shift-KP 1	Render next frame and audio.
	(No button)	Shift-KP 2	Slow speed forward play without audio.
	(No button)	Shift-KP 3	Normal speed forward play without audio.
	(No button)	Shift-KP 4	Render previous frame and audio.
	(No button)	Shift-KP 5	Slow speed reverse play without audio.
	(No button)	Shift-KP 6	Normal speed reverse play without audio.

Main window menu buttons

Identifier	Items	Shortcuts	Description
		Ctrl-KP#	If [ ] set, "KP 2,3,5,6,+,enter" between in/out.
		+Shift w/Alt	Shift with Alt+x above, adds/removes audio.
		+Shift w/Ctrl	Loops play, all or between In-/Out.
Edit Modes	Auto Gen. Keys	j	Generate keyframes while tweeking.
	Lock Labels		Lock labels from moving.
Edit Operations	In point	'[' or '<'	Toggle In point timeline marker.
	Out point	']' or '>'	Toggle Out point timeline marker.
		Ctrl-t	Clear both In and Out pointers.
	To clip	i	Copy selection and create clip.
	Split	x	Split edit into 2 sections at hairline cursor.
	Cut	x	Copy selection to cut buffer and delete.
	Copy	c	Copy selection to cut buffer.
	Paste	v	Insert selection from cut buffer.
	Toggle label	l	Toggle label at current position.
	Previous label	Ctrl ←	Move to label before cursor.
	Next label	Ctrl →	Move to label after cursor.
	Next label select	Ctrl-Shift →	Select from cursor to next label.
	Prev label select	Ctrl-Shift ←	Select from cursor to previous label.
	Next keyframe	k	Move to next keyframe.
	Previous keyframe	Ctrl-k	Move to previous keyframe.
	Next keyframe select	Shift-K	Select from cursor to next keyframe.
	Previous keyfr select	Ctrl-Shift-K	Select from cursor to previous keyframe.
	Next auto	Alt-a	Move to next auto.
	Previous auto	Ctrl-Alt-a	Move to previous auto
	Next auto select	Alt-Shift-A	Select from cursor to next auto.
	Previous auto select	Ctrl-Alt-Shift-A	Select from cursor to previous auto.
	Previous edit	Alt ←	Move to previous edit.

Main window menu buttons

Identifier	Items	Shortcuts	Description
	Next edit	Alt →	Move to next edit.
	Cut left label	Ctl-alt-shift-'<'	Cut from insert pointer to previous label.
	Cut right label	Ctl-alt-shift -'>'	Cut from insert pointer to next label.
	Cut previous edit	Ctrl-alt- ','	Cut from insert pointer to previous edit.
	Cut next edit	Ctrl-alt- '.'	Cut from insert pointer to next edit.
	Fit	f	Fit time displayed to selection.
	Auto Fit	Alt-f	Fit the max/min range of all automation types.
	Auto Fit	Ctrl-alt-f	Fit the max/min range of current automation types.
	Undo	z	Undo from history previous operation.
	Redo	Shift-Z	Redo from history next operation.
	Manual Goto	g	Jump to time selected by popup.
Drag/Drop Edits	Clear Select	Ctrl-Shift-A	Deselect all selected edits.
	Select Edits	Ctrl-Alt-'	Add highlight to selected edits.
	LMB	Alt + Drag	Drag select.
	LMB	Ctrl+Alt + Drag	Drag deselect.
	Copy	Ctrl-c	Copy selected edits into copy buffer.
	Cut	Ctrl-x	Delete selected edits/put in buffer/collapse.
	Mute	Ctrl-m	Delete selected edits/put in buffer/insert space.
	Mute	Backspace	Delete selected edits/put in buffer/insert space.
	Copy Pack	Ctrl-Shift-C	Copy selected edits and pack together.
	Cut Pack	Ctrl-Alt-z	Delete selected edits/put in buffer/collapse.
	Mute Pack	Ctrl-Shift-M	Delete selected edits/put in buffer/insert space.

Main window menu buttons

Identifier	Items	Shortcuts	Description
	Paste	Ctrl-v	Paste (splice) buffer at insertion point or reticle.
	Overwrite	Ctrl-b	Paste (overwrite) buffer at insertion pt/reticle.
		Ctrl / LMB	Toggles selected of group/edit.
		LMB	If “Clear before toggle”, only select this edit/group.
		LMB	If not “Clear before toggle”, toggle edit/group.
		Double click	Only select this edit/group.
		Double click-Ctrl	Selects column and highlights.
	LMB	Shift	Group/ungroup selected edits/-group.
	LMB	Ctrl + Drag	Column drag.
	LMB	Drag	Clip drag.
	LMB	Drag	Drag unselected edit under cursor.
FFMpeg probes early/late	FFMPEG early	Top rt. Corner	Run ffmpeg open probes first / last.

Main menu Keys

Main window menu keys

Identifier	Key	Qualifier	Description
Key	e		Toggle editing mode: from cut&paste or drag&drop.
	1 through 9		Toggle auto mode settings as in main menu shortcuts.
	1 through 8	Alt	Select asset pgm 1 — 8; reset format (remove other trks).
	← (left arrow)		Scroll window timeline display left (not insertion pt).
	← (left arrow)	Ctrl	Move cursor to label left of cursor (main shortcuts).
	← (left arrow)	Alt	Move cursor to edit boundary left of cursor position.

Main window menu keys

Identifier	Key	Qualifier	Description
	← (left arrow)	Shift-ctrl	Move cursor to label left of cursor, expand selection.
	← (left arrow)	Shift-alt	Move cursor to edit boundary left of cursor&expand.
	→ (right arrow)		Scroll window timeline display right (not insertion pt).
	→ (right arrow)	Ctrl	Move cursor to label right of cursor (main shortcuts).
	→ (right arrow)	Alt	Move cursor to edit boundary right of cursor position.
	→ (right arrow)	Shift-ctrl	Move cursor to label right of cursor, expand selection.
	→ (right arrow)	Shift-alt	Move cursor to edit boundary right of cursor&expand.
	, (comma)		Scroll window timeline display left (not insertion pt).
	, (comma)	Ctrl	Move cursor to label left of cursor (main shortcuts).
	, (comma)	Alt	Decrease auto curve limits (zoombar selected curve).
	. (period)		Scroll window timeline display right (not insertion pt).
	. (period)	Ctrl	Move cursor to label right of cursor (main shortcuts).
	. (period)	Alt	Move cursor to edit boundary right of cursor position.
	↑ (up arrow)		Increase timeline duration (zoom out timeline).
	↑ (up arrow)	Ctrl	Increase audio sample waveform scale.
	↑ (up arrow)	Alt	Increase auto curve limits (zoombar selected curve).
	↑ (up arrow)	Ctrl-alt	Increase all auto curve limits.
	↑ (up arrow)	Shift	Move all tracks up by 1.
	↓ (down arrow)		Decrease timeline duration (zoom in timeline).
	↓ (down arrow)	Ctrl	Decrease audio sample waveform scale.
	↓ (down arrow)	Alt	Decrease auto curve limits (zoombar selected curve).

Main window menu keys

Identifier	Key	Qualifier	Description
	↓ (down arrow)	Ctrl-alt	Decrease all auto curve limits.
	↓ (down arrow)	Shift	Move all tracks down by 1.
	PGUP (page up)		Scroll current track vertical window pane up.
	Mouse wheel	UP	Scroll current track vertical window pane up.
	PGUP (page up)	Ctrl	Increase track scale geometry (zoom in track data).
	PGDN (page down)		Scroll current track vertical window pane down.
	Mouse wheel	Down	Scroll current track vertical window pane down.
	PGDN (page down)	Ctrl	Decrease track scale geometry (zoom out track data).
	Home		Move insertion point to beginning of timeline.
	End		Move insertion point to end of timeline.
	Mouse wheel	Shift + up	Zoom out view without resizing the window.
	Mouse wheel	Shift + down	Zoom in view without resizing the window.
	Mouse wheel	Ctrl + up	Move left on the timeline without resize.
	Mouse wheel	Ctrl + down	Move right on the timeline without resize.
		Ctrl + down	Move right on the timeline without resize.
		Shift+click	Over edit causes highlight section to extend to cursor.
		Shift+click	Over boundary of effect, trims only that effect.
		Shift+click	Over Hard Edge of Blade Cut, toggles marker.
		Shift+click	Between labels, highlights selection.
		Shift+click	Over Automation Range values, changes value.
	Toggle single trk	Tab	Toggle single track arming status.

Main window menu keys			
Identifier	**Key**	**Qualifier**	**Description**
	Toggle other trks	Shift-tab	Toggle all of the other tracks arming status.
		Double click	On plugin title bar, selects that area.
		Double click	On an edit, selects that area.
		Double click	On subtitle track, displays text in Line text.
		Double click	On fade/speed, synch video/audio ganged.
		Shift	While dragging auto, turns of gang.
		Double MMB	On auto or keyframe, select that position.
	U		Paste the last Video transition.
	u		Paste the last Audio transition.
	r	Ctrl	Proxy quick switch.
		Ctrl +!	Assigns the timecode of the asset to the timebar.
	F1	Shift	Toggle on/off all XYZ of camera.
	F2	Shift	Toggle on/off all XYZ of projector.
	F1	Ctrl+Shift	Use window layout #1.
	F2	Ctrl+Shift	Use window layout #2.
	F3	Ctrl+Shift	Use window layout #3.
	F4	Ctrl+Shift	Use window layout #4.
Timebar	LMB	Ctrl	Changes to the next time format.
	MMB	Ctrl	Change to the previous time format.
	LMB		Moves cursor position on the timeline.
	Click label	Hold LMB	Drag label to elsewhere.
	Click In/Out	Hold LMB	Drag In/Out pointer elsewhere.
Patchbay	Shift/Move	Hold LMB	On Fade slider bar, sets gain to 100% or $0db$.

15.2 Compositor window shortcuts

Compositor buttons

Compositor window buttons

Identifier	Items	Shortcuts	Description
Tools	Protect video	F1	Disable changes to the compositor output.
	Zoom view	F2	Zoom in view without resizing the window.
	Edit mask	F3	Brings up the mask editing tool.
	+ Edit mask	Shift-drag pt.	Hold down shift+left mouse to drag a point.
	+ Edit mask	Ctrl-drag pt.	Hold down Ctrl+left mouse to move control pointers.
	+ Edit mask	Alt-drag mask	Hold down Alt+left mouse to translate mask.
	+ Edit point	Shift-set pivot	Hold down Shift+middle mouse to set Pivot point.
	+ Rotate mask	Wheel-rotate	Wheel middle mouse to rotate around Pivot.
	+ Scale mask	Shift+Wheel	Hold down Shift+Wheel mouse to scale around pivot.
	+ Rotate/Scale	Ctrl+Wheel	Hold down Ctrl+Wheel to rotate/scale around pointer
	+ Delete mask	Shift + Del	Hold down Shift + click Delete button to delete mask.
	Ruler	F4	Brings up the ruler for usage.
	+ ruler hotkey	Ctrl	Lock ruler to nearest 45 degree angle.
	+ ruler hotkey	Alt	Translate the ruler.
	Adjust camera	F5	Brings up the camera editing tool.
	Adjust projector	F6	Brings up the projector editing tool.
	Crop layer/outpt	F7	Crop a layer or output.
	+ crop tool	Click-drag	In video, starts a new rectangle.
	+ crop tool	Click-drag	Over rectangle to reposition.
	+ crop tool	Alt-click	In cropping rectangle to translate.
	Get color	F8	Detects color at current spot & stores temporary.
	Show tool info	F9	Works with editing/cropping buttons.

Compositor window buttons

Identifier	Items	Shortcuts	Description
	Show safe regs.	F10	Draws safe regions in the video output.
		p	Turn on/off Click to Play.
		Shift+w/Ctrl	With transport btn. loops play; all or with [ ].
Transport	(plus 3 below)		Same as in Main window.
Edit	(plus 3 below)		Same as in Main window.

Compositor keys

Compositor window keys

Identifier	Key	Qualifier	Description
Keypresses	Mid. click+drag		Pans the view/moves the image.
	Middle mouse	Shift	Returns to Auto zoom.
	RMB		Brings up zoom%/auto;reset cam/proj;hide ctrls.
	LMB Zoom		Zooms in.
	Mouse wheel up		Zooms in.
	LMB Zoom	Ctrl	Zooms out.
	Mouse wheel dn		Zooms out.
	LMB Zoom	Shift-alt	Zooms out.
	'+' or '='		Zooms in video output.
	'-'		Zooms out video output.
	f		Toggle on/off fullscreen.
	F11		Reset camera.
	F12		Reset projector.
	LMB	Shift	Camera enabled - vertical motion zooms in/out
	LMB	Shift	Projector enabled - vertical motion zooms in/out
	F1	Ctrl+Shift	Use window layout #1.
	F2	Ctrl+Shift	Use window layout #2.
	F3	Ctrl+Shift	Use window layout #3.
	F4	Ctrl+Shift	Use window layout #4.
	Click label	Hold LMB	Drag label to elsewhere.
	Click In/Out	Hold LMB	Drag In/Out elsewhere.

Compositor window keys

Identifier	Key	Qualifier	Description
Timebar	LMB	Ctrl	Changes to the next time format.
	MMB	Ctrl	Change to the previous time format.
	LMB		Moves cursor position on the timeline.
	Click label	Hold LMB	Drag label to elsewhere.
		Hold LMB	Drag In/Out pointer elsewhere.
		Hold LMB	Preview region drag on blue-colored bar.
Transport		LMB	Only Protect enabled, start/stop forward play.
Edit		MMB	Only Protect enabled, start/stop play reverse.
		Middlewheel	Only Protect enabled, plays for-w/rev 1 frame.
		p	Turns on/off Click to Play button.

15.3 Viewer window shortcuts

Viewer buttons

Viewer window buttons

Identifier	Items	Shortcuts	Description
Transport	Rewind	home	Jump to beginning.
	Fast reverse	'KP +' /Alt-p	Toggle double speed reverse play.
	Normal reverse	'KP 6' / Alt-o	Toggle normal speed reverse play.
	Frame reverse	'KP 4' / Alt-u	Render previous frame.
	Stop	'KP 0' / Alt-m	Stop playback.
	Frame forward	'KP 1' / Alt-j	Render next frame.
	Normal forward	'KP 3' / Alt-l	Toggle normal speed forward play.
	Fast forward	'KP Enter'/A-;	Toggle double speed forward play.
	Jump to end	end	Jump to end.
	(No button)	'KP 5' / Alt-i	Toggle slow speed reverse play.
	(No button)	'KP 2' / Alt-k	Toggle slow speed forward play.

Viewer window buttons

Identifier	Items	Shortcuts	Description
		'KP#-Shift'	Shift added to KP #, adds or subtracts audio.
		'Ctrl-KP#'	If [ ] set, "KP2,3,5,6,+,enter" play between ptrs..
		+ Shift w/Alt	Shift with Alt+x above, adds/removes audio.
		+ Shift w/Ctrl	Loops play; all or between In-/Out.
		Left click	Start or stop play forward.
		Middle click	Start or stop play reverse.
		Middle wheel	Play forward or reverse 1 frame.
		p	Turns on/off Click to Play button.
Edit	In point	'[' or '<'	Toggle In point timeline marker
	Out point	']' or '>'	Toggle Out point timeline marker.
		Ctrl-t	Clear both In and Out pointers.
	Splice	v	Create splice.
	Overwrite	b	Overwrite.
	To clip	i	Copy selection and create clip.
	Copy	c	Copy selection to cut buffer.
	Splice	'v' + Shift	Create splice of the entire clip.
	Overwrite	'b' + Shift	Overwrite using the entire clip.
	To clip	'i' + Shift	Copy all and create clip.
	Copy	'c' + Shift	Copy entire clip to cut buffer.
	Show meters		Toggle show meters.
	Toggle label	l	Toggle label at current position.
	Previous label	'Ctrl ←'	Move to label before cursor.
	Next label	'Ctrl →'	Move to label after cursor.
	Manual Goto	g	Jump to time selected by popup.
Mixer		Double click	On mixer window, pastes into timeline

Viewer Keys

Viewer window keys

Identifier	Key	Qualifier	Description
Keypresses	Right click		Brings up fullscreen/zoom/close menu.

Viewer window keys			
Identifier	**Key**	**Qualifier**	**Description**
	F1	Ctrl+Shift	Use window layout #1.
	F2	Ctrl+Shift	Use window layout #2.
	F3	Ctrl+Shift	Use window layout #3.
	F4	Ctrl+Shift	Use window layout #4.
Timebar	LMB		Moves cursor position on the timeline.
	Double click		If between labels, selects that section.
	Click label	Hold LMB	Drag label to elsewhere.
	Click In/Out	Hold LMB	Drag In/Out pointer elsewhere.
	Fat arrow end	Hold LMB	Preview region drag on blue-colored bar.

15.4 Resources window Keys

Resources window keys			
Identifier	**Key**	**Qualifier**	**Description**
Keypresses	Drag Mouse	Shift	Hold down Shift and drag to move text icon.
		Double click	Draw vicons off- loads media in Viewer.
		Single click	Draw vicons on- bigger thumbnail plays or stops.
		Middle click	Draw on – brings up Inter-View modeling.
		Wheel move	Draw on – increase/decrease preview size.
		Mouse over	Draw on – instantly brings up thumbnail.
	v		Switch to next Display option.
	i		Switch on/off info for effects.
	o		Opens Load files... menu.
	- load tool	Ctrl-a	Selects all files.
	- load tool	Ctrl-z	Deselects any highlighted files.
	DEL key		Resources/Media, deletes highlighted from project.

Resources window keys

Identifier	Key	Qualifier	Description
	DEL key	Shift	Resources/Media, deletes highlighted from disk + OK.
	DEL key	Ctrl+Shift	Resources/plugin, deletes highlighted from disk + OK.
	F1	Ctrl+Shift	Use window layout #1.
	F2	Ctrl+Shift	Use window layout #2.
	F3	Ctrl+Shift	Use window layout #3.
	F4	Ctrl+Shift	Use window layout #4.
	In Labels folder	Double click	On a Label, timeline cursor moves to label.

15.5 Other windows

Other Buttons

Buttons in other windows

Identifier	Items	Shortcuts	Description
An Event		Double click	On mixer window, pastes into timeline.
Record	Start/pause	space bar	Start and pause recording of current batch.
	Label	l	Toggle label at current position.
		Double click	On channel in window, pops edit ch window.
Channels	Select		Select a channel.
	Add...		Add another channel.
	Edit...		Edit the wording associated with the channel.
	Move up		Move the channel up in the selection.
	Move down		Move the channel down in the selection.
	Sort		Sort all of the channels numerically.
	Scan		Scan the available channels.
	Delete		Delete the selected channel.
	Picture...		Select the device picture.

Buttons in other windows

Identifier	Items	Shortcuts	Description
		Double click	In Find window, cursor moves to hilite program.
		Double click	On batchbay line, selects and loads ch info.
Remote	MENU		Toggle remote control state (grabs/ungrabs keys).
	KPTV		Open or close the record window.
	KPBOOK		Start/stop dvb channel scan.
	KPHAND		Quit CINELERRA-GG program.
Commercial		Shift-A	Edit commercial

Other Keys

Keys in other windows

Identifier	Key	Qualifier	Description
Most windows	ESC		Cancels operation if no one grabs first.
	ENTER		Cancels operation if no one grabs first.
Overlays	LMB	Shift	Turns off all else, or reverts to previous state.
Textboxes	Tab		Move cursor to the next textbox.
	Tab	Shift	Move cursor to the previous textbox.
	Esc		Cancels any active suggestions.
	Left arrow	with ibeam	Backs up 1 character.
	Left arrow	Ctrl	Backs up to beginning of word/next word.
	Left arrow	Shift	Selects/Deselects previous character highlight.
	Left arrow	Ctrl-Shift	Selects/Deselects previous word highlight.
	Right arrow	with ibeam	Moves forward 1 character.
	Right arrow	Ctrl	Moves to beginning of next word.
	Right arrow	Shift	Adds selection highlight to the next character.

Keys in other windows

Identifier	Key	Qualifier	Description
	Right arrow	Ctrl-Shift	Expands highlight to the next word.
	Up arrow		Move up to next line, same position.
	Down arrow		Move down to next line, same position.
	Up arrow	Shift	Selects/Deselects up to the previous line X-coord.
	Down arrow	Shift	Select/Deselect down 1 line from the X-coord.
	PGUP (page up)		Goes to the beginning of the page.
	Pgdn (page down)		Goes to the end of the page.
	PGUP (page up)	Shift	Highlights selection above Ibeam to page start.
	Pgdn (page down)	Shift	Highlights selection below Ibeam to page end.
	End		Cancels selection and goes to end of line.
	End	Shift	Selects to end of line.
	Home		Goes to the beginning of the line.
	Home	Shift	Highlights from ibeam to beginning of line.
	Backspace		Deletes the character in back of the ibeam.
	Delete		Deletes the next character.
	Ctrl-c	(also+'Shift')	Copies selection to the 2nd cut buffer.
	Ctrl-v	(also+'Shift')	Pastes selection from the 2nd cut buffer.
	Ctrl-x	(also+'Shift')	Copies selection to 2nd cut buffer and delete.
		Double click	On word in textbox, highlites that word.
		Double click	On highlighted Probe order, enables/disables.
		Triple click	In subtitle script text, loads line into line text.
Unicode	Ctrl-U	Shift	Go into Unicode enter mode.

Keys in other windows			
Identifier	**Key**	**Qualifier**	**Description**
	Enter		Fires the Unicode.
	Esc		Cancels Unicode state.
	Backspace		Deletes previous character.
	A-F, 0-9		Enters the hex code.

15.6 Alternative Shortcuts

The alternative shortcuts are changed from CINELERRA-GG INFINITY default to what most other NLEs use as standard. For example:

`J, K, L` (Play Normal Reverse, Stop, Play Normal Forward) and when pressed twice J and L will Play Fast and then if playing fast and pressed again, will go back to playing normal.

`I, O` (set/unset InPoint, set/unset OutPoint)

`A, S` (Jump backward next cut, Jump forward next cut)

To use these alternative shortcuts you must use the appimage found here – you may want to go up 1 directory level to get a more recent dated version – or compile your own CINELERRA-GG with the the patch and the instructions found here: Notable Options and Caveats.

These shortcuts will only display in the English version for the caption/hints and the included shortcuts.html file in the "Shell Cmds" of the program does not include them. The texts in the menu and the captions of the icons, written with the '-' character between the Special key and the key in the default set, for example `Shift-s`, have been replaced by the '+' character. And the letters are always in capital letters although *Shift* is not pressed. This notation is more consistent in what many, if not all, the other NLE programs use. Some of the original default shortcuts like `BD Render... (Ctrl+Shift+d)` and `DVD Render... (Alt+d)` have been deleted because they are not used frequently. If a shortcut in the After column as shown in the following table has a '+' symbol (for example: `+ 'I'`), it means that it is an additional shortcut added to the old shortcut/s in the Before column and also that that shortcut was confiscated from a default CINELERRA-GG INFINITY shortcut where it had a different function.

These shortcuts are very useful because the keys more frequently used have priority in editing, and are easy to use with the left hand on the keyboard and the right hand on the mouse.

Alternative Shortcuts		
Description	**Before**	**After**
To clip	'i'	Ctrl+I
Scroll window timeline... left	LeftArrow, ','	LeftArrow
Scroll window timeline... right	RightArrow, '.'	RightArrow
Label	'l', ';'	'" (single quote)
Go To	'g'	Ctrl+G
Generate keyframes... tweeking	'j'	'G'
Jump forward... next Keyframe	'k'	Shift+Right Arrow
Jump backward... next Keyframe	Shift+K	Shift+Left Arrow
New Project...	'n'	Ctrl+N
Load files...	'o'	Ctrl+O
Save	's'	Ctrl+S
Save as...	Shift+s	Ctrl+Shift+S
Save Session	Ctrl+S	no shortcut
Select All	'a'	Ctrl+A
Deselect All	'a' or cursor move	Ctrl+A, Ctrl+Shift+A
Select Edits	Ctrl+Alt+'	Ctrl+Alt+A
Undo	'z', Ctrl+Z	Ctrl+Z
Redo	Shift+Z	Shift+Z
InPoint	'[', '<'	+ 'I'
OutPoint	']', '>'	+ 'O'
Play Normal reverse	NumKP 6, Alt+O	+ 'J' Normal&Fast
Play Stop	NumKP 0, Alt+M	+ 'K'
Play Normal forward	NumKP 3, Alt+L	+ 'L' Normal&Fast
One Frame back	NumKP 4, Alt+U	+ ',' (comma)
One Frame forward	NumKP 1, Alt+J	+ '.' (period)
Jump backward to the next cut	Alt+Left Arrow	+ 'A'
Jump forward to the next cut	Alt+Right Arrow	+ 'S'
Load window: Select All files	Ctrl+A	Ctrl+A
Load window: Deselect All files	Ctrl+Z	Ctrl+Shift+A
BD Render...	Ctrl+Shift+d	no shortcut
DVD Render...	Alt+d	no shortcut
Delete last track	'd'	Ctrl+D
Quit	'q'	Ctrl+Q
Settings → Save settings now	Ctrl+Shift+S	Shift+S
Settings → Align cursor on frames	Ctrl+A	Ctrl+F

A common practice in video editing is the ability to jump from one part of the timeline to another: jumping from one *cut* to another or from one *label* to another or even from one *keyframe* to another. Using shortcuts in these cases is even more advantageous. In the next table we summarize these alternative shortcuts of the main jumps.

Alternatives - Some jumps backward and forward	
Jump backward to the next cut	Alt+Left Arrow, 'A'
Jump forward to the next cut	Alt+RightArrow, 'S'
Jump backward to the next label	Ctrl+Left Arrow
Jump selecting forward label	Ctrl+Shift+Right Arrow
Jump selecting backward label	Ctrl+Shift+Left Arrow
Jump forward to the next label	Ctrl+Right Arrow
Jump backward... next Keyframes	Shift+Left Arrow
Jump forward... next Keyframes	Shift+Right Arrow
Jump backward to the next Auto	Ctrl+Shift+Alt+Left Arrow (only works on UbuntuStudio)
Jump forward to the next Auto	Ctrl+Shift+Alt+Right Arrow (only works on UbuntuStudio)

CHAPTER 16

Configuration, Settings and Preferences

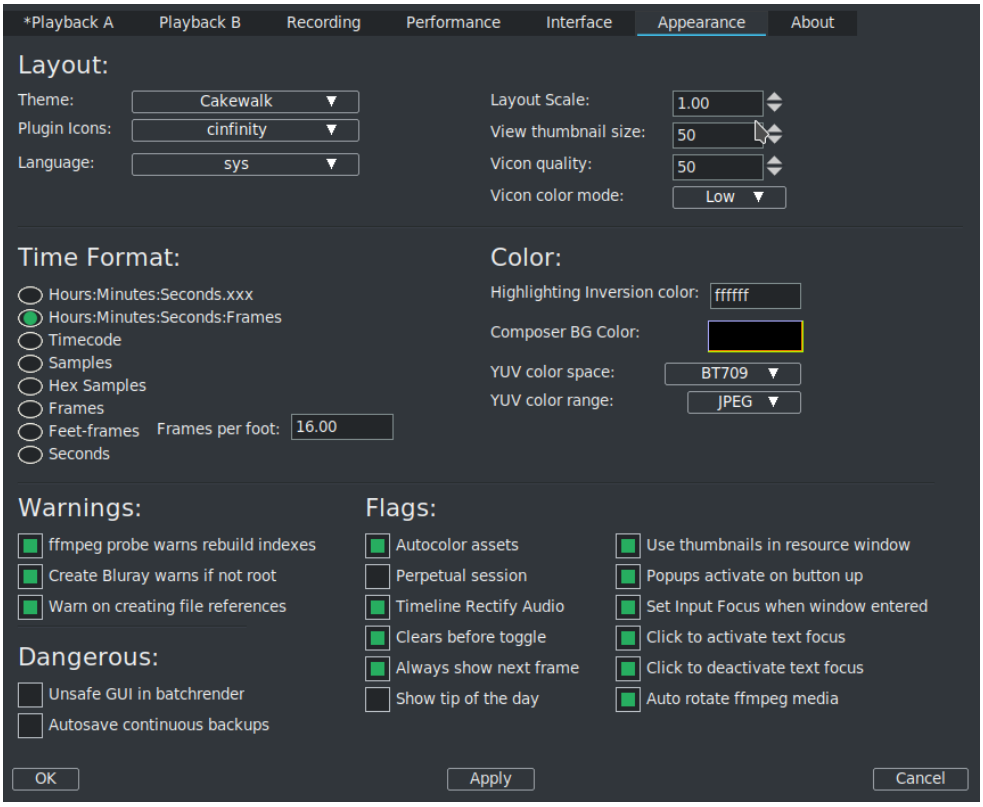


Figure 16.1: *Settings of default Preferences with the Appearance tab selected*

16.1 .bcast5

The user's default settings, preferences, and other helpful files are retained across sessions in a hidden file, called `.bcast5`, in the user's `$HOME` directory. Initially when CINELERRA-GG is launched there is an empty project and there are program default settings, and from then on the `.bcast5` directory will contain the settings that were set when quitting. If you need to revert to the default settings, delete the `.bcast5` directory contents and restart CINELERRA-GG. Or you may want to rename it temporarily if you

think you might want it back later. Although the location defaults to `$HOME/.bcast5`, you can use the `CIN_CONFIG` variable to override this location. For example: `export CIN_CONFIG=/tmp/.bcast5` will use a temporary setup for testing purposes. It is also useful for multiple users sharing the same home directory who would like to have different configuration/preferences settings data. And if you are experiencing inexplicable errors or crashes in CINELERRA-GG, they may be due to a problem with `.bcast5` in which case taking it out of the picture can at least eliminate this as the cause. Another use case is if we want multiple installations of CINELERRA-GG, for example one stable and one experimental, we can create the `.bcastX` folder externally and then set it as default with the option *Index files go here* in Index Files section.

Several ways exist to change CINELERRA-GG's operational characteristics. A lot of variations can be made to settings and preferences by using the *Settings* pulldown from the main window and choosing *Preferences*.

In addition, there are currently 2 settings that can only be manually changed in your $Home/.bcast5/Cinelerra_rc file and then done so by very careful editing. These include FAST_SPEED and SLOW_SPEED which allow you to modify the default 2X speed of *Fast forwward* and the .5X speed of the #2 keypad for *Slow speed forward play*. For example, adding the 2 lines FAST_SPEED 4 and SLOW_SPEED .2 to the end of the Cinelerra_rc wile will result in playing at 4X faster and 5X slower.

Another example of user modifiable settings involving the plugins menu in the Resources window, can be found in Expanders for Plugin Subtrees in the Resources Window.

16.2 Playback A / Playback B

CINELERRA-GG supports 2 separate preferences for the playback configuration. CINELERRA-GG can be operated in a single or dual screen configuration, both by using Xinerama or dual screen configuration of X windows. It will take some setup using Xconfig to make this work.

The `Settings → Preferences` menu has *Playback A and Playback B* tabs. The target display and audio device configuration can be separate, to support up to 2 display and/or audio device stations. The active configuration displays an asterisk (*) in its selection tab and the selected tab will be made active when OK is pressed. For example: you may have a dual screen monitor system with the left screen showing the CINELERRA-GG main window and the right screen showing the composer. Another setup might use a monitor for the left screen and an HDTV as the right screen displaying the composer. When a playback configuration is selected, the audio/video device configuration is switched to the playback selection. The active playback setup can be changed through use of the menu pulldown of `Settings → Preferences` or via the remote control menu selection (see the section Remote Control for DVB for more detail).

Yes, you can watch TV on CINELERRA-GG instead of CINELERRA-GG on TV.

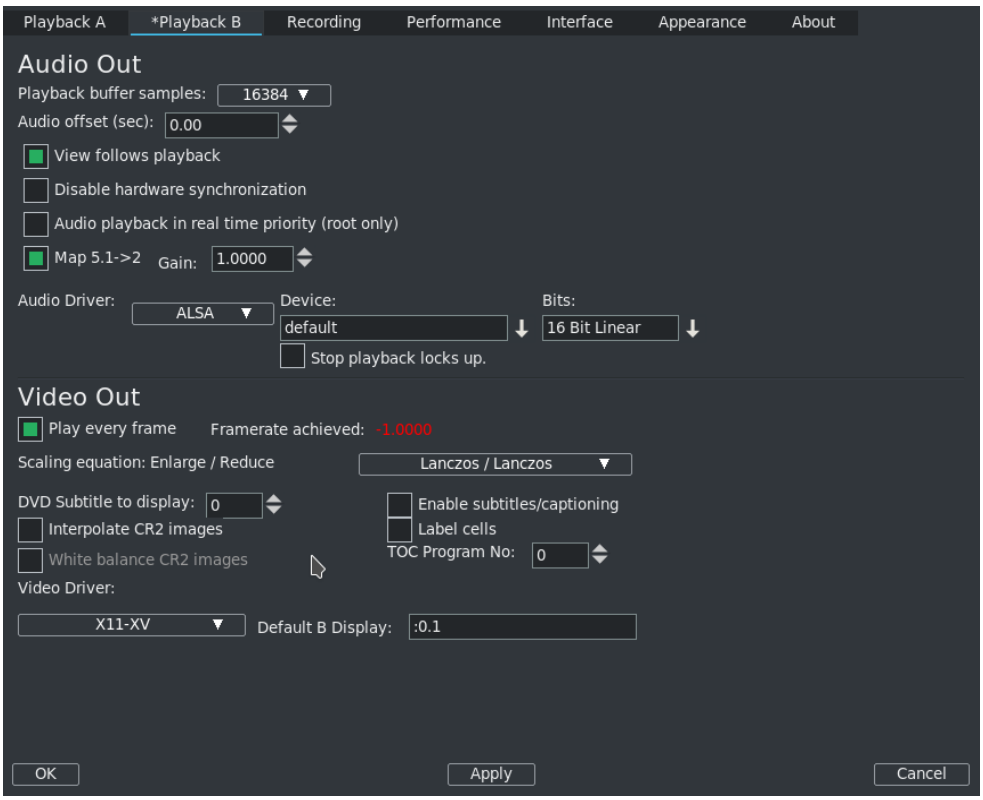


Figure 16.2: *Multi-screen Playback example useful for watching* CINELERRA-GG *run on the big screen*

Figure 16.2 shows partial window of **Playback A* selected and the second tab for *Playback B*. Note that on the bottom right of the window, *Default B Display:* is set to : 0.1, representing the setting for Screen 1. On the unseen *Playback A* window, the *Default A Display:* will be set to : 0.0 meaning for Screen 0. Otherwise, the default would be nothing there or just <empty>.

16.2.1 Audio Out section

The audio drivers are used for both recording and playback. The Audio Out settings affect the outcome when you play sound on the timeline.

Playback buffer samples for playing audio, small fragments of sound are read from disk and processed sequentially. A larger value here causes more latency when you change mixing parameters but yields more reliable playback. Some sound drivers do not allow changing of the fragment, so latency is unchanged no matter what the value. Since different stages of the rendering pipeline can change the rate of the incoming data, it would be difficult to disconnect the size of the console fragments from the size of the fragments read from disk.

Audio offset (sec) the ability to tell the exact playback position on Linux sound drivers is poor. The audio offset allows users to adjust the position returned by the sound driver in order to reflect reality. The audio offset does not affect the audio playback or rendering at all. It merely changes the synchronization of video playback. The easiest way to set the audio offset is to create a timeline with one video track and one audio track. Expand the audio track and center the audio pan. The frame rate should be larger than $24fps$ and the sampling rate should be greater than 32000. The frame size should be small enough for your computer to render it at the full framerate. Highlight a region of the timeline starting at 10 seconds and ending at 20 seconds. Drop a gradient effect on the video track and configure it to be clearly visible. Drop a synthesizer effect on the audio and configure it to be clearly audible. Play the timeline from 0 and watch to see if the gradient effect starts exactly when the audio starts. If it does not, expand the audio track and adjust the nudge. If the audio starts ahead of the video, decrease the nudge value. If the audio starts after the video, increase the nudge value. Once the tracks play back synchronized, copy the nudge value to the audio offset value in preferences. Note: if you change sound drivers or Disable hardware synchronization, you will need to change the audio offset because different sound drivers are unequally inaccurate.

View follows playback this causes the timeline window to scroll when the playback cursor moves. This can slow down the X Server or cause the timeline window to lock up for long periods of time while drawing the assets.

Disable hardware synchronization most sound cards and sound drivers do not give reliable information on the number of samples the card has played. You need this information for synchronization when playing back video. This option causes the sound driver to be ignored and a software timer to be used for synchronization.

Audio playback in realtime priority (root only) for really old computers, this setting allows uninterrupted playback during periods of heavy load. It forces the audio playback to the highest priority in the kernel. Today, it is most useful for achieving very low latency between console tweaks and sound card output. You must be root to get real-time priority. Only experts might want to use this because it interferes with ordinary time-share scheduling and can lock up the system. When this is enabled, audio gets the first shot and burns audio until audio lets go. To explain, there are 2 kinds of scheduling, *time-sharing* which is the default, and *real time* where the scheduled task must explicitly request scheduling to allow other tasks to execute. Time-share interrupts when you use up your allocated time slice. Realtime priority audio will execute audio decode until it finishes, which may slow down other types of processing like video decoding. Most decoders use a policy that video may be downsampled to accommodate scheduling, but will never skip audio because it creates a much more obvious defect. This feature helps to make sure audio gets priority over video during decode. Be sure to check apply in order for this feature to take effect.

Map 5.1→2 playback 5.1 → 2 driver downmix maps 6 tracks to 2 channels when checked, that is mixes 5.1 down to stereo on the output device side. This is different from the patchbay and menubar functions which reset the pan/mix levels of the input

channels. In this way, you can render 5.1 media, and use stereo speakers to listen in the same session setup. This downmix only occurs if the playback is 5.1 (6 channels) and the device config is stereo (2 channels).

Gain set audio gain to a different value than the default of 1.0 This feature, device level gain, corrects for hardware conditions which some devices may need to be useful. For example, you may need to increase the gain for a weak microphone or a noisy speaker, since it affects rendering when you crank up or down the audio via use of the patchbay. With the audio H/W gain support, you have the ability to fine tune the audio volume by some numerical value for the scale. You are adjusting the scaling of data into the audio driver – H/W scaling is done before it goes into or out of the driver. This is a one time linear multiplication of the sample values, and may offer better control than the logarithmic DB gain controls of the application.

Audio driver there are many sound drivers for Linux. This allows selecting one sound driver and setting parameters specific to it. The currently available possibilities are listed next.

ALSA is the most common sound driver these days and supports almost all sound cards. ALSA is frequently updated but is very stable.

OSS was one of the first Linux sound drivers and has an open source implementation with many sound cards supported.

OSS Envy24 is the commercial version of OSS with a variant for $24bit96KHz$ sound cards. This variant required changes to the way the sound drivers were used and so needed a different driver.

Raw 1394, DV 1394, IEC 61883 are older audio drivers used by camcorders and not much else.

Pulseaudio Extends the functionality of ALSA. It is a more modern and highly supported driver.

Device with the down arrow, you can see the device choices on your computer.

Bits 8, 16 or 24 Bit Linear are the current choices for the number of bits of precision CINELERRA-GG should set the device for. The meaning of the number of bits can be misleading. Some sound drivers need to be set to 32 bits to perform 24 bit playback and will not play anything when set to 24 bits. Other sound drivers need to be set to 24 bits for 24 bit playback.

Stop playback locks up this ALSA only checkbox is needed if stopping playback causes the software to lock up. This has worked some time ago, but may no longer work as expected

16.2.2 Video Out section

The video drivers are used for video playback in the compositor and the viewer. These determine how you will see video on the timeline.

Play every frame this causes every frame of video to be displayed even if it means that the playback of the video tracks fall behind. Most likely you will want this enabled because, after all, in order to edit you want to see each frame. However, if you are just watching a big video, you can switch to not play every frame so that you can at least not be distracted by slowness.

Framerate achieved the number of frames per second being displayed during playback. This is updated during playback only. The goal is to get as close to the frame rate as possible, even if Play every frame is not enabled.

Scaling equation Enlarge / Reduce – this algorithm is used when video playback involves scaling or translation (only X11 video driver). This does not affect 1 : 1 playback. Choices available are:

Nearest Neighbor / Nearest Neighbor low quality output with fast playback. Often produces jagged edges and uneven motion.

Bicubic / Bicubic Bicubic interpolation is used for both enlarging and reducing, enlarging blurs slightly but does not show stair step artifacts.

Bicubic / Bilinear High quality output with slow playback. Bicubic interpolation is used for enlarging, which blurs slightly but does not show stair step artifacts. A bilinear interpolation is used for reduction, which produces very sharp images and reduces noise. Bilinear reduced images can be sharpened with a sharpen effect with less noise side effects than a normal sized image.

Bilinear / Bilinear when slight enlargement is needed, a bilinear enlargement looks better than a bicubic enlargement. Bilinear uses less CPU than either Bicubic or Lanczos.

Lanczos / Lanczos is not necessarily a general purpose upscaler, but is intended for low resolution sources. However many people like the sharpening effects. More quality from Lanczos does take more CPU.

DVD subtitle to display DVD IFO files usually contain subtitle tracks. These must be decoded with the MPEG decoder. Select Enable subtitles to enable subtitle decoding. There are usually multiple subtitle tracks indexed by number and starting from 0. Enter the index number of the subtitle track to be decoded in the *DVD Subtitle to display* text box or use the tumbler to increase the index value. Go to the asset corresponding to the MPEG file in the *Resources* window and right click. Click on *Info*. The number of subtitle tracks is shown at the bottom.

Enable subtitles/captioning for broadcast TV ??

Label cells ??

TOC Program No Table of Contents program number used in DVB ??

Interpolate CR2 images enables interpolation of CR2 images. Interpolation is required since the raw image in a CR2 file is a Bayer pattern. The interpolation uses dcraw's built-in interpolation and is very slow. This operation can be disabled and the Interpolate Pixels effect used instead for faster previewing

White balance CR2 images this enables white balancing for CR2 images if interpolation is also enabled. This is because proper white balancing needs a blending of all 3 primary colors. White balance uses the camera's matrix which is contained in the CR2 file. Disabling white balancing is useful for operations involving dark frame subtraction. The dark frame and the long exposure need to have the same color matrix. If you disable Interpolate CR2 Images and use the Interpolate Pixels effect, be aware the Interpolate Pixels effect always does both interpolation and white balancing using the camera's matrix, regardless of the settings in Preferences. Dark frame subtraction needs to be performed before Interpolate Pixels.

Video driver normally video on the timeline goes to the compositor window during both continuous playback and when the insertion point is repositioned. Instead of sending video to the Compositor window, the video driver can be set to send video to another output device during continuous playback. However, this does not affect where video is routed when the insertion point is repositioned. Options are listed next.

X11 this was the first method of graphical display on Unix systems. It just writes the RGB triplet for each pixel directly to the window. It is useful when graphics hardware can not handle very large frames. And when X11 is used with the associated checkbox enabled of *use direct x11 render if possible* it can be a really good playback method to speed up playback for large frames.

X11-XV this was an enhancement to X11 in 1999. It converts YUV to RGB in hardware with scaling. In some cases it may be the preferred playback method, but it can not handle large frame sizes. Maximum video size for XV is usually 1920 × 1080.

X11-OpenGL the most powerful video playback method is OpenGL. With this driver, most effects are done in hardware with the graphics board installed in the computer. OpenGL allows video sizes up to the maximum texture size, which is usually larger than what XV supports, depending on the graphics driver. OpenGL relies on PBuffers and shaders to do video rendering. Plugins or transitions that do not have *handle OpenGL* in the code will use software instead of hardware and this will slow down playback. OpenGL does not affect rendering. It just accelerates playback. X11-OpenGL processes everything in 8 bit color models, although the difference between YUV and RGB is retained. The scaling equation set in the preferences window is ignored by OpenGL – it always uses linear scaling. Camera and projector operations use OpenGL, but some of the effects may not support OpenGL acceleration.

Raw 1394, DV 1394, and IEC 61883 are for old camcorders.

Default A/B Display the interface is intended for dual monitor displays. Depending on the value of Display, the Compositor window will appear on a different monitor from the rest of the windows.

16.3 Recording

The parameters here expedite the `File` → `Record...` function by allowing the user to pre-configure the file format and the hardware used for recording, since the hardware generally determines the supported file format. Once set, the file format is applied to all recordings.

16.3.1 File Format section

File Format this determines the output file format for recordings. It depends heavily on the type of driver used. The menu selections are the same as those of the rendering interface.

Record audio tracks toggle must be enabled to record audio.

Record video tracks toggle must be enabled to record video. The wrench button left of both the audio and video tracks toggle, opens a configuration dialog in order to set the compression scheme (codec) for each audio and video output stream. The audio and video is wrapped in a container format defined by the *File Format* menu. Different wrappers may record audio only, video only, or both. Some video drivers can only record to a certain container. If the video driver is changed, the file format may be updated to give the supported output. If you change the file format to an unsupported format, it may not work with the video driver.

Realtime TOC setup for DVB recording to automatically generate a Table of Contents. This will scan the stream data *on the fly* on its way to being written while the asset is being captured. ??

16.3.2 Audio In section

Record Driver is used for recording audio in the Record window. It may be configured the same as the Record Driver for video if the audio and video are wrapped in the same stream. Available parameters vary depending on the driver. The drivers are the same as described in Playback A/B with the addition of DVB and V4L2 MPEG but no Raw 1394.

DVB Adapter name of a suitable DVB adapter for linux that is usb connected to your computer and has a connected broadcast TV antenna.??

dev your DVB adapter device number, which is usually 0.

Bits same as described in Playback A/B audio section.

Follow audio config ??

Samples read from device a good value is 2048 or approximate dev buffer size ($2k-16k$ probably). Samples to write to disk – at a time. First, audio is read in small fragments from the device. Then, many small fragments are combined into a large fragment before writing to disk. The disk writing process is done in a different thread.

The value here determines how large the combination of fragments is for each disk write. A good starting value is 48000 but this will will most likely automatically change, probably to 44100 if necessary.

Sample rate for recording regardless of what the project settings are, the value set here will be the sample rate used for recording. The sample rate should be set to the highest value the audio device supports.

Channels to record usually set to 2.

Map 5.1→2 eave unchecked to record all possible channels.

Gain usually leave at default of 1.0, but this device level gain corrects for hardware conditions on some devices which need help. This gives you the ability to fine tune the audio volume by some numerical value for the scale. It is useful as better explained for the Gain in the Playback A/B Audio section discussed previously.

Record in realtime priority (root only) – only experts might want to use this because it interferes with ordinary time-share scheduling and can lock up the system. When this is enabled, audio gets the first shot and burns audio until audio lets go.

16.3.3 Video In section

Record driver used for recording video in the Record window. It may be configured the same as the Record Driver for video if the audio and video are wrapped in the same container. Available parameters vary depending on the driver. The drivers available are as follows.

- Video4Linux2
- JPEG webcam
- YUYV webcam
- Video4Linux2 JPEG
- Video4Linux2 MPEG
- Screencapture
- DV1394
- IEC61883
- DVB

DVB Adapter name of a suitable DVB adapter for linux that is usb connected to your computer and has a connected broadcast TV antenna.??

dev your DVB adapter device number, which is usually 0.

Follow video config ??

Frames to record to disk at a time frames are recorded in a pipeline. First, frames are buffered in the device. Then, they are read into a larger buffer for writing to disk. The disk writing is done in a separate thread from the device reading. For certain codecs the disk writing uses multiple processors. The value set here determines how many frames are written to disk at a time.

Frames to buffer in device the number of frames to store in the device before reading and this determines how much latency there can be in the system before frames are dropped.

Positioning if set to *Software timing*, use software for positioning information. Video uses audio for synchronization, but most sound cards do not give accurate position information so selecting this option makes CINELERRA-GG calculate an estimation of audio position in software instead of hardware for synchronization. You can also choose *Presentation Timestamps*, *Device Position*, or *Sample Position*.

Sync drives automatically for high bitrate recording, the disk drives you use may be fast enough to store the data but your operating system may wait several minutes and stall as it writes several minutes of data at a time. This forces the operating system to flush its buffers every second instead of every few minutes to produce slightly better real-time behavior.

Size of captured frame is the size of the recorded frames in pixels. It is independent of the project frame size because most video devices only record a fixed frame size.

Frame rate for recording the frame rate recorded is different from the project settings. This sets the recorded frame rate.

16.4 Performance

16.4.1 Performance section

The main focus of the performance section is rendering parameters not available in the rendering dialog with the obvious gain of perhaps better performance. For more details about using GPU hardware acceleration for both decoding and encoding see Hardware video acceleration.

Cache size to speed up rendering, several assets are kept open simultaneously. This determines how many are kept open. A number too large may exhaust your memory rapidly. A number too small may result in slow playback as assets need to be reopened more frequently.

Cache Transition transitions can be slow on timeline playback. With this parameter (active by default) a cache is used to smooth the transition. You might see a slight hesitation at the beginning of a transition when memory is being loaded then quite often it will play at full speed.

Seconds to preroll renders some effects need a certain amount of time to settle in. Checking this option sets a number of seconds to render without writing to disk before the selected region is rendered. When using the render farm, you will sometimes need to preroll to get seamless transitions between the jobs. Every job in a render farm is prerolled by this value. This does not affect background rendering because background rendering uses a different preroll value.

Force single processor use CINELERRA-GG tries to use all processors on the system by default, but sometimes you will only want to use one processor, like in a render farm client. This forces only one processor to be used. The operating system usually uses the second processor for disk access. The value of this parameter is used in render farm clients.

Project SMP cpus to restrict the number of processors utilized, change the count number. This number will be used for the plugin per load balance operation cpu limit, which uses smp-cpus to stripe your data. It does not affect the number of cpus used in any other CINELERRA-GG operation besides plugins. On large cpu systems, it can come in handy to downgrade the number of cpus used for some plugins; otherwise it uses all of the processors and splits up the program into too many pieces which may add considerable overhead in high cpu count systems.

Use HW Device CINELERRA-GG can use hardware timeline acceleration (decoding) via GPUs thanks to the *OpenGL* video driver. By default the *X11* video driver is used, which works only with the CPU. See Video Out section. For h264, h265 (HEVC) and VP9 codecs you can use libraries specific to AMD, Nvidia and Intel graphics cards. For AMD and Intel set **vaapi**; for Nvidia set **vdpau** (also works with AMD thanks to a wrapper). **Cuda** does not accelerate decoding with Nvidia, but it does allow some plugins to run that are not available otherwise. **None** (default) uses the video driver set in the `Playback A/B` tab.

16.4.2 Background Rendering section

Use background rendering checking this box, enables automatic background rendering. This works in conjunction with the interactive function `Settings menu` → `Toggle background rendering` which sets the point where background rendering starts up to the position of the insertion point.

Frames per background rendering job his only works if a render farm is being used; otherwise, background rendering creates a single job for the entire timeline. The number of frames specified here is scaled to the relative CPU speed of rendering nodes and used in a single render farm job.

Frames to preroll background the number of frames to render ahead of each background rendering job. Background rendering is degraded when preroll is used since the jobs are small. When using background rendering, this number is ideally 0. Some effects may require 3 frames of preroll.

Output for background rendering background rendering generates a sequence of image files in a certain directory. This parameter determines the filename prefix of the

image files. It should be on a disk, accessible to every node in the render farm by the same path.

File format the file format for background rendering has to be a sequence of images. The format of the image sequences determines the quality and speed of playback. JPEG is a good choice usually.

Video wrench this has the single option of *use alpha*. It is by default unchecked.

16.4.3 Render Farm section

In the Render Farm Section are many options that are explained in detail in the Rendering chapter of this manual. Just make sure if you do not intend to use a render farm, that *Use render farm* is not checked.

16.5 Interface

16.5.1 Editing section

Clicking on edit boundaries (trimming) CINELERRA-GG not only allows you to perform editing by dragging edit boundaries, but also defines five separate operations that occur when you drag an edit boundary. Here you can select the behavior of each mouse button. The usage of each editing mode is described in great detail in the Advanced Editing chapter.

Keyframe reticle the options are *Newer*, *Dragging*, or *Always*. This is used to help in checking edit alignment across tracks. Always renders a line over all plugins, and dragging only over the drag icon. Never draws nothing.

Snapshot path designates the default directory path for snapshot and grabshot generated output.

16.5.2 Operation section

Probe Order clicking on this box brings up a popup allowing you to change the probe order usually for media that is raw camera output but it is also helpful if you want to ensure that a specific driver is used for certain media; for example you may want *tiff* files to be read natively instead of by ffmpeg.

trap sigSEGV always enable this so that if CINELERRA-GG crashes, a dump will be generated for analysis.

trap sigINT always enable this so that you can use Ctrl-c to interrupt the program if it appears to be hanging. This will often generate some useful information for analysis.

Use yuv420p dvd interlace format for DVD media this option maintains the interlacing in Chroma sample addressing, which ordinarily would be deleted because the upsampling of interlaced chroma fields is normally done using a progressive algorithm.

With this mode enabled, the MPEG decoder uses a different algorithm for interlaced frames so that the 4:2:0 format chroma interlacing is preserved.

Min / Max DB for meter *Min DB* is useful because some sound sources have a lower noise threshold than others. Everything below the noise threshold is meaningless. This option sets the meters to clip below a certain level. *Max DB* sets the maximum sound level represented by the sound meters. This value is presented merely to show how far over the limit a sound wave is. No matter what this value is, no sound card can play sound over 0 dB.

Import images with a duration of # seconds when you load single images, like *png* or *jpeg*, automatically load for # number of seconds. This makes it easier to see an image on the timeline. If you just want the single frames, uncheck this option.

Auto start lv2 gui some lv2 plugins display a *glitzy* UI (User Interface); for example the Calf plugins. For these LV2 plugins, if you want that to automatically come up without having to click on the UI button on the simplified ui interface, this is the flag to enable that.

Android Remote Control check this to enable using an android device as a remote control for broradcast TV.

Port default port 23432 is used for the android remote control.

Pin default PIN Cinelerra is used for the android remote control.

Shell Commands this button brings up the controls for setting up your own shell commands or editing previously set up commands. See the section on Menu Bar Shell Commands for information.

Reload plugin index execute this reload command when you have modified plugins and want to make sure your changes take effect.

Nested Proxy Path designates the default directory path for Nested Proxy files.

Default LV2_Path when there is no system `LV2_PATH` set, if you want lv2 plugins loaded, you must set the correct directory path name here. When you change this field, cin will automatically restart and load the newly specified lv2 plugins.

16.5.3 Index Files section

Screencast below shows part of the Preferences menu where you can change the index files setup (figure 16.3).

Index files go here index files exist in order to speed up drawing the audio/video tracks. This option determines where index files are placed on the disk

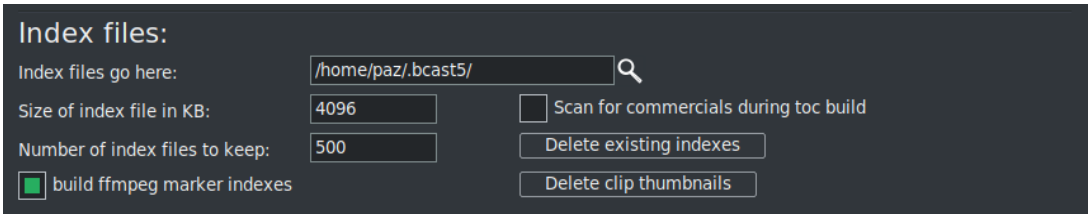


Figure 16.3: *Index file setup for your preferred configuration for Render Farm sharing or anything*

Size of index file determines the size of an index file. Larger index sizes allow smaller files to be drawn faster, while slowing down the drawing of large files. Smaller index sizes allow large files to be drawn faster, while slowing down small files. The default is currently 4kB for average size files.

Number of index files to keep to keep the index directory from becoming very large, old index files are deleted. This determines the maximum number of index files to keep in the directory.

build ffmpeg marker indexes improves ffmpeg seeks in certain cases although not clear which ones.

Scan for commercial during toc build used for working with broadcast TV commercial removal.

Delete existing indexes when you change the index size or you want to clean out excess index files, this deletes all the index files.

Delete clip thumbnails as clip thumbnails accumulate over time, you may want to delete them to get the disk space back.

16.6 Appearance

16.6.1 Layout section

Theme CINELERRA-GG supports 11 different themes to suit the preferences of different users (figure 16.4). When you change the theme, CINELERRA-GG automatically saves your session and restarts exactly where you were. The *Themes User Interface* are described in more detail next. *Akirad* themes are all available in Cin-GG as designed by the *Cinecutie* project. They provide alternative User Interface environments allowing you to choose your favorite look. Be aware that when you change your theme, CINELERRA-GG will automatically shutdown and restart for it to take effect. The theme you choose is preserved across sessions.

Theme choices available:

- Blond

- Blond-cv
- Blue
- Blue Dot
- Bright
- Cakewalk (default)
- Hulk
- Neophyte
- Pink Lady
- S.U.V.
- UnFlat

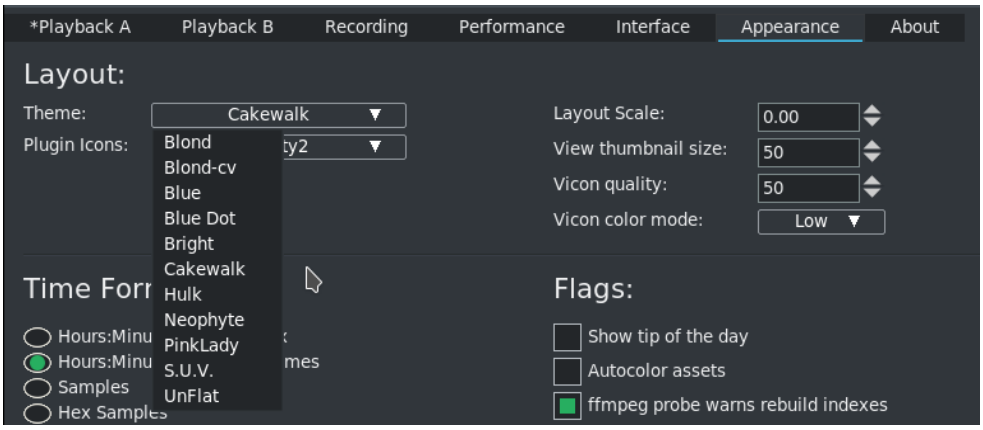


Figure 16.4: *Shows the Cakewalk theme (courtesy Olaf) on Preferences window with list of themes*

How To Change your Theme:

To change a *Theme* in the main window pulldown, select:

`Settings → Preferences;`

then in Preferences window, click on the *Appearance* tab. In the Editing section in the lower left hand corner, click on the *down arrow* next to Theme to see your choices. Click on your desired choice from the list given. Check OK, CINELERRA-GG will automatically shutdown and restart.

To create a personal theme see How to create your own theme.

Plugin Icons There are currently 4 choices for different plugin icons to include the old original.

Locale The default is *sys* so that the system language is active. With the pulldown menu we can choose among the other languages present in ... This language will be saved in your Configuration and used each time you start up CinGG. In order to change the environment

variable, LANGUAGE, the setting must be *sys* because that is the best way we could get it working.

Layout Scale Allows for setting up scaling for your 4K monitors or any monitor where you would like the text and icons to be just a little bigger or a lot bigger. This scale setting is automatically saved across sessions. When first using CINELERRA-GG, or if *Layout Scale* has never been set, the initial value is 0.0. This means an automatic probe of the biggest monitor in use will be used for the setting. The advantage of this is that "new users" with a 4K monitor will not immediately be discouraged with too small text/icons. Leaving it at 0 instead of 1 is what most people will do and is probably preferable so that if you move to a different monitor with different dimensions/resolution, it will automatically probe. If a user wants to prevent the automatic scaling, *Layout Scale* should be set to 1.0 to avoid the smaller characters that might result due to the probe of a non-1080p monitor.

For testing or when you are using a different sized monitor and want to ensure the expected size for larger text/fonts before you start the application from a window, you can keyin:

```
BC_SCALE=2.0 {your Cinelerra path}/bin/cin
```

The scaling size would only be in effect for that run of CINELERRA-GG. This is particularly useful in the case where the user makes a mistake in setting the *Layout Scale* and CINELERRA-GG becomes unusable. Then the environment variable, BC_SCALE, can be used to overcome the bad setting so that you can get back into CINELERRA-GG and fix the scaling parameter. For example, if you accidentally set *Layout Scale* to 112.6, keyin the following and then when you get back into CINELERRA-GG, fix *Layout Scale* value in Preferences.

```
BC_SCALE=1.0 {your Cinelerra path}/bin/cin
```

View thumbnail size You can increase or decrease the thumbnail size – larger size uses more cpu.

Vicon quality Increase the quality used for thumbnails to get more clarity of pixels – this will use more memory.

Vicon color mode Modify the color mode to Low, Medium, or High for the thumbnails – High will look the best but takes more memory.

16.6.2 Time Format section

Various representations of time are given so that you can select the most convenient one for you. The time representation can also be changed by Ctrl-clicking on the timebar in the main window.

- Hours : Minutes : Seconds : xxx
- Hours : Minutes : Seconds : frames
- Timecode
- Samples
- Hex Samples
- Frames
- Feet-frames [frames per foot . . .]
- Seconds

16.6.3 Color section

Highlighting Inversion color Modify the selection area color; default is *ffffff* which is white. When you make a selection, that area becomes an inverse image which by default becomes a whitish color. You can set it to a different color by modifying the hex value in the box next to *Highlight inversion color*. Keep in mind that if you set the value to a low value, you will not be able to see the outlined selected area (for example the hex value "f" is not readily visible and leads to confusion). A leading 0 or blank is not allowed and will be automatically changed to *ffffff*.

Composer BG Color You can choose the background color of the viewer window.

YUV color space Default is *BT601*; others *BT709* (high definition), *BT2020* (ultra high definition). PAL and NTSC have different color primaries, so $BT601_NTSC$ = SMPTE 170M = BT 601_525 and $BT601_PAL$ = BT 470 BG = BT 601_625. BT 2020 can be $BT2020_NCL$ (non-constant luminance) or $BT2020_CL$ (constant luminance). NCL is the standard.

YUV color range JPEG $[0-255]$ and MPEG $[16-235]$

16.6.4 Warnings section

ffmpeg probe warns rebuild indexes this warning is very important for switching from using ffmpeg to using native formats, such as in the case of MPEG, so that you are reminded to *rebuild indexes*. If you do not rebuild the indexes, seeking on the timeline back and forth could very well be problematic, meaning it might not go to the right place. Notification about rebuilding the indexes will appear by default as shown in the figure 16.5 when you click on the FF icon in the main timeline in the upper right hand corner. Once you click on *Don't show this warning again* you will no longer be warned and this flag will no longer be enabled.

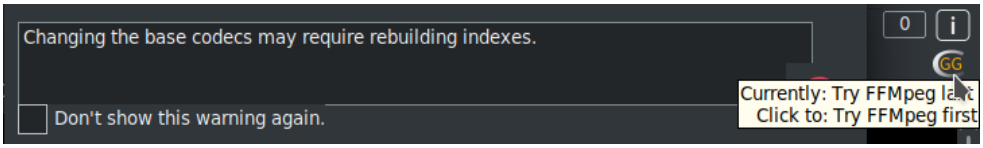


Figure 16.5: *Default warning when you click on FF icon in main window*

EDL version warns if mismatched in the case of a Batch Render, it is often helpful to be warned if the EDL has been changed so that you are aware that what is going to be rendered is different than your current EDL session.

Create Bluray warns if not root if checked and you are not logged in as root, you will get an error message in order to avoid doing a lot of work and then failing out because root is required for automount and to write on DVD hardware.

Warn on creating file references if checked, you will always be warned when using "File by Reference", that is when an EDL is opened as *Reference*. This is best left checked to ensure that you are aware of the fact that when changes are made to this file and rendered, any other uses of the same file will be affected and modified also.

16.6.5 Dangerous section

Unsafe GUI in batchrender It serves to hide the button *Save to EDL Path* present in the Batch Render window. In fact if used accidentally, it could lead to the loss of the original EDL content, which is overwritten without changing the project name. See Batch Rendering, section Advanced features.

Autosave continuous backups For each editing action CINELERRA-GG creates an automatic backup. To avoid excessive creation, this option limits their number to 50 when quitting out of the program.

16.6.6 Flags section

This section contains many useful options to cater to the various preferences of individual users.

Autocolor assets to make it visually easier to see your clips on the timeline that are from the same media file, you can have them automatically colored. Use of this feature requires additional memory and cpu on every timeline redraw, therefore smaller computers may not want this checked on.

Perpetual session is very useful for working on a project over many days so you can just quit before shutting down and the next time you start up CINELERRA-GG you will be right back where you left off. You will retain all of your undo's and redo's.

Timeline Rectify Audio for displaying rectified audio on the timeline instead of a standard audio waveform, check this flag. The waveform is cut on the zero line, thus

making the silent areas more visible and the waveform is stretched more over the entire height of the audio track, which improves the visibility of certain areas. This only affects the timeline and not any other audio waveform displays. Once you check or uncheck this flag, you must reload any media on the timeline in order to see the changed waveform.

Clears before toggle when using copy/paste in drag and drop mode some users prefer to resort to the addition of the Ctrl key for adding multiple selections. By checking this flag, the user retains usage as is commonly done for listbox operations.

Always show next frame in this mode the insertion pointer reflects the same as the Compositor so that for playing forward, the result is what looks like 1 was added to the frame displayed in the Compositor window. This is fully explained in another section (Playing/Seeking).

Show tip of the day if checked, a tip will be displayed in a popup box when start up CINELERRA-GG.

Use thumbnails in resource window the Resource Window displays thumbnails of assets by default, but drawing asset thumbnails can take more time and CPU as well as other system resources so you may want to uncheck this.

Popups activate on button up this is the default but if unchecked, popups activate on button down.

Set Input Focus when window entered this is checked on by default because on some operating system distros, when you move your mouse to a different window, nothing happens and you are left wondering why you can not enter information. When checked this causes the input focus to shift to any CINELERRA-GG window when the cursor enters an exposed region of the window which eliminates the need to switch input focus by tabbing.

Click to activate text focus Click to activate text focus

Click to deactivate text focus if checked, you will have to click to deactivate text focus.

Auto rotate ffmpeg media this is the default setting so that your media will automatically be rotated when there is metadata in the file with a rotation value. Especially useful for cell phone.

16.7 About

This section gives you information about the CINELERRA-GG program and version you are running. The original author's copyright and name are first and foremost. Next is a textbox with additional information and a summary of the monthly new features of note. Below that is a summary of the GPL License and the fact that it is provided without any warranty. Then the licensing verbage is the item that you may need to refer to most often – the *built* date and time in case you need to know which version you are currently running.

16.8 Environment Variables for Customization

Environment variables are global variables in the shell which all applications can read. They are set with a command like `set VARIABLE=value` or `export VARIABLE=value`. Environment variables can be viewed with a command like `env`. The values set can be removed with `unset VARIABLE`.

The following exported variables can be set to customize your environment. The `CIN_CONFIG` variable could be extremely useful for testing purposes or for multiple users sharing the same home directory who would like different default preference settings.

CIN_BROWSER name of browser to use by *Shell Cmds* options.

CIN_CC currently used by *Blend Algebra/BlendProgram* plugins' C compiler executable-with default of gcc; may be used in the future for other purposes.

CIN_CONFIG configuration data; defaults to `$HOME/.bcast5`.

CIN_DAT location of data files, such as documentation, models, tip of the day; also for *Blend Algebra/Blend Program* plugins' system library with scripts, headers, and functions that are distributed with Cinelerra reside in subdirectory `$CIN_DAT/dlfcn`.

CIN_EDITOR *BlendAlgebra/BlendProgram* plugins' C text editor with default of emacs; may be used in the future for other purposes.

CIN_LADSPA LADSPA directory path; use colons to separate multiple paths; this is convenient to define an alternate directory if you share the same executable directory among computers via NFS.

CIN_LIB location of library programs, such as bdwrite.

CIN_LOCALE locale text domain path to use for translating text.

CIN_PLUGIN plugin directory path.

CIN_RENDER complete filename with path, that was used for *select file to render* to in the current session's last successful Render job; this is used in the `RenderMux` defined *Shell Cmds* and is available for any user-defined script inside CINELERRA-GG.

CIN_PKG used to set your text domain, that is the locale path pointing to the cin.mo language file.

CIN_USERLIB for use by *BlendAlgebra/BlendProgram* plugins for user's library, default is `$HOME/.bcast5lib`). User's functions reside in subdirectory `$CIN_USERLIB/dlfcn`; may be used in the future for other purposes.

CIN_XSYNCH (for Developers only) set to 1 helps debugging for windows primitives to execute immediately and not be buffered up so you can see what is happening.

One example:

```
export CIN_BROWSER=chrome #would override default firefox for Shell Cmds.
```

Another example:

```
export CIN_CONFIG=/tmp/.bcast5 #use a temporary setup for testing purposes.
```

BC_USE_COMMERCIALS=1 to activate the commercial database (db) methods.

BC_TRAP_LV2_SEGV to get a dump of the failure of an LV2 plugin for help in debugging.

BC_FONT_SCALE=1.2 for changing the default size of the characters to be twice as big.

BC_ICON_SCALE=1.1 for changing the default size of the icons to be bigger; can change # to any. You can increase the size of the characters in the fonts and icons on your CINELERRA-GG system. This will make it easier to read the characters if you have trouble seeing the default small letters, which have been auto-scaled based on the window geometry. The user-friendly font/icon scaling default is 1 but you can set it to any decimal value. To defeat default auto scaling and get any size characters/fonts, override the setting via the previous 2 listed shell environment variables. This is very sensitive, meaning that even a small increase in the numeric value can vary the size quite dramatically.

BC_FONT_PATH=<colon-separated-search-path-for-fonts> to add additional font sets for the *Title* plugin or to remove all fonts set to `:(colon)`. An example: `export BC_FONT_PATH=/usr/share/fonts`.

BC_FONT_DEBUG=1 debug for determining which font is causing problems. 0 for no debug.

BC_SCALE=1.2 for setting up scaling for your monitor where you would like the text and icons to be bigger where 1.2 is a # chosen accordingly; in this case the text and fonts will be 1.2 times the normal size of 1. Refer to the previous Appearance section, *Layout Scale* for more details.

LADSPA_PATH specify an alternate set of ladspa plugins or include the default with the use of a colon separated list of directories to search for LADSPA plugins. This is always used first and if it does not exist, then the value for `CIN_LADSPA` becomes `LADSPA_PATH`.

LV2_PATH specify a certain set of LV2 plugins to use. Separate multiple paths with colons.

LANG CINELERRA-GG can be localized to display menus and messages in many languages. Currently there are 16 languages in addition to English, of varying quality. Some information on these is in the Translations chapter. Language settings are normally

read from your Linux O/S *locale* settings. The CINELERRA-GG program checks environment variables to determine your settings to use as the language. First it checks the value of LANGUAGE and will use that if set. If that is not set, then it checks LC_ALL and finally if that is not set either, it will use the value of LANG. You can check the current settings from a window by looking at the output of the *printenv* command.

To run on a language different than the one selected on your system just change the `LANG` environment variable before starting up CINELERRA-GG like this:

`export LANG=`*2 letter language code_2 letter country code.UTF-8*

For example, from a window type:

```
export LANG=es_ES.UTF-8
```

then startup CINELERRA-GG and you will have translations in Spanish. If you started CINELERRA-GG from that same window using `{cinelerra_path}/bin/cin` you will see a message in that window something similar to:

lang changed from en_US.UTF-8 to es_ES-UTF-8

However, although generally LANGUAGE is not set, it is possible that another progam that was run may have set it so you may have to substitute LANGUAGE for the word LANG above.

Depending on your Operating System and Desktop application, there may be other ways to accomplish a language change. For example, on Fedora with the Gnome desktop, the simplest method of switching your language to German and immediately starting CINELERRA-GG is to use:

```
LANGUAGE=de {cinelerra_path}/bin/cin
```

SHUTTLE_CONFIG_FILE alternate shuttle configuration file.

Chapter 17

How some stuff works

This section describes in detail some areas of CINELERRA-GG to help explain how things work.

17.1 Copy/Paste and Highlight Usage

There are 3 types of copy/cut and paste methods which exist in X windows, and most modern programs use 2 of them. The 3 cases are:

cut_buffers 0-7 these are obsolete but they still work and are the simplest to use.

highlighting called primary selection; almost all clipboard programs only use this.

copy and paste called secondary (or clipboard) selection. Some more modern programs use Ctrl-C/Ctrl-X and Ctrl-V for this (some use other keys or qualifiers, like Shift).

How Copy/Paste works:

All of the methods use window *properties* to attach data, called a selection, to a source window. The program advertises the selection by using the X server. The window property used determines which selection type is set/advertised by the new selection.

When a paste is used in a target window, the target program requests the advertised selection data. This may access one of two buffers depending on which type of load/paste action is used. The user loads a *cut buffer* via Drag select or Ctrl-C/Ctrl-X, and pastes a *cut buffer* via middle mouse press or Ctrl-V.

CINELERRA-GG cut and paste:

1. Text cut and paste operations

To use a text selection, create a drag selection in textboxes by pressing and holding left mouse button with the pointer over the beginning of the text selection, then move the pointer over the desired selection to the selection end, then release the mouse button. This continuously reloads the *primary* clipboard buffer and highlights the text selection. It can then be pasted to most programs by pressing the middle mouse button with the

pointer over the text insertion point. Some examples of these programs are *xterm*, *gnome-terminal*, *cinelerra*, and *browser* input text boxes. After you create a text selection, if you then press Ctrl-C the selection will also be copied to the *secondary* (or *clipboard*) selection buffer. This second paste buffer can be used for a more lasting save effect, since it will not be lost until you again press Ctrl-C (copy). Using Ctrl-V (cut) will also copy the selection to the secondary clipboard buffer, and then delete the selection from the textbox. If you press Ctrl-V (or paste) in a target window, the secondary selection will be inserted at the target window cursor. If a text selection exists in the target window, it is replaced by the pasted text.

2. Media cut and paste operations

To create a media selection, highlight a region on the CINELERRA-GG media timeline, then use the main menubar or compositor/viewer edit panel to operate the clip cut, copy, or copy-keyframe menu buttons. This selection can then be pasted to a target selection on the timeline using the main menubar or compositor/viewer edit panel to operate the clip paste or paste-keyframe operation. Also, by using the resource window you can select the *Clips* folder and right mouse the resources list box, then use the *Paste Clip* menu item to paste the selection to a named clip. Additionally, these methods work between running instances of CINELERRA-GG, which means you can move media clips between the CINELERRA-GG program instances. The clip data is also copied to the secondary clipboard buffer. This makes it possible to examine the clip content directly if so desired.

3. The older cut_buffer method

- For text, if there is an active selection when a window closes, it uses `cut_buffer0`. Normally when a paste is performed, the target window *notifies* the selection owner to *send it now* when you do a paste, but if the window has closed there is no source window, so no pasting. Some programs, like CINELERRA-GG, use `cut_buffer0` as a fallback. This makes it possible to paste data from a closed window.
- To move media clip, data `cut_buffer2` is used because it does not require the selection owner interface, and works simply and reliably. This buffer is not normally in use by other programs.

Final note

When a text selection is set, the selected text is redrawn using selected-highlight color when the textbox loses focus. This convenience feature shows the active text selection as you move the pointer to the new target window. When a new selection is set anywhere else on your screen, the current text selection will be redrawn using the inactive-highlight color as the textbox loses selection ownership. In most CINELERRA-GG themes, the drag selection text-highlight color is BLUE (#0000*FF*), the selected-highlight color is SLBLUE (#6040*C*0) – really sort of purple, and the inactive-highlight color is MEGREY (*#AFAFAF*).

17.2 Playing is Different than Seeking/Positioning!

17.2.1 Playing/Seeking

Seeking targets and displays the next frame. The next frame is targeted because frame zero has no previous. When you seek, you reposition to just before the target frame, and since the play direction has not been established (there is no direction when seeking) it shows you the next frame. This produces the expected behavior when you seek to frame zero; you see the first frame. Seeking displays in the compositor what you are getting ready to work with/edit/etc; always showing the next frame in relation to the cursor. Technically, since seeking just resets the position, it would be correct not to update the compositor, but it is best to seek and show the next frame to confirm that it is the frame you expected to see.

Playing shows you what has just been played in the compositor window. It is not the same as seeking. When you use Keypad 1 to play frame forward, then the 1st frame that is played and shown in the compositor window is frame zero (which was already displayed). The position is incremented to 1. Press Keypad 1 yet again, and the next frame displayed is 1, and the new position is 2, and so on. According to the implemented strategy, the insertion point moves to track playback. When playback stops, the insertion point stays where playback stopped. Thus, with playback you change the position of the insertion point.

Simple explanation of what you will be seeing in the compositor when playing:

Play forward the frame to the right of the cursor in the timeline gets displayed.

Play backward the frame to the left of the cursor in the timeline gets displayed.

The reason behind this *play* methodology is that you want to know what you just played so that you know what matches what you just saw/heard in case that is the desired stuff. You don't want the compositor to show you what you have not yet played – you need to see this frame to analyze/check to see if it is what you want. This behavior applies to any playing operation, such as the *keypad* or *Frame forward / Frame reverse* buttons. You can still easily see the actual insertion point in the zoombar at the bottom of the timeline – seventh button over or 3rd button from the right side. Also note the following:

Blinking insertion point on the timeline seeking/positioning was the last operation.

Solid non-blinking insertion point on timeline playing was the last operation.

Example and explanation

1. open a small example of 10 numbered frames (or use the `Title` plugin to add a timestamp)

2. press "f" to *fit* the timeline
3. make sure `settings` → `align_cursor_on_frames` is set
4. seek to frame 4 by clicking on the timeline at position 4, compositor shows the 5th frame, since the media counts from 1 and the timeline counts from 0. This is correct behavior.
5. press KP1 to play next frame. According to playback strategy: *When play is forwards, the next unit is displayed, and the position is advanced one unit*. So the next frame is 4, (shows the 5^{th} frame) it is displayed. The position is advanced from 4 to 5. This is correct behavior.
6. press KP4 to play the previous frame. According to playback strategy: *When play is in reverse, the previous unit is displayed, and the position is reduced one unit*. So the previous frame is 4, (shows the 5^{th} frame) it is displayed. The position is reduced from 5 to 4. This is correct behavior.

If you watch the zoombar (bottom of main window) position, it shows the current position is just before the next frame to be displayed when going forwards, and just after the frame to be displayed when going backward.

To recap, position is usually set in the program as a location that is between a previous and next frame/sample unit such that the next unit equals the seek target. After position is reset using a *seek* operation, the next unit is displayed, which is the seek target. When *play is forward*, the next unit is shown, and the position is advanced one unit. When *play is in reverse*, the previous unit is shown, and the position is reduced one unit. At the beginning, there is no previous, and at the end, there is no next, but silence is rendered at the end.

17.2.2 Always Show Next Frame

Since some users prefer the insertion pointer to reflect the same as the Compositor a choice is available. For playing forward, there is a preference option which results in what looks like 1 was added to the frame displayed in the Compositor window. To enable this mode, check the box `Always show next frame`, and this will be saved to `.bcast5`. The option checkbox is in the `Settings` → `Preferences` → `Appearance` tab and when checked, any forward *plays* in the Compositor window show the same frame as you would with a seek. Reverse plays and plays using a selection or In/Out pointers (with Ctrl) work the same as without this preference set. But you will no longer see the odd behavior where if you frame advance forward and then frame advance backward, the displayed frame does not change – instead it will change and look more natural. A color indicator that shows in the main track canvas timeline and the compositor timeline reminds the user which mode is currently active. The cursor in the compositor turns *red* for default mode and *white* for *Always show next frame* mode. The top portion of the insertion cursor in the track canvas mirrors this, with red for default and white otherwise.

Figure 17.1 using the default *playing* method where the frame in the compositor is the one that was just played; in this case play was in the forward direction. Note that the insertion pointer in the main track canvas shows 03:16 but the compositor show 03:15 so you know what you last saw. Also, the cursor/cursor tops in both windows is red.

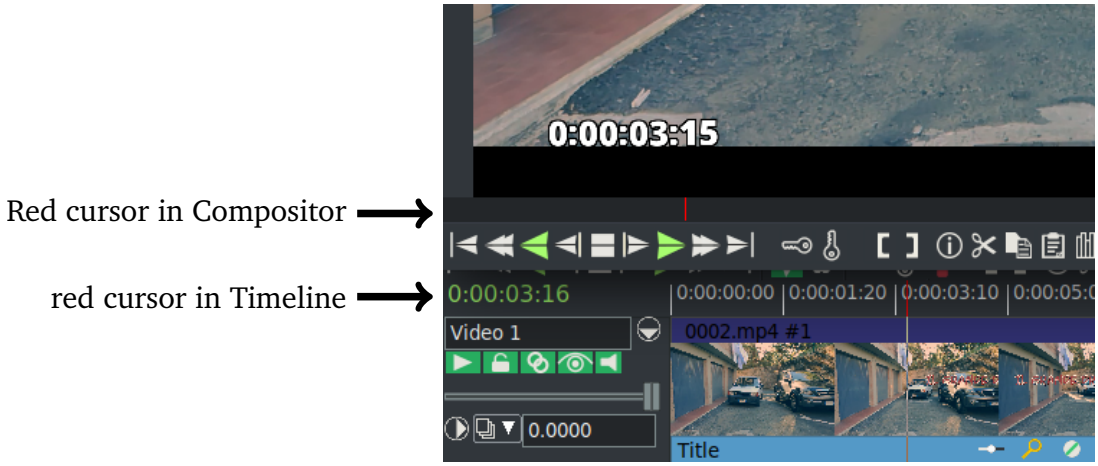


Figure 17.1: "*Default*" *mode with red cursors*

Figure 17.2 using the *Always show next frame* method where the frame in the compositor is the same one that would have shown with a seek; in this case play was in the forward direction. Note that the insertion pointer in the main track canvas shows 03:16 and the compositor shows 03:16. Also, the cursor/cursor tops in both windows is white.

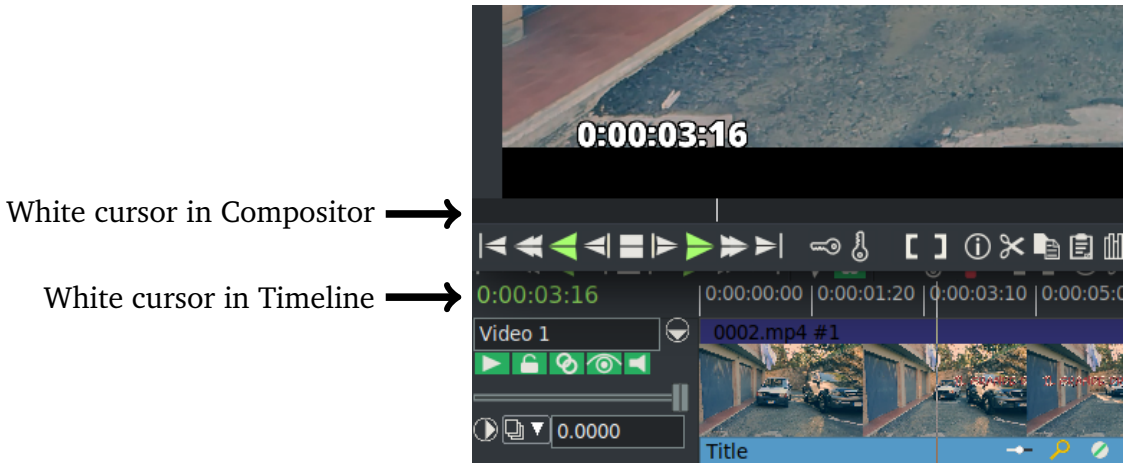


Figure 17.2: "*Always show next frame*" *mode with white cursors*

17.2.3 Seeking Issues

If you have an issue playing a video and not seeing it in the Compositor (just see a black screen), it is most likely due to the media not being designed to be *editable*. It is most likely not damaged. Generally it just does not have keyframes which are needed for seeking

which is what is done when you move around the media and start playing in the middle. The media plays just fine in the compositor if you always play from the beginning because then you don't need keyframes to seek. You can get around this problem if you proxy the media. A good choice to use for the proxy would be *use scalar*, *ffmpeg/mp4* and size of $\frac{1}{2}$. The proxied media can then seek and you will see it play in the compositor because keyframes exist.

17.3 Color Space and Color Range Affecting Playback

Playback *single step* and *plugins* cause the render to be in the session color model, while continuous playback with no plugins tries to use the file's best color model for the display (for speed). This can create a visible effect of a switch in color in the Compositor, usually shown as grayish versus over-bright.

The cause of the issue is that X11 is RGB only and it is used to draw the *refresh frame*. So single step is always drawn in RGB. To make a YUV frame into RGB, a color model transfer function is used. The math equations are based on color_space and color_range. In this case, color_range is the cause of the *gray* offset. The *YUV mpeg* color range is [16..235] for Y, [16..240] for UV, and the color range used by *YUV jpeg* is [0..255] for YUV.

The mpeg YUV color range [16..235] looks sort of like an old TV if it is viewed on a display with jpeg range [0..255]. A common expression for the short *mpeg* color range is *compressed* color range. If you are using color compressed data with no display decompression, or using uncompressed data and the display is configured to use compression, the color range will appear *squished* or *stretched*, as too gray or too much contrast.

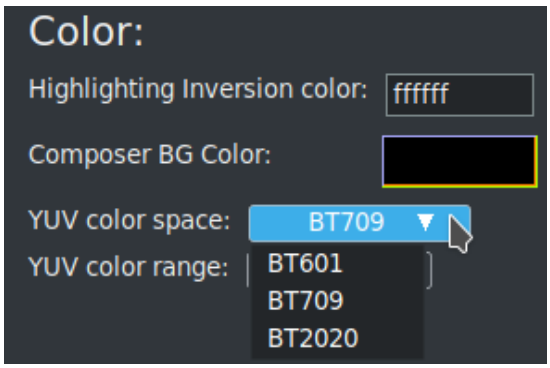


Figure 17.3: *Color space and Color range*

The *use X11 direct when possible* preference with X11 as your video driver, generally means that you value speed over color range. When you use this feature and the color range preference is mismatched, the color switch offsets will still appear. This is a personal choice solely for improved speed.

There is now program code to look for RGB versus YUV color model mismatches. You can override the default setup, which mirrors the original code, via the following.

`Settings` → `Preferences` → `Appearance` tab in the lower left hand corner (Figure 17.3):

YUV color space default choice is BT601, alternate is BT709 (High Definition), BT2020 (UHD)

YUV color range default choice is JPEG, alternate is MPEG

Some general tips (See also 3.2 *Color model* and Overview on Color Management):

- If your hardware allows it use RGB-Float (in `Settings` → `Format`); this format does not lead to transfer errors from one model to another, but it uses more cpu.
- Use RGB-8 if the source is RGB and YUV-8 if the source is YUV (most commonly used).
- If you notice alterations in color/brightness representation, try playing with color models in `Settings` → `Format` and with *YUV color space* and *YUV color range* in `Settings` → `Preferences` → `Appearance` tab. Another possibility is to check if the display color model conforms to the project color model. A practical case that may arise is as follows[1]: YUV source with limited color range (MPEG or TV); CINELERRA-GG set with the color range to extended (JPEG or PC); the colors on the compositor will be flattened. If we set the color range to MPEG the colors will be correct but hard clipping will occur. In this case the best result is presented by setting the color range to JPEG but then doing a conversion of the source to JPEG color range via the `ColorSpace` plugin. Summary table:

Source	YUV Color Range	Colors
MPEG	JPEG	Flat colors
MPEG	MPEG	Hard clipping
MPEG + conversion to JPEG	JPEG	Colors OK

17.4 Automatic "Best Model" Media Load

When you load media with the insertion strategy of *replace current project*, the program code will automatically use the "best model" for the render based on the media's codec. The best model is pretty much going to be what works well for television. This automation was added to facilitate easy use of CINELERRA-GG. Which is to say that it is difficult for a new or occasional user to set all of the necessary parameters as best as possible so the program does it for you. This means you do not have to *conform your project* which ordinarily would have to have been done in the Resources window with RMB click on the highlighted media and choosing *Match project size*.

However, this automatic method leads to the dilemma of where you have a 10-bit media file and it would get loaded as RGBA-8 when you would prefer it to be RGBA-Float.

[1] thanks to DeJay

So instead of using *replace current project* when loading your media, you would have to make sure the project is first set to your desired Format. This could be done with the `File→New project` and then setting your Color Model to RGBA-Float and whatever other parameters you want. Next when doing a `File→Load`, use *Append in new tracks* or *Create resources only*. This avoids using the "best model" technique and uses instead what you have designated so that if you set the Color Model to RGBA-Float that will be in effect.

It is important to note that even when using the "best model" no bits are lost if the input media is 10-bit and the Color Model is RGBA-8. This is because the media will be loaded using the "case BC_RGB16161616" where 16 stands for 16 bits. It fills the other 6 bits not used for 10 bits with zeros.

17.5 Simple Animation (Festival)

This functionality was added to CINELERRA-GG by the original author to create simple animation. The file type for this animation is *Scene*.

To get started making a simple animated movie copy from the directory:
`<cin_path>/cinelerra/tests` the `text2movie` and `text2movie.xml`.
You can see what this does via `File → Load...→ text2movie.xml`. The file text2movie acts like a normal asset, except changes to it are immediately reflected on the timeline without reloading and the length is infinite. You can just edit the text2movie file to change the script. If the length of the movie increases, drag the right edit handle to extend the edit or use the pulldown `Edit → edit length`. There is one audio channel created for every character. The frame rate, sample rate, frame size, and camera angles are fixed. To see these values, right click on the asset and look at the *Asset info*.

Currently the functionality that is implemented focuses on dialog between two people. The models are defined in model files saved in CINELERRA-GG's executable directory (for example, `/opt/cinelerra/models`). The character model and voice is selected separately in the script. The model files have the same name that appears in the script and are usually saved in the directory the script is in, but there is a defined search path, if not. You can create new models for the script without affecting the entire system. These models define the total size of the model along with the images used – the model images are 2D png images because all the animations are baked. Since there is no 3D renderer, no custom movement is supported.

There are currently 2 actions implemented:

1. Character2 can cut off character1 if character1's dialog ends in "..."
2. Inserting "[pause]" anywhere causes the character to pause. This is useful for adjusting the timing of dialog.

This is *simple* animation so you can expect speech synthesis not to be that good. And you will have to adjust punctuation and spelling based on the sound. Since the dialog is rendered on-demand, there is a delay when each character starts to speak but you can split dialog into shorter blocks to reduce the delay.

You can see a step by step example in the website http://www.g-raffa.eu/Cinelerra/HOWTO/animations.html

17.6 Textbox Non-std Character / Unicode Insertion

If you want to enter a special character – like a bullet, an accent grave character, or a mathematical summation symbol – you can use the unicode equivalent in a textbox to do so. In the textbox, keyin Ctrl-Shift-U which puts you into single character unicode mode, then keyin the numerical value for the intended single character followed by the carriage return. For a voluminous list of possible special characters, you can go to https://unicode-table.com/en/ on the internet to choose by highlighting a character to get its numerical equivalence. For example, $U + 2022$ is a bullet. If you make a mistake, you can use the *backspace* key or if you want to exit unicode-insert-mode, use the *ESC* key. This feature is especially useful with the *Title* plugin and for naming Tracks in the main window.

However, it is worth mentioning that some special characters are available via the *compose* key in the current distribution. https://en.wikipedia.org/wiki/Compose_key

Chapter 18

Troubleshooting and Help

To help new users, a pdf has been created comparing the editing workflow of Cinelerra-GG with the more usual editing workflow of Adobe Premiere Pro. See here.

18.1 Help and Context Help

Cinelerra-GG is a complex and feature-rich program. Using a guide is indispensable. The official manual (in English) can be found in PDF and HTML format:

https://cinelerra-gg.org/download/CinelerraGG_Manual.pdf

https://cinelerra-gg.org/download/CinelerraGG_Manual/

From within the program, you can invoke *Context Help*, which references sections of the HTML manual[1].

Context help can be requested from almost any Cinelerra-GG window or subwindow by pointing with the mouse cursor on the GUI element of interest and pressing `Alt/h`. That HTML manual page will be shown via the configured web browser, which is assumed as most relevant to the GUI element currently under the mouse pointer.

18.1.1 How Context Help works

The hotkey to request context help is `Alt/h`. What particular help page is shown, depends on the mouse cursor location while `Alt/h` is pressed. Usually, when the mouse is located in some window or dialog, the help page related to the functionality of this window or dialog is shown. In this case the mouse can point either on some widget, or on an empty area in the window. In Cinelerra-GG GUI there are several rich loaded windows, *Timeline* and *Compositor* for example. In such a case different help pages can be requested depending on the particular GUI element under the mouse. For example, pressing `Alt/h` while pointing on the *Autos curve* in a track would show help on *automation keyframes* section, pointing on the *Overlay* mode button in the *patchbay* would show help on *overlays*, pointing on the *camera control* in *Compositor* would show help on *camera and projector*.

[1] credit Georgy(sge) for full implementation

If no context dependent help page is found, the manual page of Contents itself is shown.

18.1.2 Requesting context help for plugins

There are several possibilities to request the help page for a particular plugin of interest.

1. Pressing `Alt/h` with the mouse in the dialog window of that plugin's settings menu.
2. Pressing `Alt/h` while pointing with the mouse on the plugin bar in a track, transition bar, or transition icon.
3. Pressing `Alt/h` while pointing on the plugin name (icon) in the Resources window. If plugin tooltips are on, help for the highlighted plugin under the mouse is shown. If plugin tooltips are off, help for the selected plugin is shown.
4. Pressing `Alt/h` while pointing on the plugin name in the *Attach Effect* dialog menu.

18.1.3 Requesting context help on Contour Shuttle

Contour Shuttle has no `Alt/h`. Nevertheless, its help page can be requested by simultaneously pressing the `Alt` key on the keyboard and any button on the Contour Shuttle. Note that the default Shuttle Configuration will be shown, rather than the one that you may have redefined.

18.1.4 Alternative web browser configuration

If you prefer to get *Context Help* pages displayed in the same tab in your browser instead of each help request displayed in a different tab, choose an alternative method as outlined here that works for you.

1. Use another browser which has such a configurable mode. Here is an example for Seamonkey:

   ```
   export CIN_BROWSER=seamonkey
   ```

   In the seamonkey browser go to Edit -> Preferences... -> Browser -> Link Behavior -> Links from other applications . Set the option "Open links passed from other applications in:" to the value "The current tab/window".

2. Hack a default browser if you know how to hack it. Here is an example for Firefox. Start Firefox and open the pseudo-URL:

   ```
   about:config
   ```

There will be a warning like "I'll be careful, I promise!", acknowledge it. Then there is a very long list with lots of undecipherable variable names. Scroll down to the variable: browser.link.open_newwindow.override.external . By default it has value of -1, which means "use value of the more general variable: browser.link.open_newwindow . Next, place the mouse cursor over: browser.link.open_newwindow.override.external , press the right mouse button, and select from the popup menu "Modify". You can now edit the value. Set it to 1, and you get new links from external apps opened in the same tab.

If you set the variable "browser.link.open_newwindow" instead, you get this behavior not only for external, but also for all links which otherwise would be opened in new tabs or new windows. The possible values of both variables are:

Value = 1: open in the current window and tab
Value = 2: open in the new window
Value = 3: (default): open in the new tab, current window

18.2 Troubleshooting

You can report potential problems, bugs, and crashes to the CINELERRA-GG website at:

https://www.cinelerra-gg.org

Here you can log the problem into the MantisBT bugtracker, or use the forum Q&A for help from other users, or email the problem to the Mailing List using the address:cin@lists.cinelerra-gg.org. It is usually more helpful if instead of starting CINELERRA-GG from its application icon, start from a window so that if there are error messages related to the problem, they can be captured from the screen and emailed or logged. The command to run from a window is: `<directory_path of where you installed cinelerra>/bin/cin` – for example if installed in `/mnt0/build5/cinelerra-5.1`, you would execute the following command to start the program:

```
/mnt0/build5/cinelerra-5.1/bin/cin
```

The problem you are experiencing may be as simple as an error due to the settings in your `$HOME/.bcast5` subdirectory so you may want to first rename your current `.bcast5` in order to start with default settings. By renaming the directory instead of deleting it, you will be able to put it back and not lose all of your preferences.

A note about using the Mailing List cin@lists.cinelerra-gg.org is that to avoid receiving spam, it will require you to subscribe. But it is also easy to unsubscribe once you have the information you need. The Mailing List, although routinely used by developers or contributors to discuss changes, can provide more technical answers in some cases and the email goes directly to individuals who are quite likely sitting at their computers depending on the timezone.

There are some easy things to do to fix errors that may have resulted from media problems, computer problems, or operational missteps so you can proceed without having to wait

for help. These are outlined in 18.5 - be sure to read down through *Common Problems* where some exact error messages are mentioned along with their cause or solution. Other troubleshooting help is included in other sections of this manual for specific features.

18.3 What to Include in Problem Reports

For the best help, if you have a reproducible problem and can provide the following list of materials for analysis, it is usually possible to figure out what the problem is and how to fix it. It may be a simple usage or setup mistake or a real bug which a programmer would like to fix rather quickly. Although it is not always necessary to provide this much information, it eliminates the *try this* or *did you do this?* and a lot of *guessing* rather than heading straight for a solution. Even if you can not provide all of this information, please report the problem in case other users are having the same issue.

Basically we **need to see what you see** with the input, output, and session file. Here is a list of the items that will most likely help to resolve the problem, but all of them will not necessarily be needed.

- Provide a detailed description of the problem and the last operation performed if you remember it.
- Supply a small representative sample of the original input that exhibits the issue.
- If possible, also provide the rendered output, again using that representative sample.
- Save a session file used with that same sample which will contain a lot of setup parameters; the best method to do this is to use the `File` → `Export Project...` with the Copy option. That way all of the files will be in 1 location and easily loaded onto any other computer.
- To make sure that the same rendering setup is used, it may be necessary to send an additional session file at the definition point just before rendering starts.
- Include the Operating System name and version number and version of CINELERRA-GG that you are running. You can find the date and time *built* in the `Settings` → `Preferences, About` tab, bottom left corner.

It is better to upload any files to a drop site as some of them can be quite large.

How to create a session file: from the menubar pulldown `File` → `Save_as...` and choose a filename, for example `/tmp/beforerender.xml` Do this after you have the situation setup so can see values in use. As just stated, *Export Project* option provides the most inclusive session information, but size could make it very large and cumbersome to upload or download.

How to create a representative small sample:

1. open the input media, set up the scenario as you normally would to start the project render;

2. select 5-15 seconds of media using the edit mode left mouse button drag highlighting on the timeline;
3. start the render dialog (Shift-R), set the filename path, and set *Render range* to *Selection*;
4. recommended *Insertion strategy* is *Replace current project*; then press the checkmark OK. The result will *load replace* the current project for you to review to ensure the error still occurs. If this small sample does not create the error, try a different section or upload the entire input media.

18.4 Crash Dumps for Analysis

If you get a SEGV crash and can explain what steps you took, a `/tmp/cinelerra_<pid> .dmp` file is very useful for analysis. You can also use Ctrl-c in the controlling window to force an INTR signal interrupt when you think the program is hung up. You can only perform one Ctrl-c as the second Ctrl-c quits out of the program. It is best to upload the `.dmp`, input files and other files to datafilehost or similar temporary site as they may be quite large. The `.dmp` filename looks like `/tmp/cinelerra_<pid> .dmp` Log a bug report or email all helpful information on the crash, location of uploaded files, and list of setup steps that illustrate the problem to cin@lists.cinelerra-gg.org

1. Use the latest version of CINELERRA-GG as you want to make sure the problem has not been fixed.
2. Be sure *gdb* is installed (usually installed but if not, it is easy to do so).
3. Run as root if at all possible (this enables using gdb to create a full `/tmp/cinelerra* .dmp` file).
4. Be sure `Settings → Preferences → Interface → trap SEGV` and `trap INTR` are checked.
5. Invoke the error. This should create the file `/tmp/cinelerra-<pid>.dmp`

For those who can not run as root or can not create the problem while running as root, another method to create a dump is available. Here are the steps to do so – you should also have gdb installed.

1. Temporarily login as root and key in:
   `echo 0 > /proc/sys/kernel/yama/ptrace_scope`
   This opens up *ptrace*, used by gdb (the debugger). You will want to reverse this when done.
2. Run CINELERRA-GG as an ordinary user from a window using the command line and try to create the problem. If the program crashes, this should produce a file named `/tmp/cinelerra*.dmp` where * is a number. The dump usually takes about 30 seconds or so to complete after the failure invokes the crash handler, so be patient.
3. Email the output that shows in the window from where you started up CINELERRA-GG and upload or directly email the `/tmp/cinelerra*.dmp` file (this file may be too large for the Mailing List).

4. After you are finished creating/testing the problem, reverse the previous setting while logged in as root by keying in:
   `echo 1 > /proc/sys/kernel/yama/ptrace_scope`

Even if you are not logged in as root or have ptrace enabled, if you have a .dmp file and know what the last thing you did is, please provide the .dmp file for analysis. There is a low probability that the dump will prove useful, but if the problem can be recreated it can usually be fixed!

18.5 When things go wrong

Sometimes things go wrong and there are some ways to continue your work without much trouble. Below is a list of items to try before abandoning your session.

1. If your computer or the program crashed, you can use the *File* pulldown choice of *Load backup* to get back to the last automatically saved session. It will most likely not include the last few operations that were done though. But if you forgot to Load backup when you restarted ,CINELERRA-GG you have a second chance to use `File → Load` and select `$HOME/.bcast5/backup .prev` as long as you only loaded a different file and have performed no editing operations.

2. If you accidentally destroyed your current project by a Load with *Replace current project* when you should have changed the Insertion strategy, and you had not saved a recent backup, this is another time when it is useful to use `File → Load` and select `$HOME/.bcast5/backup.prev` as long as you only loaded a different file and had performed no editing operations.

3. If you suddenly get errors or hangs on media or strange looking tracks, you might want to rebuild indexes. You can do this in the Resources window with a right mouse click on the media to get a pulldown with the *Rebuild index* option. For an easy way to rebuild for all of your media use `Settings → Preferences, Interface` tab, Index Files section, *Delete existing indexes*. Over time, as you load media, the indexes will automatically be recreated so each load takes a little longer, but this quite often is all that is needed to repair a media problem.

4. On an older computer, if you are playing media and it can not keep up, you can turn off *Play every frame* in the *Video Out* tab of `Settings → Preferences, Playback` tab. You will then see the video jump as it skips frames in order to stay caught up.

5. The *Cache size* can be lowered to 1048 if playback seems choppy or if you have problems with lv2 plugins, or you can increase the *Cache size* for better flow. This can be changed in `Settings → Preferences, Performance` tab.

6. After saving your session and settings and exiting CINELERRA-GG, you might want to rename your current `$HOME/.bcast5` directory and start with the default setup.

This will eliminate your settings as the potential cause of a problem; however, all of your preferences will be lost until you go back to your original `.bcast5`.

7. You can also temporarily rename just Cinelerra_rc in your $HOME/.bcast5 directory, so that a new file with the original name will be created with original defaults. You will lose your preferences, but it is just for testing and you can move back the renamed CINELERRA-GG_rc over the new one if that is not the cause of the problem. Be sure to stop and restart CINELERRA-GG whenever you rename this file.

8. Some media has only a single keyframe at the beginning of the file so that moving anywhere on the timeline results in just a black image in the compositor. CINELERRA-GG needs more keyframes to determine position. You can temporarily use `Settings → Proxy` to $\frac{1}{2}$ size to put in keyframes. How to use Proxy is explained in 19.3 Proxy Settings and Transcode.

9. If you have enabled an *Automation Curve* such as Fade or Speed on your track but it does not show, it could simply be because the range minimum and maximum are incorrectly set in the zoom bar. Set them to more reasonable values until you see the lines appear. Just as a reminder, the zoom bar is on the bottom of the main window (figure 8.1).

10. If the rate at which frames are captured during Recording is much lower than the framerate of the source, the video will accumulate in the recording buffers over time and the audio and video will become out of sync. Decrease the number of frames to buffer in the device in `Settings → Preferences, Recording` tab so that the excess frames are dropped instead of buffered.

11. If loading files locks up, this might be because CINELERRA-GG is building picons/vicons for the Resources window. If you load a large number of images it needs to decompress every single image to build a picon/vicon. Go into `Settings → Preferences, Appearance` tab and disable *Use thumbnails in resource window* to skip this process. Keep in mind though, that it only has to create these thumbnails the first time a new piece of media is loaded or the values are changed.

12. After a crash, look on the window where you started running CINELERRA-GG to see if there is a line towards the end of the output that reads something like a bunch of numbers followed by the word "AssetVIconThread". Frequently this indicates that some system resouce has been exhaused in creating/running the Vicon images in the Resources window. You can disable *Use thumbnails in resource window* in Settings/Preferences, Appearace tab to see if this resolves the crashing. In most cases when this is the cause, it will be the second to last line shown in the window. In addition, you might want to *Delete clip thumbnails* in the Interface tab section of Settings/Preferences.

13. For an older computer with less CPU power, in `Settings → Preferences, Appearance` tab, be sure that *Autocolor assets* is disabled; set *View thumbnail size* & *Vicon quality*& *Vicon color mode* to lower values or switch to *No Play* instead of *Full Play* in the Resources window (this is to the right of the word *Visibility* in the left hand side

of that window). You will then have more CPU and more memory available to do actual editing.

14. When using AppImage to run with the OpenGL video driver, you must have the OpenGL drivers for your Operating System graphics board installed as it is not included in the AppImage library set. The error message you might see if this is not installed is: *error while loading shared libraries: libGLU.so.1: cannot open shared object: No such file or directory*.

15. If you have updated your Operating System or newly installed some applications, it is possible that your LV2 plugin path may have been modified and be in disagreement with what you have set for CINELERRA-GG. This could result in a crash upon startup. Look at the messages in the window from where you started CINELERRA-GG to see if they look similiar to the following:

```
LOAD: http://eq10q.sourceforge.net/matrix\_lr2ms
LOAD: http://eq10q.sourceforge.net/matrix\_ms2lr
LOAD: https://community.ardour.org/node/7596
LOAD: urna-comp
**segv at 0x7f2cd80813c4 in pid 4540, tid 4540
```

    The last line before the *segv* indicates the name of the LV2 plugin that is causing problems. Please refer to the section on LV2 plugins 10.9 to resolve the issue.

16. Check your *Overlays* window if you do not see your Assets, Titles, Transitions, Plugin Keyframes, or other lines such as Fade, Cameras, etc. on your timeline. These items will still be functional, but you may be confused when you do not actually see their physical presence if you inadvertently unchecked them in the *Overlays* window. Use the *Window* pulldown to enable/disable the *Show Overlays* window.

17. BugTracker - sometimes there is a problem creating a new report issue in the website's Mantis Bugtracker using the Chrome web browser and you lose what you just typed in. Generally when logging into the bugtracker, the option "only allow with this IP address" needs to be disabled, then the bug tracker will work fine.

18. Forum -

    a) If you can not register your username in the website's Forum, it could be because the protection measures against spammers also is in effect with real users, so sometimes the website blocks the registration for the forum. This often affects email addresses of well-known email providers. Send email to the mailing list which requires you to subscribe ((`https://lists.cinelerra-gg.org/mailman/listinfo/cin`) so that you can be added to the forum manually as long as you confirm that you agree with the terms of use of this website and the forum.

b) In the website's Forum, use of unacceptable characters or strings can result in the error message "A potentially unsafe operation has been detected in your request to this site". You lose what you had typed in and you will have to check what you remember typing in for a non-converational set of characters. This can be as simple as ../../ (dot dot slash dot dot slash) or some types of C++ code.

c) Entries not showing up for several hours could be due to taking a long time to propagate around the world on some time basis.

d) Not seeing a comment or reply section could be because you are not logged in or recognized as logged in. If you seem to be logged in, logout and then login again to see if that resolves the problem. Another potential solution is the switch the language being used to English, even if you think it is already English. Sometimes it is left in a different language and needs to be reset.

e) Until you have made a few entries in the Forum, you will have to wait a few hours for your entry to be approved. This is because there are people who spam the forum so attempts to alleviate this by requiring moderation for new users until validated.

Some Helpful User Readable Text Dumps

On the *File* pulldown, there is a *Dumps* option with a submenu of dump EDL, Plugins, Assets, and Undo. In all cases, you will have to have started CINELERRA-GG from a window instead of from an application icon and the text results will be shown in that window.

- *Dump EDL* will display your current EDL state on the screen in the window from where you started .CINELERRA-GG This can be useful to see information concerning a specific edit or a file path.
- *Dump Plugins* will show the names of the currently loaded plugins.
- *Dump Assets* displays the media assets that you have loaded and various pertinent details on each, such as samplerate, width, and height.
- *Dump Undo* will dump the last 32 edits on the undo stack exactly as kept, which can be useful if you are looking to see how far back in the undo to go to get to a specific spot.

Common Problems

Some messages you may see in the startup window may or may not be errors. Some examples are:

FFMPEG::open_decoder: some stream times estimated: `/your directory/filename`

This is not a problem. Basically, when you open a file if a stream has a known duration, there is no message. If the duration is unknown, it is estimated by using the File Size and

Bitrate to estimate the duration. This may just indicate that the stream number CINELERRA-GG uses versus ffmpeg may be counted differently.

FFMPEG::scan: codec open failed followed by
FFMPEG::scan:file= `your directory/filename`

This is not a problem. CINELERRA-GG is building an index for your file in order to better seek. In that process, different methods are tried until a successful scan is complete.

int FFMPEG::init_encoder(const char);* followed by
bad file format: `your directory/filename`

This error occurs when you are rendering, or possibly capturing media via recording, when the file format/type are set to an incompatible option. To fix this in the Render window, check the Video and Audio wrenches configure compression settings and choose a compatible Compression as shown when clicking on the down arrow in the Preset window.

AudioALSA::write_buffer err -32(Broken pipe) at sample #

This indicates that there is something wrong with the audio. Some reasons for this are:

- You simply stopped playing in CINELERRA-GG while the audio is in progress.
- Running on a computer where there is no sound card.
- Incorrect setup of the audio parameters in the `Settings → Preferences, Playback` tab.
- Your sound system is already in use by another program, like when playing *tunes* outside CINELERRA-GG.

void AssetPicon::create_objects():

While rendering, you may see this in the Message log window as well as in your terminal startup window. Your render will be incomplete. This indicates that CINELERRA-GG is unable to update the image of the asset picon either on the Timeline or in the Compositor. Usually not a problem unless rendering large media such as 4K. Because this is just for displaying approximately where you are in the render process, turn off "Draw media" in the patchbay for the video tracks (3rd icon from the left). You can still see how far along the render is by checking the progress bar which shows the percent complete in the bottom right corner of the main program window.

Playback does not keep up so you hear the audio ahead of the video.

When this occurs, it most likely means that there is not enough CPU power to keep up. You can go into `Settings → Preferences, Playback` tab and uncheck *Play every frame*. Not playing every frame means that as it plays, if it can not keep up with doing all it has to do to change the frame into a vframe and then draw it, the program just notes that it is falling behind so skips frames to catch back up. This does not affect rendering. A different solution is to use Background Rendering as described in section 7.6.

CINELERRA-GG *will not start and produces error message on the startup window.*

There can be various reasons that CINELERRA-GG does not come up. Some of the recent reasons are listed here. Please notify the www.cinelerra-gg.org website when you have any unknown startup issues.

- CINELERRA-GG doesn't come up in Debian with compiz window manager running. Workaround is to use a different window manager or bring up cin first and then compiz. There is also a report that Compiz leads to single frame problems after a certain amount of time in the case where you switch to fullscreen mode and than back to normal node – cin stops working and so you will have to restart cin.
- When a library goes from one version to a later version, sometimes a pre-built CINELERRA-GG binary will fail because it was created at a different version than the one the user has on their computer. This seems to happen more frequently on Arch distros because Arch has continuous releases and is usually kept up to date. An example of the error message you might see in your startup window would be:
  `cin: error while loading shared libraries: libvpx.so.5: cannot open shared object file: No such file`

You can usually install the required library to fix the problem. A temporary fix may be to create a symlink but this must be done with extreme caution as it may create an unstable condition. A better workaround is to use a tarball to install the software instead of the package build until the libraries are in sync between the build and your Operating System install.

Loading a very large number of media files, for example 500 clips, crashes CINELERRA-GG *with messages similar to the following that are displayed in the window from where you started* CINELERRA-GG*:*

```
BC_DisplayInfo::init_window: cannot open display "".
BC_DisplayInfo::init_window: cannot connect to X server.
unjoined tids / owner 42
  00007feb27fff700 / 00007feb8f496700 9MainError
  00007feb267fc700 / 00007feb7affd700 18FFMPEGScanProgress
  ...
```

This usually indicates that you are out of Operating System file descriptors. You can increase the amount easily with the following command line: `ulimit -n 4096` where 4096 is a size suggestion but can be increased. You can include this command line in your `.bashrc` or `.profile` file for the user login which gets run every time you login, or modify the Operating System limit for everyone in the system file, which for Fedora is `/etc/security/limits.conf`. Alternatively, you can reduce the number of file descriptors needed by going into `Settings → Preferences, Appearance` tab and unchecking the flag *use thumbnails in resource window*.

Using a non-supported locale could cause a failure in starting CINELERRA-GG *with the following error:*

```
BC_WindowBase::init_im: Could not open input method.
unjoined tids / owner 1
  00007f543dffb700 / 00007f54989d5840 12BC_Clipboard
```

This is caused by LC_CTYPE setting you are using, and maybe other Locale settings too. For example, using the following will cause the *unjoined tids* error.

```
LC_CTYPE="en_IL" /mnt0/build5/cinelerra-5.1/bin/cin # set LC_CTYPE and
    ↪ start cin
```

This is an operating system error which is resolved by using the following startup instead:

```
LC_CTYPE="en_GB.utf8" /mnt0/build5/cinelerra-5.1/bin/cin # set LC_CTYPE
    ↪ and start cin
```

You can either export the LC_CTYPE variable before starting CINELERRA-GG or add this in your bash startup file or simply run from the command line as shown above.

Masking Feather is not working and produces error messages on the startup window similar to:

```
0:1(10): error: GLSL 4.30 is not supported. Supported versions are: 1.10, 1.20,
    ↪ 1.30, 1.40, 1.00 ES, and 3.00 ES
Playback3D::print_error:
```

```
#version 430 // vertex shader
in vec3 in_pos;
void main000() \{
gl_Position = vec4(in_pos-vec3(0.5,0.5,0.), .5);
}
void main() \{
main000();
}
```

In the mask window, check the box *Disable OpenGL masking* to use software instead of OpenGL.

Stack smashing detected error message.

In at least one scenario, the fix for this is to go to Settings -> Performance and set "Use HW Device" to "none".

18.6 Menu Bar Shell Commands

In order to provide some types of help, the Menu Bar Shell Commands are available for customization purposes (figure 18.1).

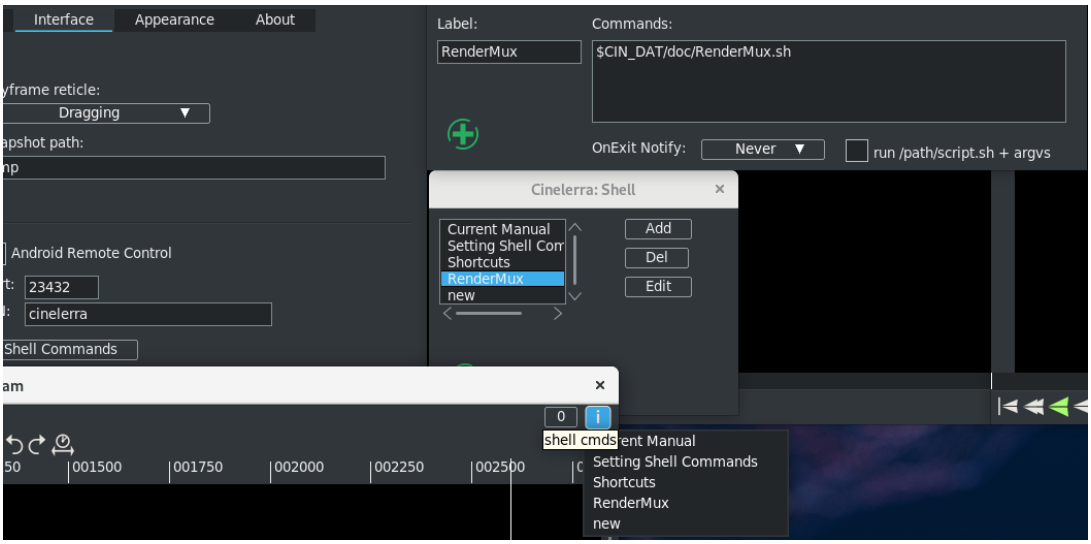


Figure 18.1: *Some windows used to manipulate Shell Commands scripts*

In the main window on the top line containing the *File*, *Edit*, . . . *Window* pulldown menus, all the way to the right hand side is the *shell cmds* icon. You might see a small white-color bordered box with the "i" inside and if you mouse over it, the tooltip says *shell cmds*. This is a configurable popup which gives you the ability to see a table of Shortcuts in html

format, refer to the reference manual or execute a pre-defined script to perform a specific task, such as some type of post processing which you plan on performing on a repeat basis. Unless redirected elsewhere, the output from executing any script-type commands will be displayed in the window from where you started CINELERRA-GG.

The Shell Cmds popup menu items are configured in:

`Settings → Preferences → Interface → Shell Commands`

This runs a dialog that lets you Add/Del/Edit the scripts in the main menu popup. Possible options are:

1. *Add* a new menu item/script:
    - click on the *Add* button
    - in the Label box, change the word *new* to the desired menu label (for example: Graphics Editor)
    - in the Commands box, type the command lines to be included in the script (for example: gimp)
    - click the green checkmark for OK in the Commands window
    - click the green checkmark for OK in the Shell window
    - click on OK or Apply in the Preferences window
2. *Del* to delete an existing menu item
    - select to highlight an entry in the CINELERRA-GG: Shell listbox
    - click on the *Del* button
    - click the green checkmark for OK in the Shell window
    - click on OK or Apply in the Preferences window (the entry is now deleted)
3. *Edit* an existing menu item:
    - select to highlight an entry in the CINELERRA-GG: Shell listbox
    - click on the *Edit* button (the Commands window appears)
    - set the label and/or commands to the new desired values
    - click on the green checkmark for OK in the Commands window
    - click on the green checkmark for OK in the Shell window
    - click on OK or Apply in the Preferences window

Next time you click on the *shell cmds* icon, you will have the changes you made in effect.

Both the *Add* and *Edit* options have checkboxes in the Commands window for the following:

- *OnExit notify* choose Always, Never, or On error to get a popup window notify

- run `path/script.sh + argvs` to run a particular script with argvs ($1, $2,... arguments)

When modifications to the Shell Cmds are included in a new CINELERRA-GG release, they are not added to your Preferences file automatically unless you create a new one. But if you have added no custom scripts, you can update your currently existing `$HOME/.bcast5/Cinelerra-GG_rc` file to delete the lines that are all grouped together that begin with the letters: `SHBTN` (the entire line – these are the **SH**ell **B**u**T**to**N**s). You should first Quit out of CINELERRA-GG, edit the file, and then when you restart CINELERRA-GG, it will create the new default buttons in that same file.

There are 2 specific CINELERRA-GG file names that have unique characteristics that allow for usage from a user-defined script. The first one is an environment variable, `CIN_RENDER` which is used in the *RenderMux* shell command that is already setup. It contains the complete path and filename of the file in your session that you selected as *Select a file to render to* when you ran a Render operation.

The second one concerns usage of the Resource window assets to use as argvs in shell scripts. The added command line can ONLY be a script. Then if your script refers to $1, $2 and $3 and you highlight 3 files in the Media folder, their full path name will be used as the argvs to your script. For example, if you added a shell commands script `/workspace/script.sh` that contains the following lines:

```
cp "\$1" /tmp/temporary_copy
cp "\$2" /tmp/new_copy
cp "\$3" /tmp/file_to_upload
```

and you highlight 3 files in the Resources window called *ABC.mp4*, *123.mp4*, and *Final.mp4* and execute your added script from the Shell Cmds icon, you will be copying those 3 highlighted files to the new names on `/tmp`.

There are some Help features currently included in the Shell Cmds menu. Those available are:

1. *Current Manual* in PDF format from cinelerra-gg.org or automatically downloaded.
2. *Setting Shell Commands how to* which explains how to configure your own commands.
3. *Shortcuts* html file for easily looking up a particular shortcut.
4. *RenderMux* shell script to use ffmpeg concatenate to copy files such as *look.mp4001*, *look.mp4002*, *look.mp4005*... that were rendered using *Create new file at each label* or with the Render Farm.

5. *Delete brender files in tmp* shell script to delete the many brenderX files in /tmp when using the default location.

18.7 CINELERRA-GG Command Line -h

To see the command line parameters available to use with CINELERRA-GG, key in:

```
<your Cinelerra-GG directory path>/bin/cin -h
```

What will be shown in the window where this line was entered is:

Usage:

```
./cin [-f] [-c configuration] [-d port] [-n nice] [-r batch file] [filenames]
```

-d = Run in the background as renderfarm client. The port (400) is optional.
-f = Run in the foreground as renderfarm client. Substitute for -d.
-n = Nice value if running as renderfarm client. (19)
-c = Configuration file to use instead of /root/.bcast5/Cinelerra_rc.
-r = batch render the contents of the batch file (/root/.bcast5/batchrender.rc) with no gui.
batch file is optional.
-S = do not reload perpetual session
-x = reload from backup
filenames = files to load

Chapter 19

Performance and other Tips

Performance of CINELERRA-GG is related to the software and video format being used in conjunction with your computer system hardware – the number of CPUs and its speed, I/O bus speed, graphics card, and amount of available memory. A basic, less powerful system will be sufficient for users working with audio only or lower resolution video formats. Higher end computers will be needed when playing and working with higher resolution formats, like 1080p or 4k. Adding effects and multiple tracks will require more cpu, memory, and various other resources to perform at an acceptable level.

Perhaps the easiest method for determining if your performance could be improved is to look at the numerical value displayed as *Framerate achieved*. Good performance means that when *Play every frame* is set in `Settings → Preferences, Playback A` tab, the frames/second (frames per second or fps) in playback might be almost always at the maximum rate of your project setting and/or video frame rate. You can check this in `Settings → Preferences, Playback A`, by watching *Framerate achieved* while playing forward. A higher number is better, up to the format frame rate of the video.

Some computer hardware factors to consider for better performance are listed here:

- Multi-core and more SMP processors greatly improve speed by making use of threads. CINELERRA-GG automatically uses the available threads, but some processes are single-threaded anyway and these become the weak link in the chain. The *Project SMP cpus* parameter is used to limit the use of threads in the deconding (playback) stage only. It is better to lower the number of threads to leave some for the system and for the plugins in use.
- RAM: In video editing in general, the more RAM the better. A large amount of free memory available can help speed up operations by avoiding unnecessary disk swaps and keeping videos easily accessible in memory. You can optimize RAM utilization with *Cache size* and *Seconds to preroll render* parameters. See section 19.4. If there is limited RAM you must necessarily have a large swap partition. For system swap, (2 x RAM) GB seems to be more than sufficient. If the amount of memory being used by the program is close, then swap might save you but often if swapping becomes necessary, it presents more problems and you end up killing the CINELERRA-GG process anyway.
- Video editing is almost always I/O intensive. To create longer running videos at

high resolution you will want to have a lot of disk space available on fast access disks. The higher the transfer rate, the less slowdowns and problems. So a modern SSD is better than an old HDD.

- CINELERRA-GG benefits from OpenGL hardware acceleration. A good graphics card is worthwhile to have.
- Multiple monitors really come in handy to increase productivity as you can see more information and in bigger windows so you do not have to keep moving windows around.

Besides the above hardware recommendations, this section covers tips for performance improvements and tips on how to perform some specific tasks, often for older media.

19.1 Hardware video acceleration

With certain newer, more powerful graphics boards and newer device drivers, there is the potential for enhanced *decode* and *encode* performance. Decode refers to loading and playing video in CINELERRA-GG. The GPU, Graphics Processing Unit, on the graphics board is accessed via one of the following libraries: vdpau or vaapi. The hardware acceleration done by the graphics card increases performance by activating certain functions in connection with a few of the FFmpeg decoders. This use makes it possible for the graphics card to decode video, thus offloading the CPU. Decode operations are described here next. Encode refers to rendering video and is described at the end of this section under GPU hardware encoding.

VDPAU, Video Decode and Presentation API for Unix, is an open source library to offload portions of the video decoding process and video post-processing to the GPU of graphics boards, such as Nvidia. It may also apply to Nouveau and Amdgpu boards (by wrapper), but that has not been verified.

VA-API, Video Acceleration API, is an open source library which provides both hardware accelerated video encoding and decoding for use mostly with Intel (and AMD) graphics boards.

AppImage will probably not allow for either VDPAU or VA-API hardware acceleration because the computer where AppImage is created will not have the same graphics card and/or vaapi/vdpau libraries as yours. It is recommended for best results that you build CINELERRA-GG on your specific computer as described in 1.3.2. So in summary:

- Hardware acceleration, vaapi or vdpau, works the same running either the created binary or the created AppImage from that binary on that same computer.
- Hardware acceleration using an AppImage created on another computer running a different Operating System or even just a different level of Operating System,

most likely will not work unless the user's computer just happens to have the same characteristics with respect to vaapi/vdpau as the computer where the AppImage was created.

Currently only the most common codecs, such as MPEG-1, MPEG-2, MPEG-4, H.264 /MPEG-4 and h265 (hevc), are accelerated/optimized by the graphics card to play these particular video formats efficiently. The other formats are not optimized so you will see no performance improvement since the CPU will handle them as before, just as if no hardware acceleration was activated. There are many different graphics cards and computer systems setup, so you will have to test which specific settings work best for you. So far this has been tested at least with Nvidia, Radeon, and Broadwell graphics boards on some AMD and Intel computers; depending on the graphics card, two to ten times higher processing speeds can be achieved. However, most graphic operations are single-threaded so that performing the operations in the hardware may actually be slower than in software making use of multiple CPUs, which frequently multi-thread many operations simultaneously.

19.1.1 GPU hardware decoding

1. Verify that you have installed *libva-dev* or *libva* on your operating system.
2. Verify that you have installed *libva-intel-driver* on your operating system
3. Verify that you also have *libvdpau-dev* or *libvdpau* installed.
4. Verify `Settings → Preferences, Playback` tab, Video Driver is set to *X11* – or *X11-OpenGL* if that produces better results for your configuration.
5. Before starting CINELERRA-GG, you can set an environment variable that can be easily reversed and then, to run from the CINELERRA-GG installed directory, key in:

```
CIN_HW_DEV=vdpau ./cin # for computers with Nvidia and some other graphics cards
CIN_HW_DEV=vaapi ./cin # mostly for computers with Intel or AMD specific graphics
    ↪ hardware
```

If you find that the environment variable setting is advantageous for your CinGG usage and you want to always use it, you can add it in your `$HOME` directory `.profile` file which takes effect every time you log in. The line you would add would look something like this:

```
export CIN_HW_DEV=vdpau
or
export CIN_HW_DEV=vaapi
```

In addition for a computer with dual graphics cards, you can select your specific device with an environment variable to decode and encode. For example:

```
CIN_DRM_DEC=/dev/dri/renderD129 CIN_DRM_ENC=/dev/dri/renderD128 ./cin
```

and substituting your specific location instead of /dev/dri/renderD129 or D128.

It might be more difficult to analyze problems as a result of using the GPU because of the wide variation in hardware. When you do not set the `CIN_HW_DEV` environment variable, the code will work exactly as before since this feature is self-contained.

There is also a `Settings → Preferences, Performance` tab, *Use HW device* flag with a pulldown to set up *none, vdpau, vaapi,* or *cuda*. To ensure it takes effect, it is best to set it the way you want, quit out of CINELERRA-GG and then restart. Its current purpose is for flexibility, but there is a possibility that it might eventually take the place of `CIN_HW_DEV` – both are not needed.

Precedence of the decode hardware acceleration settings are:

– `yourfile.opts` is checked first so is of the highest priority; special .opts usage is described below

– environment variable `CIN_HW_DEV` is checked next

– preferences *Use HW device* settings is of the lowest priority

Hardware decoding in CINELERRA-GG

There are 4 phases during CINELERRA-GG's handling of hardware acceleration. These first 2 steps occur just *before* the first read.

1. Check to see if Hardware Acceleration is enabled, usually indicated by `CIN_HW_DEV` being set to vaapi or vdpau. If enabled, try to activate the decode device, and if that fails revert to software.
2. The next step is to send some data to decode to see if that works. If this does not work, you will see an error message of *HW device init failed, using SW decode*.

These next 2 steps occur *during* any read. Now there is no turning back to software so if the hardware gets an error, that video will not be handled correctly.

3. Read the media and send the raw stream data to the device for processing.

4. Read the device to receive the decoded data, and convert it to the session color model. If the GPU can not convert the data, you will see the error message of *Error retrieving data from GPU to CPU*.

AppImage will probably not allow for either VDPAU or VA-API hardware acceler- ation because the computer where AppImage is created will not have the same graphics card and/or vaapi/vdpau libraries as yours. In other words:

- Hardware acceleration, vaapi or vdpau, works the same running either the created binary or the created AppImage from that binary on that same computer. See Build the CinGG.AppImage from scratch.
- Hardware acceleration using an AppImage created on another computer running a different Operating System or even just a different level of Operating System, most likely will not work unless the user's computer just happens to have the same characteristics with respect to vaapi/vdpau as the computer where the AppImage was created.

Due to these variations in user's computer hardware configuration, it is often suggested that you refer to your startup window to check for error messages. Since your situation is unique, the error may not have been seen by anyone else and is probably unknown/undocumented. For debugging problems, modify in the CINELERRA-GG ffmpeg subdirectory, the file: `decode.opts` by temporarily changing the line from *loglevel =fatal* to *loglevel =verbose* and restarting CINELERRA-GG.

Possible improvements or differences

1. The Frames Per Second (FPS) in playback might usually be at the maximum rate. You can check this in `Settings → Preferences, Playback A`, looking at *Framerate achieved*; the higher, the better.
2. Percent of the CPU used should be less, thus saving more CPU for other operations.
3. Some users get the the impression that playing seems smoother.
4. The CPU fan noise may go down as the CPU is used less.
5. The GPU graphics fan noise may go up as the GPU is used more.

Using the GPU is going to react differently depending on your hardware, software, and the number of files loaded. A good way to determine how well it is performing, is to watch the CPU load from another window running the command line routine `top`. Consider the following possibilities:

- If you only use smaller videos occasionally and then use other codecs than the ones mentioned previously as being the optimized set, it is usually a better idea to stay with the default settings without all the hardware tests.
- If you have 4 cores or less, but a really good *gaming card*, using vaapi/vdpau could be a big help.
- If you load only a couple of files, the GPU vaapi/vdpau should be faster depending on your graphics board and how capable it is.
- If you load 10 camera Mixers of H.264 format, it always seems to work a lot better playing them.
- If you have an epyc AMD chip with 128 CPUs and load 50 files, without vaapi/vdpau may be better.

Special .opts file

The situation may arise where you have enabled hardware acceleration and after loading several files for a project, you find that a file had some kind of error resulting in a black video instead of an image or you see an error message pop up which states something like *Error retrieving data from GPU to CPU* or *err: Unknown error occurred*. Because the `CIN_HW_DEV` environment variable is either all or none, ordinarily in order to correct the non-working video you would have to turn off hardware acceleration for the entire project/session. However, there is a way to continue working on your project without having to reload all of your files. You still use the environment variable and it will be effective for all of the formats it is able to handle, but you make an exception for any of the files that erred out. To do this you simply create a file in the same directory with the same name as the erring file with the different extension of .opts. The contents of this .opts file would just be the one line of:

```
cin_hw_dev=none
```

Conversely, if you have a bunch of files in your project, like dnxhd format, that are not hardware accelerated, but you have an accompanying large file of type .mp4 for which you would like the hardware acceleration, you can leave the `CIN_HW_DEV` variable unset (that is, do not use it) and just create an .opts file containing the line:

```
cin_hw_dev=vdpau
```

For example your file, `test.mp4`, would have a side-kick called `test.opts` that will use the GPU for decoding/playing and the other files will just use the software. This is of some advantage because the ones that can not use the GPU if the environment variable is enabled, will not have to even check which saves a minuscule bit of time.

It is important to note that if using the .opts file to override the default `ffmpeg / decode.opts` file, you will most likely see more warnings (not really errors) in the CINELERRA-GG startup window because the standard `decode.opts` file has *loglevel = fatal* whereas the default is *loglevel = error*. To avoid seeing all of the extra warnings, you can simply add the line *loglevel=fatal* to your .opts file.

To verify hardware acceleration

Probably the easiest way to tell if hardware acceleration is working, is just to look at the messages in the window from where you started Cin (not available if start using the application icon). For example load a png, dnxhd, or some other non-supported format file and you will see messages similar to those below. The line *HW device init failed, using SW decode* indicates that the vdpau/vaapi HW (hardware) decode is not available so will use SW (software) decode instead.

```
Failed to get HW surface format.
HW device init failed, using SW decode.
file:/tmp/media/aer_zypr.mp4
err: Success

or

Decoder dnxhd does not support device type vdpau.
HW device init failed, using SW decode.
File:/tmp/media/test.gxf
err: Success

or

HEVC with NVIDIA, VDPAU driver is buggy, skipping
```

AppImage does not provide this capability. If you would like to see more information on what is occurring, you can modify in the CINELERRA-GG ffmpeg subdirectory, the file: `decode.opts` by temporarily changing the line from *loglevel =fatal* to *loglevel =verbose* and restarting CINELERRA-GG. Then you will see messages in the startup window like:

```
[AVHWDeviceContext @ 0x7fc9540be940] Successfully created a VDPAU device
(NVIDIA VDPAU Driver Shared Library 390.116 Sun Jan 27 06:28:58 PST 2019) on X11
    ↪ display :0
[h264 @ 0x7fc950159380] Reinit context to 1920x1088, pix_fmt: vdpau
[h264 @ 0x7fc92c3dd4c0] Reinit context to 1920x1088, pix_fmt: yuv420p
```

Again, to measure the performance run `top` from another window to check the CPU usage which will go lower as more work is offloaded to the GPU graphics card instead (it may go down by 2 to 10 times) or check the *Framerate achieved* while playing.

Some mixed preliminary results that have been reported are provided below.

Case 1:

X11 Video Driver set in `Settings → Preferences, Playback A` tab:

CIN_HW_DEV=off ./cin	CPU 58%
CIN_HW_DEV=vdpau ./cin	CPU 32%
CIN_HW_DEV=vaapi ./cin	CPU 82%

X11-OpenGL Video Driver set in `Settings → Preferences, Playback A` tab:

CIN_HW_DEV=off ./cin	CPU 48%
CIN_HW_DEV=vdpau ./cin	CPU 12%
CIN_HW_DEV=vaapi ./cin	CPU 80%

Best is the least amount of CPU usage. Note that in this Case 1, using the X11-OpenGL video driver is better than using the X11 video driver.

Case 2:

X11 Video Driver set in `Settings → Preferences, Playback A` tab:

CIN_HW_DEV=off ./cin	CPU 60%
CIN_HW_DEV=vdpau ./cin	CPU 11%
CIN_HW_DEV=vaapi ./cin	CPU 60%

X11-OpenGL Video Driver set in `Settings → Preferences, Playback A` tab:

CIN_HW_DEV=off ./cin	CPU 67%
CIN_HW_DEV=vdpau ./cin	CPU 60%
CIN_HW_DEV=vaapi ./cin	CPU 67%

Best is the least amount of CPU usage. Note that in this Case 2, using the X11 video driver is better than using the X11-OpenGL video driver.

Older graphics cards or non-performing graphics cards will probably bring only a small amount of improvement or no speed advantage at all. You can check to see if vdpau is implemented for your specific Nvidia board at:

https://download.nvidia.com/XFree86/Linux-x86_64/304.137/README/supportedchips.html

And, you can see what your specific hardware and software might support by running either `vainfo` (available in the libva-utils package) or `vdpauinfo` (available in the vdpauinfo package) from the command line. For example which "Profiles" your hardware can handle is listed in the output such as Main and High. Partial examples of each are shown below.

```
# vainfo
vainfo: VA-API version: 1.4 (libva 2.4.0)
vainfo: Driver version: Intel i965 driver for Intel(R) Broadwell -2.4.0.pre1 (2.3.0-11-
    ↪ g881e67a)
vainfo: Supported profile and entrypoints
VAProfileMPEG2Simple
...
VAProfileH264Main
VAProfileH264High
...
VAProfileH264MultiviewHigh
VAProfileH264StereoHigh
...
VAProfileVC1Simple
...
VAProfileVP8Version0_3
```

```
# vdpauinfo
display: :0 screen: 0
API version: 1
Information string: G3DVL VDPAU Driver Shared Library version 1.0
...
Decoder capabilities:

name level macbs width height
----------------------------------------------------
MPEG1 ---not supported ---
MPEG2_SIMPLE 3 65536 4096 4096
```

```
MPEG2_MAIN 3 65536 4096 4096
H264_BASELINE 52 65536 4096 4096
H264_MAIN 52 65536 4096 4096
H264_HIGH 52 65536 4096 4096
VC1_SIMPLE 1 65536 4096 4096
VC1_MAIN 2 65536 4096 4096
VC1_ADVANCED 4 65536 4096 4096
```

One last item of note, *nvdec* is also enabled in the ffmpeg build, but at this time it is not known how this decode option on Nvidia graphics boards works or does not work in conjunction with vdpau.

19.1.2 GPU hardware encoding

Encoding using hardware acceleration of your graphics board GPU is included in CINELERRA-GG but it is of limited availability and works only with a specific set of hardware graphics boards, a certain level of graphics driver versions and only with certain ffmpeg formats. The encoding is done via vaapi (libva installed), which is known to work with Intel HD graphics boards and some others or via nvenc as developed by Nvidia for Nvidia graphics boards.

Broadcom, Intel HD, AMD

To use hardware acceleration for rendering (that is, encoding) you do not have to set a preference or an environment variable, as was required for decoding. To use this feature you use an ffmpeg render options file which specifies a vaapi codec, such as h264_vaapi. You must include this line in that options file to trigger the hardware probe: `CIN_HW_DEV=vaapi`

There are currently 4 options files available in the Render menu already set up for you that you see when you select the Video wrench and use the down arrow on the first line in the menu. These are:

h264_vaapi.mp4 known to work on Intel computer with Intel Broadwell graphics driver

mpeg2_vaapi.mp4 known to work on Intel computer with Intel Broadwell graphics driver

mjpeg_vaapi.mp4 error message of *open failed with mjpeg_vaapi...* on above computer

hevc_vaapi.mp4 error message of *open failed with hevc_vaapi...* on above computer

Other option files can be added as needed for your specific hardware if it is known to work for you, such as VP8 and VP9. An example of the included CINELERRA-GG's `ffmpeg/video/h264_vaapi.mp4` file (figure 19.1):

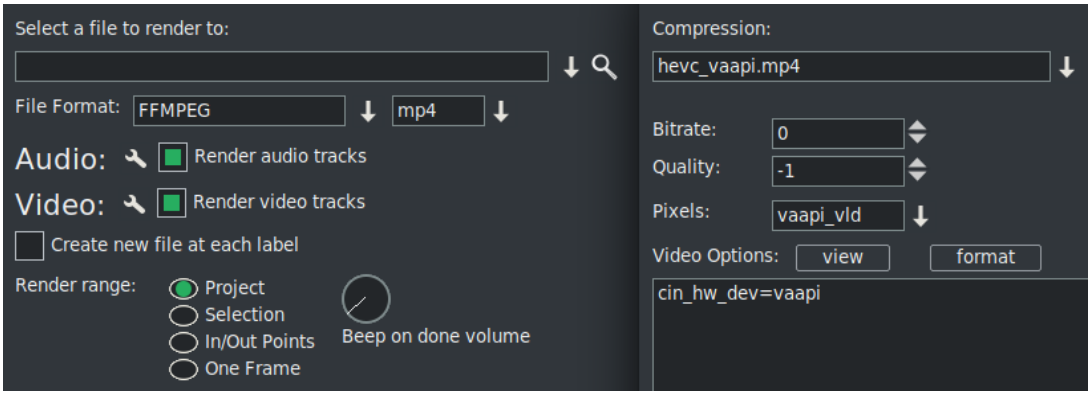


Figure 19.1: *Render menu setup to encode using GPU with vaapi*

```
mp4 h264_vaapi
cin_hw_dev=vaapi
profile=high
```

According to an online wiki, hardware encoders usually create output of lower quality than some software encoders like x264, but are much faster and use less CPU. Keep this in mind as you might want to set a higher bitrate to get output of similar visual quality.

Results of a particular test case performed on a Intel, 4-core computer, with Broadwell Graphics using an mp4 input video/audio file with dimensions of $1440x1080/29.97fps$ is shown next (note, filename is `tutorial.mp4`). This may very well be a *best case* scenario! But clearly, at least on this computer with only 4 cores, the hardware acceleration seems to be quite advantageous. A comparison of the 2 output files using `ydiff` as described in the Appendix (C.1) shows no obvious defects.

	CPU usage	Render Time	File Size	File
none	388%	100 secs	36,862,542	h264.mp4
vaapi	150%	19 secs	74,522,736	h264_vaapi.mp4

Nvidia graphics boards

To use hardware acceleration for rendering (that is, encoding) you do not have to set a preference or an environment variable, as was required for decoding. To use this feature you use an ffmpeg render options file which specifies the nvenc codec, either `h264_nvenc.mp4`, `h265_nvenc.mp4`, or `h264_nvenc.qt`. There are several requirements in order for this to work on your computer as listed here:

1. Nvidia graphics board at or above a certain hardware level. For h265, newer boards are required. Check this support matrix to determine if your board will work:

https://developer.nvidia.com/video-encode-and-decode-gpu-support-matrix-new

2. Software drivers for your graphics board must be installed on your computer.
3. The driver must support at least API version 12.0.16 – minimum required Nvidia driver for nvenc is 550.54.14. You will see error messages on the startup window if you are on lower versions.

If you try to render using `h264/h265_nvenc` formats and do not have an Nvidia graphics card or this feature was not built in, you will see in the window from where you started CINELERRA-GG, the error message: *Cannot load libcuda.so.1*

A small test using 2 minutes from the 4k version of Big Buck Bunny shows using nvenc can be about 4 times faster. The test was done on a 4 core Intel laptop with an Nvidia 950M graphics board.

	CPU usage	Render Time	File Size	File
none	388%	20 mins 18 secs	156,517,069	h264.mp4
nvenc	252%	5 mins 44 secs	42,052,920	h264_nvenc.mp4

Of note in this test, 388% CPU usage with only 4 cores shows that there is probably slow down because there is no more CPU power available. Therefore, using the GPU hardware acceleration with nvenc provides a significant speed-up. Also, note the larger file size without making use of the GPU – this probably indicates that there is a big difference in bitrate or quality parameter settings used in the options file and this should be taken into consideration.

Important Tip

There is one last potentially significant graphics speedup when using the X11-OpenGL driver for users with Nvidia graphics boards who are seeing frames/sec achieved lower than what the video format is set to. You may want to disable *sync to vblank* (an option for OpenGL) in NVIDIA X Server Settings for the proprietary drivers. This could increase your frames per second on playback.

19.1.3 Effects (OpenCL, Cuda)

CUDA® is a parallel computing platform / programming model developed by Nvidia that provides big increases in computing performance through use of the GPU. It was first introduced in about 2006 for applications in computationally intense fields such as astronomy, biology, chemistry, and physics.

At the time this was written, the use of Cuda is not going to improve the playing and rendering of video in CINELERRA-GG except in the case where you use a specific Cuda-enabled plugin that is computationally intense – sadly, most of what Cin does, Cuda will not help. Cuda is mostly a *block oriented algorithm* which works well for such things as *a flock of birds all flying next to each other*.

The same as for vaapi and vdpau, you can enable Cuda in the:
`Settings → Preferences, Performance` tab, *Use HW Device*
but it will not affect anything unless you have Cuda installed on your system and have built CINELERRA-GG yourself with Cuda build enabled. To install it on your computer, you will need to do the following:

1. Make sure you have the Nvidia proprietary library drivers for your graphics board already installed and up to date.
2. Go to the Nvidia Cuda development website and choose one of the available operating system's such as Fedora, OpenSuse, CentOS, Ubuntu, ... at `https://developer.nvidia.com/`
3. You will be installing repositories by package – this will be around 3 GB.
4. Also, install the Fusion repo, although it is unknown if necessary or not.

There is a very good set of directions on the website to just follow. Once you have installed the Cuda software on your computer, you must build CINELERRA-GG yourself – the default flag for a build in the configure script for cuda is *auto*. For Arch, and possibly other distros, you may have to supply the CUDA_PATH if it is not in the standard place. This would be something like:

```
  env CUDA_PATH=/opt/cuda
OR if using the bash shell:
  export CUDA_PATH=/opt/cuda
```

For all distros you will need to add to the ./configure line in your build script, the following:

```
  --with-cuda
so your configure line will look something like this:
  ./configure --with-single-user --with-booby --with-cuda
```

There are currently 2 available plugins for *show and tell* that take advantage of the hardware acceleration of Cuda – *N_Body* and *Mandelbrot* (see CUDA plugins).

An error you may see on your CINELERRA-GG startup window when you have Cuda installed and try to run one of the 2 plugins is *cudaErrorInsufficientDriver*. This indicates CUDA 10 (the current version at the time of this writing) is not compatible with the driver version on your computer. You can either:

1. Upgrade the driver if your board supports newer nvidia builds.
2. downgrade the cuda development package to a version that works for your board.

19.1.4 Additional topics on hardware acceleration

In wrapping up this Hardware Acceleration section, you may want to refer to the following to determine the current supported formats:

`https://wiki.archlinux.org/index.php/Hardware_video_acceleration`

19.2 Optimized Playback – X11 Direct

Normally, when CINELERRA-GG reads a video frame, it is copied into a *Vframe*. This frame may also need other actions performed on it, such as a color model change. In addition, ffmpeg and libzmpeg *can_scale_input*. So the read can be color transformed and scaled just by asking the library to do that. This means that if the compositor is in a *good* state with no zoom, the VFrame read can be done in the fastest render color model, and already scaled to the correct size for the compositor. In reality, this is not what you need for editing, but quite often the *virtualconsole* is not used because the render is media only – that is *just data*. If no data transforms are needed and the input scaling can be done, the vrender program detects this, and tells the codec to transmit the data in a compatible way to the compositor canvas. This is the *X11 direct* data path.

With the X11 video driver choice, large format files such as 4K, will playback faster than either X11-XV or X11-OpenGL. However, you still have the option to turn off the X11 direct data path if you use `Settings → Preferences, Playback A` tab set video driver to X11 and uncheck *use direct X11 render if possible*.

19.3 Proxy Settings and Transcode

Information on proxies to reduce required CPU for less powerful computers is in the section Proxy.

Information on transcode which is used to provide keyframes to make the video seekable is in the section Transcode.

19.4 Cache size and Seconds to preroll render

Cache size in `Settings → Preferences, Performance` tab is used to store images on the timeline. One 1080p frame uses about 10 MB. The default setting is 256 and this is enough for testing and running. However, why not use more memory if it is available. To experiment for testing a good number tuned to the way you use your computer, set the cache to 0, start up CINELERRA-GG, load a typical media file, play it and run *top* on the command line in another window to see how much memory is being used. In the *top* display, look at *free* memory. Whatever your computer is not using, is a good number to use for cache. If you start other programs, or change the design of the session so that it uses a lot of frame storage, you may need to experiment again later and resize accordingly. The system keeps all requested data cached until it is replaced by other data or you reboot your PC. Reboot if you want to clear the cache.

Seconds to preroll render in `Settings → Preferences, Performance` tab are used to increase the amount of frames (which are calculated in advance) at the time of their use. The default setting is 0.5 sec but you can increase it to 1.0 sec to improve smoothness in the timeline. Higher values are always less effective.

19.5 Tips for Improving Smaller Computers Use

A list of items to check for smaller computers that will help to use less cpu/memory/resources follows:

- For large media files, use proxy to do your main editing.
- In `Settings → Preferences, Appearance` tab, uncheck *Use thumbnails in resource window*.
- In `Settings → Preferences, Appearance` tab, uncheck *Autocolor assets*.
- Speed-up certain time-consuming FFmpeg plugins through use of a carefully selected `.opts` file. See Speed-up of Ffmpeg plugin usage with OPTS files
- For large media files, in `Settings → Preferences, Playback A`, Video Driver set *use direct X11 render if possible*.
- For the Video Driver in `Settings → Preferences, Playback A`, if using a good graphics card, choose *X11-OpenGL*.
- Set *CIN_HW_DEV=vdpau* or *vaapi* to use the graphics GPU for certain ffmpeg media decoding.
- If you have multiple cpus or multiple computers, even if slow, take advantage of using *Render Farm*.

- When editing, *background rendering* causes temporary output to be rendered constantly while the timeline is being modified. The temporary output is displayed during playback whenever possible so it does not have to be recalculated – very useful for transitions and previewing effects that are slow.
- In `Settings → Preferences, Playback A`, uncheck *Play every frame* which means frames will be skipped as playback of the video falls behind.
- Adjust *Cache size* in `Settings → Preferences, Performance` tab, to not exhaust the memory and yet still provide decent playback.
- When working with multiple tracks containing video at opacity 100%, if you just want to play one of the tracks, disable "Play track" in the patchbay for the tracks that you do not need to see. This uses a lot less CPU.
- For playing a track that has attached Video Effects, if you just want to play the track, you can temporarily disable the Video Effects which will decrease CPU considerably in most cases.

19.6 General Crash Handling Tips

This section is a handy guide for describing various kinds of software computer system failures. Only some of these various lockups or crashes can be dealt with. Hopefully, it will help to have some hints to know what kind of failure it is, or to save your work or to avoid future problems. For most of this, your user name must be root, although you can certainly try to see if it works for you when not root.

System lockups: When the system locks up, it is usually a system problem. Normally an application program cannot lock up the system. It is a major goal of system design to prevent an application (app) from failing a system interface. This does not mean an app can not cause a system lockup, but it is unusual.

CINELERRA-GG crash: This is covered in Crash Dumps for Analysis. Just a reminder that for best results you should be root and by providing a crash dump and as much other information as possible, you will be helping the developer to analyze the problem and fix it so that it can be avoided in the future.

X Server crash: Keyboard does not respond, screen is frozen, caps lock may operate LED light. Sometimes using Ctrl-Alt-F1 ... Ctrl-Alt-F7 (etc.) will allow you to regain control of a VT console. You can use this to login and check logs: eg. `/var/log/Xorg.0.log`, *dmesg*, *journalctl* ... etc. If you have another computer, make sure a terminal server is configured (for example: rsh, ssh, or telnet), then remote login via this other computer and check the logs. Most important is to immediately note the current software state, and the very last thing that preceded the crash, i.e. last button click, last keystroke, ... or whatever.

Kernel crash: The machine goes completely dead. The keyboard caps lock LED will probably be flashing. Most likely the only way to see anything after the kernel crashes is to use a serial port console log and usually kdb, the kernel debugger, and special cabling. This requires a lot of setup, and is normally reserved for experts. Login from another computer will not work. Pinging the ip address will not respond since the network stack is part of the kernel. There are some virtual machine setups that will let you debug a *guest* kernel, but this also requires a lot of setup, and affects which kernel is currently under test. The kdb route is preferable.

Keyboard grabs, Server grabs, and Deadlocks: A grab is an X-server state where all events are forced to just one window event stream. This forces the user to respond to the dialog. Things seems to be working, but no keypresses do anything useful. The system clock and other programs will still be working. The network will work for remote logins. Grabs can be canceled if the `/etc/X11/xorg.conf` X config contains special setup as shown below:

```
Section "ServerFlags"
   Option "HandleSpecialKeys" "Always"
   Option "AllowDeactivateGrabs" "True"
   Option "AllowClosedownGrabs" "True"
EndSection

Section "InputDevice"
   Identifier "Keyboard"
   Driver "evdev"
   ...
   Option "XkbOptions" "terminate:ctrl_alt_bksp"
   Option "XkbOptions" "grab:break_actions"
EndSection
```

or to `$HOME/.xinitrc`, add:

```
# xkb terminate/grab actions disabled in xorg.conf, use:
setxkbmap -option "grab:break_actions"
setxkbmap -option "terminate:ctrl_alt_bksp"
ctrl-alt-bksp = terminate the X-server, may restart automatically
```

Modal forms (always on top, and usually ptr/kbd grab) dialog boxes can lock a system by putting a form over another form holding a grab. This means the form that needs input may never get any because you can not get to it, and the result is a deadlock. Usually you will have to restart X (Ctrl-Alt-Bksp).

Window Manager issues: The *desktop* window manager can intercept and modify all kinds of user input. Mostly, this is a good thing, but can be a nuisance. If user keypresses can be programmed to trigger actions, then they may be useful to send `KILL` or `INTR` to an app that seems to be holding X's attention. For example:

```
killall -INTR cinelerra,
killall -9 cinelerra,
killall X,
# but you must run as root to be able do this
```

The ALT and META keys may be intercepted by the window manager, and this can cause unexpected interface operations.

19.7 Tips for Specific Operations

19.7.1 Generating a 440 Hz tone

To create a specific 440 Hz tone, follow these steps. You can vary the length, use more channels, or change the frequency to a different desired value (figure 19.2).

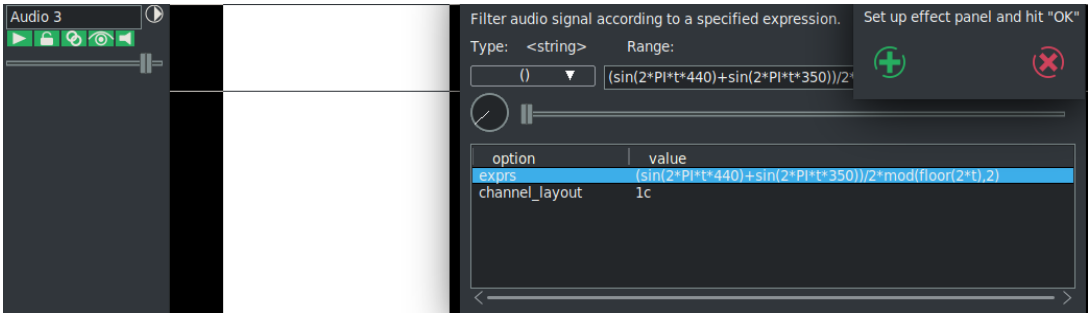


Figure 19.2: *Use Audio→Render effect to set render parameter values and then that effect can be varied.*

1. Make sure there is an armed audio track on the timeline, get into Cut and Paste mode, and highlight a selection or define In/Out points where you want to insert the audio tone.

2. Go to `Audio → Render effect`. Rendered effect usage is described in Effect Plugins. This brings up a menu where you will select the desired effect which in this case is *F_aeval*. Also Select a file to render to, a File Format, and Insertion strategy of Paste at insertion point.

3. Click on the green OK checkmark which will popup the F_aeval effect so that you can set the parameters.

4. Highlight the *exprs* option and key in a specific audio filter expression which for 440 Hz would be: $\sin(2\pi t \times 440)$. Then hit the Apply button.

5. Next when you hit the green OK checkmark on the CINELERRA-GG: Effect Prompt popup, you will have your 440 Hz tone on the timeline plus in the select file that you chose to render it to.

6. To use 2 channels instead of 1, in the F_aeval menu highlight the *channel_layout* option and change that to 1C|2C instead of the usual default of 1C.

19.7.2 Camera supplied LUTs

A LUT, acronym for Look-Up Table, is a mathematically precise way of taking specific RGB image values from a source image and modifying them to new RGB values by changing the hue, saturation and brightness values of that source image. In other words, LUTs are used to map one color space to another. Some high-end cameras supply a .cube file to use as input. There are several different ffmpeg plugins included with CinGG for using Lut's. These are:

F_lut: Compute and apply a lookup table to the RGB/YUV input video.

F_lut1d: Adjust colors using a 1D LUT "file" input.

F_lut3d: Apply a 3D LUT "file" to an input video.

F_lutrgb: Compute and apply a lookup table to the RGB input video.

F_lutyuv: Compute and apply a lookup table to the YUV input video.

For example, to use a 3dlut simply load your video, drop the F_lut3d plugin on that track, and bring up the lut3d controls window, highlight the *file* option, key in your file name (with path), and hit Apply to have the lut take effect. To easily adjust, move the *fader* slider in the patchbay for that video track. Only F_lut1d and F_lut3d allow for a file input and these usually are files with the .cube extension.

19.7.3 Encoding into Dolby Pro Logic

Dolby pro logic is an easy way to output 6 channel audio from a 2-channel soundcard with degraded but useful results. Rudimentary Dolby pro logic encoding can be achieved with usage of some effects.

1. First, create the front left and right channels. Create 2 audio tracks, each carrying either the left or right channel. Pan the left channel to the left and the right channel to the right with the pan control.

2. Now create the rear left and right channels. Create another 2 audio tracks as above – the left channel panned left and the right channel panned right. Then apply invert audio to both new channels and the signals will come out of the rear speakers.

3. Next, create the center channel by creating a single audio track with monaural audio from a different source. Center it with the pan control and the signal will come out of the center speaker.

- If a copy of the signal in the back speakers is desired in any single front speaker, the signal in the back speakers must be delayed by at least 0.05 seconds and a single new track should be created. Pan the new track to orient the signal in the front speakers.
- If the same signal is desired in all the speakers except the center speaker, delay the back speakers by 0.5 seconds and delay either the front left or front right by 0.2 seconds.
- If you want to hear something from the subwoofer, create a new track, select a range, drop a synthesizer effect, and set the frequency below 60Hz. The subwoofer merely plays anything below 60Hz or so.

Other tricks you can perform to separate the speakers are parametric equalization to play only selected ranges of frequencies through different speakers and lowpass filtering to play signals through the subwoofer.

19.7.4 Improving Analog TV Quality

The picture quality on analog TV is not always good but you can modify parameters in CINELERRA-GG to make it look more like it did in the studio.

First, when capturing the video, capture it in the highest resolution possible. For Europeans this would be 720 × 576 and for North Americans, 720 × 480. Do not adjust the brightness or contrast in the recording monitor, but you might want to max out the color. Capture the video using MJPEG or uncompressed Component Video if possible; if not possible, then capture it using JPEG preferably or RGB if that is all that will work. Now on the timeline use `Settings → Format` to set a YUV colorspace, drop a *Downsample* effect on the footage and set it as follows:

```
Horizontal:    2
Horizontal offset:  0
Vertical:    2
Vertical offset:   0
   red
x green
```

```
x blue
    alpha
```

Use the Camera in the compositor to shift the picture up or down a line to remove the most color interference from the image. If you have vertical blanking information or crawls which constantly change in each frame, block them out by using a Mask. This improves compression ratios. More invasive quality improvement techniques involve removing the interlace via deinterlacing.

19.7.5 Remove Interlacing

Interlacing often exists on older video sources, such as camcorders, and was previously used in broadcast television. Playing this video results in jagged images on a computer monitor, but with CINELERRA-GG you can use deinterlacing effects to solve this. After some experimentation, it has been determined that the FFmpeg *F_kerndeint* plugin seems to produce the best results with the least amount of fiddling. But some of the parameters described next are pertinent to other potential plugin usage.

Line Doubling: done by the *Deinterlace* effect when set to *Odd* lines or *Even* lines. When applied to a track it reduces the vertical resolution by $\frac{1}{2}$ and gives you progressive frames with stairstepping. This is only useful when followed by a scale effect which reduces the image to half its size.

Line averaging: the *Deinterlace* effect when set to *Average even* lines or *Average odd* lines does exactly what line doubling does except instead of making straight copies of the lines, it makes averages of the lines. This is actually useful for all scaling.

Inverse Telecine: this is the most effective deinterlacing tool when the footage is an NTSC TV broadcast of a film. It is described in Effect Plugins (10.10.46).

Time base correction: the previously discussed three tools either destroy footage irreversibly or do not work at times. Time base correction may be a better tool to use because it leaves the footage intact. It does not reduce resolution, perceptually at least, and does not cause jittery timing.

Frames to Fields effect: converts each frame to two frames, so it must be used on a timeline whose project frame rate is twice the footage's frame rate. In the first frame it puts a line-averaged copy of the even lines. In the second frame it puts a line-averaged copy of the odd lines. When played back at full framerate it gives the illusion of progressive video with no loss of detail. This effect can be reversed with the *Fields to Frames* effect, which combines two frames of footage back into the one original interlaced frame at half the framerate. However, keep in mind that Frames to Fields inputs frames at half the framerate as the project. Effects before Frames to Fields process at the reduced framerate. The output of Frames to Fields can not be compressed as efficiently as the original because it introduces vertical twitter and a super high framerate. Interlaced 29.97 fps footage can be made to look like film by

applying Frames to Fields and then reducing the project frame rate of the resulting 59.94 fps footage to 23.97 fps. This produces no timing jitter and the occasional odd field gives the illusion of more detail than there would be if you just line averaged the original. It is described in Effect Plugins (10.10.35).

HDTV exceptions: 1920 × 1080 HDTV is encoded in a special way. If it is a broadcast of original HDTV film, an inverse telecine works. But if it is a rebroadcast of a 720×480 source, you need to use a time base and line doubling algorithm to deinterlace it.

19.7.6 Making video look like film

With an older camcorder video which has low quality video, you can improve the results by turning it into progressive 24 fps output as close as possible. Only do this for low quality video.

1. Set project framerate to twice the video framerate.
2. Apply a *Sharpen* effect. Set it to sharpness: 25, no interlacing, and horizontal only.
3. Drop a *Frame to Fields* effect on the same track. Set Average Empty Rows to on and play through the video a few times to figure out which field is first. If the wrong field is first, the motion is shaky. Any editing in the doubled frame rate may now damage the field order. It is not clear which is the easiest way to support warnings for field glitches but you should go back to the normal framerate to do editing or play test to make sure the fields are right.
4. Render just the video to the highest quality file possible.
5. Import the video back to a new track. Set the project framerate to 24. The new track should now look more like a file with sharper images than the original footage.

This entire procedure could be implemented in one non-realtime effect, but the problem with that is you will most often want to keep the field based output and the 24 fps output for historical purposes. A non-realtime effect would require all that processing just for the 24 fps copy.

19.7.7 Clearing out haze

If you photograph a lot of haze instead of blue sky, these horizon shots will usually need more depth. You can use the `gradient` effect to improve your video. Drop the gradient effect on hazy tracks and set the following parameters:

```
Angle:     0
Inner radius: 0
Outer radius: 40
```

```
Inner color: blue 100% alpha
Outer color: blue 0% alpha
```

It is important to set the 0% alpha color to blue even though it is 0% alpha. The color of the outer alpha is still interpolated with the inner color. This is a generally applicable setting for gradient. Some scenes may work better with orange or brown for an evening feel.

19.7.8 Making a ringtone for a cell phone

1. Go to `File` → `Load files...` and load a sound file with Insertion strategy: *Replace current project*.
2. Go to `Settings` → `Format` change *Channels* to 1 and *Samplerate* to 16000 or 22050.
3. Highlight a region of the timeline to use for the ringtone. To improve sound quality on the cell phone, you need the maximum amplitude in as many parts of the sound as possible.
4. Right click on track Audio 1 and select *Attach effect*. Highlight the *Compressor* effect and hit Attach in the attachment popup.
5. Make sure the insertion point or highlighted area is in the region with the Compressor effect.
6. Right click on track Audio 2 and select *Attach effect*.
7. Highlight Audio 1 Compressor and hit Attach.
8. Click the Audio 1 Compressor's magnifying glass to bring up the compressor GUI.
9. Set the following parameters:

```
Reaction secs: -0.1
Decay secs: 0.1
Trigger Type: Total
Trigger: 0
Smooth only: No
```

10. Click Clear to clear the graph. Click anywhere in the grid area and drag a new point to 0 Output and -50 Input. The graph should look similar to the figure 19.3.
11. Go to `File` → `Render`. Specify the name of an mp3 file to output to. Set the file format to MPEG Audio. Click the wrench for Audio and set Layer to III and Kbits per second to 24 or 32. Check Render audio tracks and uncheck Render video tracks. Hit OK to render the file.

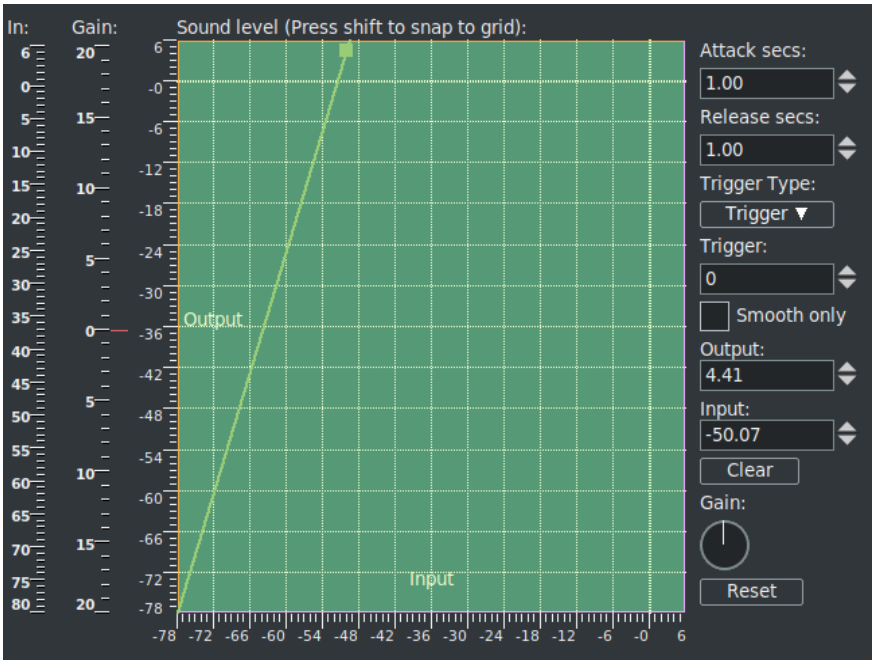


Figure 19.3: *Using the Compressor plugin graph to create a ringtone*

The resulting .mp3 file can be uploaded to a web server and then the phone's web browser can download the .mp3 file directly from the URL. There may be a size limit on the file.

19.7.9 Time stretching audio

It may appear that time stretching audio is a matter of selecting a region of the audio tracks, enabling recording for the desired tracks, going to `Audio` → `Render Effect`, and applying *TimeStretch*. In actuality there are 3 audio effects for time stretching: Time Stretch, Resample, and Asset info dialog.

Time Stretch applies a fast Fourier transform to try to change the duration without changing the pitch, but this introduces windowing artifacts to the audio. It is only useful for large changes in time because obvious changes in duration make windowing artifacts less obtrusive.

For smaller changes in duration, in the range of 5%, *Resample* should be used. This changes the pitch of the audio but small enough changes are not noticeable. Resample does not introduce any windowing artifacts, so this is most useful for slight duration changes where the listener is not supposed to know what is going on.

Another way to change duration slightly is to go to the Resources window, highlight the media folder, right click on an audio file, click on *Info*. Adjust the sample rate in the Info dialog to adjust the duration. This method also requires left clicking on the right boundary of the audio tracks and dragging left or right to correspond to the length changes.

19.7.10 Pan and zoom: still images

CINELERRA-GG's powerful keyframe features allow you to use pan and zoom effects on still pictures.

1. Load and create a clip from a still image. Make the clip 10 seconds long.
2. Activate the automatic generation of keyframes.
3. Using the transport controls, go to the beginning of the clip.
4. Using the compositing camera control set the clip's initial position.
5. Using the transport controls, move forward a couple of seconds on the clip.
6. Dragging on the compositing camera, move the camera center to a new position further along.
7. Next go to the beginning of the clip and play it.

You can see that the camera smoothly flows from keyframe point to next keyframe point, as CINELERRA-GG automatically adjusts the camera movement in straight lines from point to point.

19.7.11 Video lagging behind Audio

When there is a lag between the Audio and the Video, it can be because there are a lot of video frames and the computer can not display them as fast as the Audio can play the samples. However, this does not affect the rendered media in that the Audio and Video will be correctly synchronized. When playing the original media, you can alleviate the lag between the Audio and Video by disabling *Play every frame* in `Settings → Preferences, Playback A` tab. Now frames will be skipped in order to keep the audio/video in synch.

19.7.12 How to remove letterbox/pillarbox bands

To remove the horizontal black bands of the letterbox or the vertical black bands of the pillarbox we need to change the *size* and *aspect ratio* of the source by cropping. For example, if we want to remove the letterbox from a 4 : 3 frame to leave only the content with aspect ratio 3 : 2, We have to change the project format by doing the following steps:

1. Check the size of the base W of the original frame in pixels:

   `Resource` window → `RMB` on Asset → `Info` → `Detail`; e.g. W = 768 px

2. Obtain the height of the figure in 3 : 2, i.e., without the black bands; H can be obtained from the formula:

   $\frac{3}{2} = \frac{W}{H}$ from which $H = \frac{768 \times 2}{3}$ e.g., H = 512 px

3. Note that $W \times H = 768 \times 512$ is just the crop we are looking for to switch from 4 : 3 frame to 3 : 2 frame without letterbox.
4. Open *Set Format* window: `Settings` → `Format`
5. Change $H = 512$ and set *Display Aspect Ratio* to 3 : 2; press `Apply` and `OK`. Note that we leave W unchanged, since the frame width does not change.
6. If needed, use the *Camera* tool to get the desired viewport.

Note: in complex situations, with multiple sources of different sizes, it may be appropriate to perform an additional step first: change the size of the track on the Timeline via `RMB` → `Resize track`. In this way we crop the track to match it to the project format that we will change in the next step. Thus we avoid possible unwanted distortions.

In the case of the pillarbox, we will leave H unchanged while calculating the new value of W. The formula $\frac{x}{y} = \frac{W}{H}$ is valid for any aspect ratio ($4:3; 16:9; 2.35:1$; etc).

Chapter 20

Translations

There are several *po* files for various languages to make CINELERRA-GG more usable for non-English countries. These are listed below along with the po file name with the translations, the frequency of updates, and date of last update as of this writing.

English	cin.po	Baseline	(en_US)
French	fr.po	Updated often	Last update 2019 May
German	de.po	Updated often	Last update 2019 December
Russian	ru.po	Updated often	Last update 2019 October
Spanish	es.po	Updated often	Last update 2021 July
Basque	eu.po	From CV with Google Translate	Last update 2016 October
Italian	it.po	From CV with Google Translate	Last update 2016 October
Norwegian	nb.po	From CV with Google Translate	Last update 2016 October
Portuguese	pt.po	From CV with Google Translate	Last update 2016 October
Slovenian	sl.po	From CV with Google Translate	Last update 2016 October
Chinese	zh.po	Only Google Translate	2016 October
Greek	el.po	Only Google Translate	2016 October
Hindi	hi.po	Only Google Translate	2016 October
Hungarian	hu.po	Only Google Translate	2021 October
Japanese	ja.po	Only Google Translate	2016 October
Korean	ko.po	Only Google Translate	2016 October
Ukrainian	uk.po	Only Google Translate	2016 October
Vietnamese	vi.po	Only Google Translate	2016 October

A program, *xlat.C*, assists in providing several variations of text files that can be used in order to allow anyone to help make meaningful translations. All of the *po* files are located in CINELERRA-GG's `/po` subdirectory. There are 3 different ways to proceed described below.

Because CINELERRA-GG frequently is changing, it is a good idea to start by building a new `cin.po` file which contains the latest messages/words in English to be translated, along with a comment line of the routine name and line number. To create this, run the following

lines from a window:

```
cd /{your top level cinelerra directory}
./po/xlat.sh > /tmp/cin.po
```

Method 1 use the freely-available *poedit* program to provide translations to the current *xx.po* where xx is your language such as `fr.po`, `de.po`, `ru.po`, etc. The drawback to this is that *xx.po* files are not recreated monthly so they do not have all of the newest phrases included.

Method 2 using *msgmerge* is probably the simplest method for user translation.

To use the msgmerge command after creating a new cin.po as suggested previously:

```
cd /{your top level cinelerra directory}
./po/xlat.sh > /tmp/cin.po   # use /tmp as a temporary place
cp /{your cinelerra directory}/po/xx.po /tmp/xx.po # substitute your language for x
msgmerge -U /tmp/xx.po /tmp/cin.po # xx.po will be overwritten to include updates
```

Then use any editor or poedit to provide messages/words translations in the new xx.po file.

Method 3 using *xlat.C* program is the most versatile with a variety of features. When a non-existent language translation is first set up, you would want to use this methodology to get started.

To use the *xlat.C* program, first compile it with `c++ xlat.C`. You can see the usage help here:

```
  ./a.out
list csv ./a.out csv < data.csv > data.po
list po ./a.out po < data.po > data.csv
list po ./a.out dups < data.po
list po ./a.out nodups < data.po
get strings ./a.out key < xgettext.po
gen xlation ./a.out xlat xgettext.po xlat.csv
gen xlation ./a.out xlat -text,xlat ... < xgettext.po
```

This program has 6 commands where the desired command is the first parameter to a.out.:

1. *csv* = comma separated value file; to recreate a po from a csv file.
2. *po* = convert po to csv data; for example, what you need to convert ru.po to ru.csv. You can even open the resulting *.csv in LibrOffice and update the *key* → *value* replacements (fields separated by "," only and check quoted fields as text during import). These results can be *Saved As* a csv file, and then used by xlat.C to reformulate a *po*.
3. *dups* = list only key/value items where either the *key=value* or *value=*" ".
4. *nodups* = list only key/value items where *key≠value* and *value≠*" ".
5. *key* = list cin.po key set.
6. *xlat* = overlay translation. This is the most important use and is described next.

The xlat command line parameters specify a new cin.po template, usually created with xlat.sh, and a list of key/value files which are used to build a mapping for the desired translation. The mapping files are added to the mapping in the order they appear on the command line, and any existing key is replaced with the newest definition; so typically the newest key/value data is last in the command parameters. Once the mapping is built, the first parameter which is the new po template is scanned and the keys it contains are used to find the latest mapping in the key/value files. The new value replaces the existing value in the template. For example, to overlay a new map onto an existing po using the Spanish es.po file and where cin.po represents the latest english words:

```
c++ xlat.C
./a.out po < es.po > /tmp/es.csv
./a.out po < cin.po > /tmp/cin.csv
./a.out xlat cin.po /tmp/es.csv > /tmp/new_es.po
```

The first run preserves the existing mapping of `es.po`, the second creates new mappings from `cin.po`, and the third merges the original and new mappings to create a po with new included/overriding `es.po`.

NOTE: some words and abbreviations can lead to ambiguous language translations. Therefore, the usage of C_ and D_ in the program code was added to represent Contextual and Definitional exceptions to the usual _ and N_ . You will see the following:

`C_(‘‘msgstr’’)` is translated to `D_(‘‘qual#msgstr’’)` by *xlat.sh*, and invokes gettext with `msgid = ‘‘qual#msgstr’’`

When no *po* translation is supplied, the qual# is removed, and only the default *msgstr* text is displayed. If a *po* translation is defined for the current locale, then the translated `msgid = ‘‘qual#msgstr’’` is used to access the translated msgstr.

The default MSGQUAL is the basename of the C source file. For the file `src_file.C`, the default MSGQUAL is:

```
# define MSGQUAL ‘‘src_file’’
```

It is used to define the qualifier needed to transform:

`C_(‘‘str’’)` to `D_(‘‘src_file#str’’)`

The resulting xlat.sh'd source is scanned by xgettext to create the initial cin.po. In other words:

_(s) does normal international text translation as always -

The msgid line is: msgid “s”

`C_(s)` makes it appear as if you used `D_(‘‘src_file#’’ s)` -

The msgid line is: `msgid ‘‘src_file#s’’` If it does not translate, the default msgtext is `‘‘s’’`, not `‘‘src_file#s’’`

`D_(qual#s)` tries to look up `_(qual#s)` -

This msgid line is: `msgid ‘‘qual#s’’`, If it does not translate, the default msgtext is `‘‘s’’`, not `‘‘qual#s’’`

Chapter 21

Licenses

CINELERRA-GG codebase is licensed GPLv2+

GREYcstoration plugin is licensed CeCILL v2.0

Google SHA1 implementation is licensed BSD 3-clause

Cinfinity icons are licensed Creative Commons CC-BY 4.0

Neophyte theme is licensed Creative Commons CC-BY 4.0

OpenCV is licensed 3-clause BSD License

freeverb components and the Tapeworm font are in the public Domain

License: GPLv2+ and CeCILL and BSD and CC-BY and Public Domain

GPLv2+ GNU General Public License version 2

CINELERRA-GG codebase is licensed GPLv2+

See: https://www.gnu.org/licenses/old-licenses/gpl-2.0-standalone.html

Creative Commons Attribution 4.0 International License

Cinfinity icons (c) by "Sam"; Neophyte and Cakewalk themes by "Olaf"; Media for Real Work multi-cam workflow by "Armandux" are licensed under a Creative Commons Attribution 4.0 International License.

see: http://creativecommons.org/licenses/by/4.0/

License Agreement for OpenCV

For Open Source Computer Vision Library

(3-clause BSD License)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the names of the copyright holders nor the names of the contributors may be used to endorse or promote products derived from this software without specific prior written permission.

This software is provided by the copyright holders and contributors *as is* and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall copyright holders or contributors be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this software, even if advised of the possibility of such damage.

Chapter 22

Authors and Contributors

Adam Williams - wrote the original complete Cinelerra package Copyrighted under GPL and continues to release new versions yearly; this is the HV version <broadcast@earthling.net>
.

Sam - designed and engineered the CINELERRA-GG website and continuously maintains and updates the Operating system and applications.

Willam Morrow - was responsible for the current CINELERRA-GG software version which emphasized crash-free stability; he started adding fixes and features in 2007, merged his version based on the HV version with the CV version in 2016, continued to add new features and fixes and incorporate desirable updates from HV and CV until November 2020.

Software developers working on CINELERRA-GG since at least 2021 along with some areas where they have been crucial (in alphabetical order by first name):

- Andrew Randrianasulu - continuous ongoing support; bug fixes; library upgrades with relevant patches; some nice enhancements; created and debugged builds: NetBSD and the Android tablet using Termux.
- Andras Reuss - created very useful Autosave backup feature in Preferences with Andrew assist.
- Georgy Salnikov - added all of the Help software in the program allowing access to HTML manual pages; major improvement in the Motion plugin; provided scripts and program mods to convert Latex manual to HTML; Blend Algebra and Blend Program plugins major enhancement.
- Manuel Virgil - DPX list writer developer; EXR package upgrade to 3.1.4 (not currently in use).
- Mat Nieuwenhoven - creator of the AppImage build procedure; LV2 specific control patch; maintains CINELERRA-GG wikipaedia entry.

Personnel providing current recent testing, examples, demos, and work on the manual or distribution packages for the CINELERRA-GG version (in alphabetical order by first name):

- Andrea Paz - main person who keeps the Manual updated; does builds for early testing of changes; provides examples and demos.
- Andrey Spitsyn - responsible for the huge task of getting the CINELERRA-GG Manual in LaTex. And for the current build farm at https://github.com/einhander/cin-gg-packages/releases.
- DeJay - expert user and frequent responder to users in the Forum.
- Glen MacArthur - provides A/V Linux distro which includes CINELERRA-GG.
- Pierre Autourduglobe - designed and thoroughly tested Interview Mode for implementation; major tester of Mixers; provided reason for getting the Shuttle to work; tested other new features.
- Phyllis Smith - facilitator of checking into GIT the developers and contributors changes to the Programs and to the Manual; tester; creates monthly AppImages when needed.
- RafaMar - updated Spanish translations; instrumental in getting friend's help in adding the Help button to the Batch Render menu; provided additional ShapeWipe transitions.
- Terje Hanssen - especially tests Bluray video media creation providing invaluable feedback; created a detailed PDF document on Preserving Camcorder Media; major building/testing of QSV/OneVPL on Intel hardware.
- + **many others**, some of who wish to remain anonymous.

Software Developers in the earlier years of CINELERRA-GG (alphabetical by last name):

- Balakhnin, Alexander - author of the Descratch plugin.
- Messick, Eric - provided the initial Shuttle code (FixedImagePhoto.com/Contact).
- Olaf - Cakewalk and Neophyte theme developer - Cakewalk is the default theme due to its being modern; tested and provided much feedback.
- Roenitz, Frederic - wrote the VP8/VP9 render formats.

Contributors and Software Developers through the years either on HV or CV, the Community Version. They are listed by last name in alphabetical order, along with some of their specific expert work - often just a single known item listed, and it may not even have been the most important when they obviously did so much more. The CV developers were responsible for many improvements based on the HV version to include bugfixing, creating build systems, internationalization work, plugins, and IIRC ffmpeg support appeared in one of the CV forks.

- Arendt, David - early tester providing significant feedback.
- Basto, EaSergio M. - completely replaced/fixed autogen.sh; removed old outdated fonts.
- Baverstock, Richard - assisted with working on the SVG plugin; added tape controls in the capture monitor window; major early tester and contributor.
- Bielefeldt, Karl - created gradient wipe plugin; worked on outdated mpeg2enc.
- Brosius, Kevin - provided many patches, for example fixed scroll bar resizing in bclistbox.C; major early contributor.
- Cornet, Jerome - updated Deinterlace plugin for temporal field swap.
- Coulon, Jean-Luc - updated French translation; provided patches for translation gettextization strings.
- Delannoy, Florent - C41 plugin to convert a 16 bit C41 digital intermediate negative file to positive image.
- Diniz, Rafael - worked on the Brazillian Portuguese translation; created Slackware packages.
- Dumuid, Pierre - assisted in merge efforts; added column in File Load dialog box for sorting by extension; wrote Selective Temporal Averaging plugin.
- Erdelyi, Georgely - early on provided patches for libsndfile, titler, and other programs to compile on Suse.
- Ferralis, Nicola - updated Ubuntu PPA over many years; along with Slock Ruddy, responsible for GreyCStoration plugin (currently inactive in CINELERRA-GG), worked on Italian translation.
- Ferrer, Alex - originally put up a Cinelerra manual TWiki; added numerous screenshots / additional topics and maintained the wiki.
- Flaming, Benjamin - created the Diffkey plugin.
- Geddes, Tyler - contributed to CV Version.
- Harang, Camille - worked on fixing guicast programs for big Endian; part of small team of 3 that used Cinelerra to make short documentaries/interviews.
- Heen, Tollef Fog - contributed to CV Version.
- Iniguez, Gustavo - contributed to HV version.
- Joyeux, Sylvain - worked on the UI; worked on configuration/build scripts.
- Kielb, Andreas - created Firewire I/O patches.
- Konink, Stefan de - added patch for playing without sound.

- Kurz, Nathan - added a YUV4MPEG pipe output dialog to the user interface; yuvstream.C program developer; early contributor.
- Maufrais, Nicolas - combined Adam's original Secrets of Cinelerra manual with Alex Ferrer's Wiki into a single CV manual. He was responsible for maintenance along with many others.
- Mekkes, Greg - added patch to Motion Blur plugin.
- Menk, Michael Eric - contributed to CV Version.
- Montgomery, Monty - Blue Banana color modification plugin; fileffmpeg developer in CV fork.
- Mueller, Johannes - author of keyboard tracks up/down shortcut and horizontal scrolling with the mouse wheel.
- Muylkens, Koen - created the Blue Dot theme.
- Olson, Eric - added YUV411 plugin to improve chroma upsampling results for DV source.
- Pesegov, Valentin - added Russian translation.
- Poure, Jean-Michel - provided fixes in build procedures for Debian; first entry in Mailing List September 2004.
- Rampino, Paolo - aka Akirad, created several colorful/artistic themes; developer of the cinecutie fork (now inactive); worked on Italian translation.
- Reinholdtsen, Petter - assisted in early hardware upgrade; core developer; moved over new features from HV to CV; fixed many, many bugs.
- Robak, Herman - created initial draft repositories; major early contributor.
- Runkaru, Einar - numerous bug fixes; for a long time maintaned a separate branch and currently has his own personal CVE fork; noted first entry in Mailing List in January 2010.
- Schmarsel, Danny - past website designer and administrator for Cinelerra-CV website; created Cinelerra LiveDVD for Slackware (now inactive); provided German translation.
- Seigne, Eric - initial French translation.
- Setti, Riccardo - contributed to CV Version.
- **Sixt, Johannes - long time contributor from early on and continuous**; notable for merging into Cinelerra-CV the new releases of HV.
- Stewart, Joe - worked on Inkscape for Cinelerra; created Motion Blur plugin.

- Streetman, Dan - developed patch to correct text_to_seconds; added patch to Titler for timecode functionality; patches to ReframeRT.
- Taraba, Mark - provided support for all types of PNGs; first Mailing List entry in August 2004.
- Thaeter, Christian - maintained server to host Cinelerra-CV website.
- Tori, Andraz - responsible for Cinelerra x86 Debian packages; worked on an SVG plugin; added overlaying speedup patch; major early contributor and lots of debugging.
- Traniello, Raffaella - best known for "Cinelerra for Grandma" still active at http://g-raffa.eu/Cinelerra/HOWTO/basics.html and helped to maintain the CV manual.
- Vladimirsky, Igor - was igor_ubuntu at one time; developer of the Russian po file; currently supports the Cinelerra-CV site; and much, much more work; noted first entry in Mailing List 2014.
- Vosseler, Hermann - developer of the Reroute plugin used to transfer the Alpha channel and/or the RGB/YUV Components from source track to target track; significant Bezier automation patch for fades, camera and projector.
- Wulff, Jonas - patched Histogram plugin to provide a split output screen.

- **+ many others**. We will gladly add anyone who has been inadvertently left out and their contribution to this list upon notification.

Appendices

Appendix A

QuickStart

A.1 Cinelerra-GG Quick Start Guide

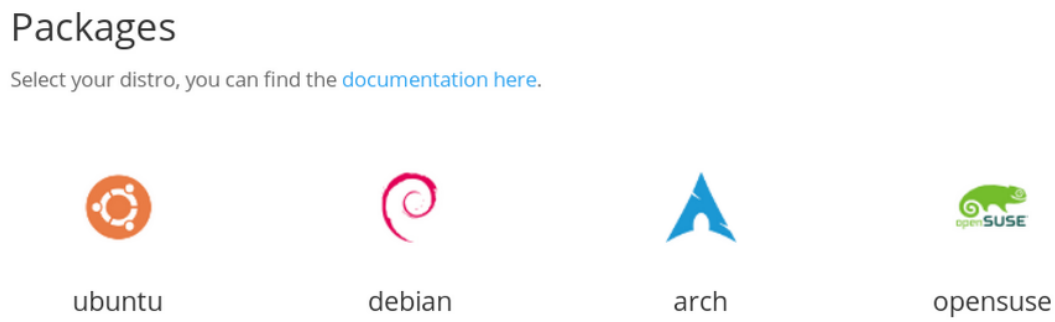
Cinelerra-GG is a software program NLE, Non-Linear Editor, that provides a way to edit, record, and play audio or video media on Linux. It can also be used to color correction, retouch photos, motion tracking, watch TV, and create DVDs.

A.1.1 Install the Software

On the internet, click on the Download page at:

`https://cinelerra-gg.org/downloads/`

Here you will see several Operating System distro packages that are already built for you to download. Click on your preference and read the specific instructions for usage.



However, if you want to get going as quickly as possible, just do this so that everything is in 1 place:

- Download your Operating System's tar file from `https://cinelerra-gg.org/download/tars/` to `/tmp`.
- Key in: cd /name-of-directory-where-you-want-the-software (for example, `cd /software`)
- Key in: `mkdir cin`

- Key in: `cd cin`
- Key in: `tar -xJf /tmp/cinelerra-5.1-*.txz` (if you put the tar in `/tmp` AND replace * with full name)

A.1.2 Start CINELERRA-GG

Depending on how you installed the software, you can log in as root or as a user if you used a package.

- Key in: `/your-software-directory-path/bin/cin`
- Or if you installed using the pkg method, click on the *Cin icon*.

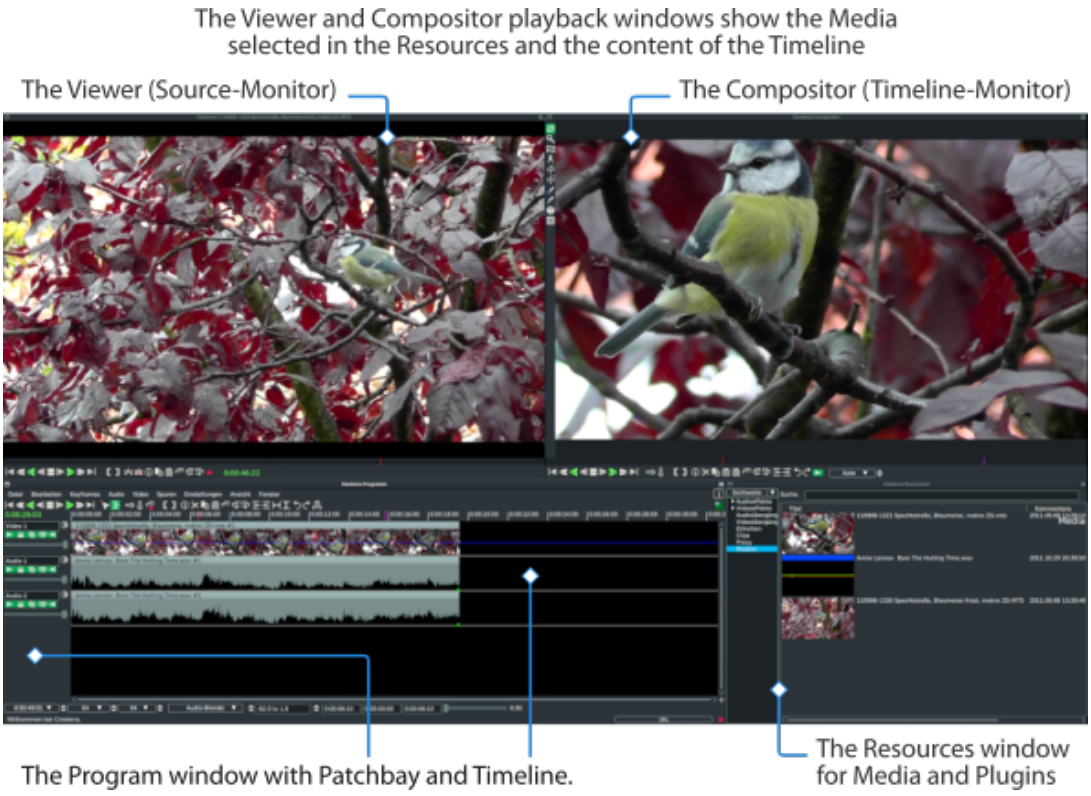


Figure A.1: *Clockwise: Viewer; Compositor; Resources and Main/Program/Timeline*

You will now see 4 separate windows appear. The top 2 windows from left to right are the Viewer which is most useful for previewing clips and media and the Compositor which displays the current working frame at the timeline position. The bottom 2 windows are the CINELERRA-GG Program, also called the timeline, which is where the real work gets done and the Resources window showing a selection of media or effects.

Any of these windows can be resized to better suit your needs. Note that if your system's native language is not English, some of the words you see on the screen will be correctly translated for you, others will be in english, and some will have not very good translations.

It is important to know that CINELERRA-GG does not directly change your media. It writes all changes to what is called the EDL, Edit Decision List. This way you original media remains completely intact.

Before you get startedi here is a note about *Context Help*. If you need more detailed information on a window, menu, button or other particular GUI element than is shown in the tooltip, press Alt/h hotkey and the HTML page in your configured web browser will display the documentation for the item currently under the mouse.

A.1.3 Load Media

On the main timeline program window are many pulldowns, the first of which is *File*.

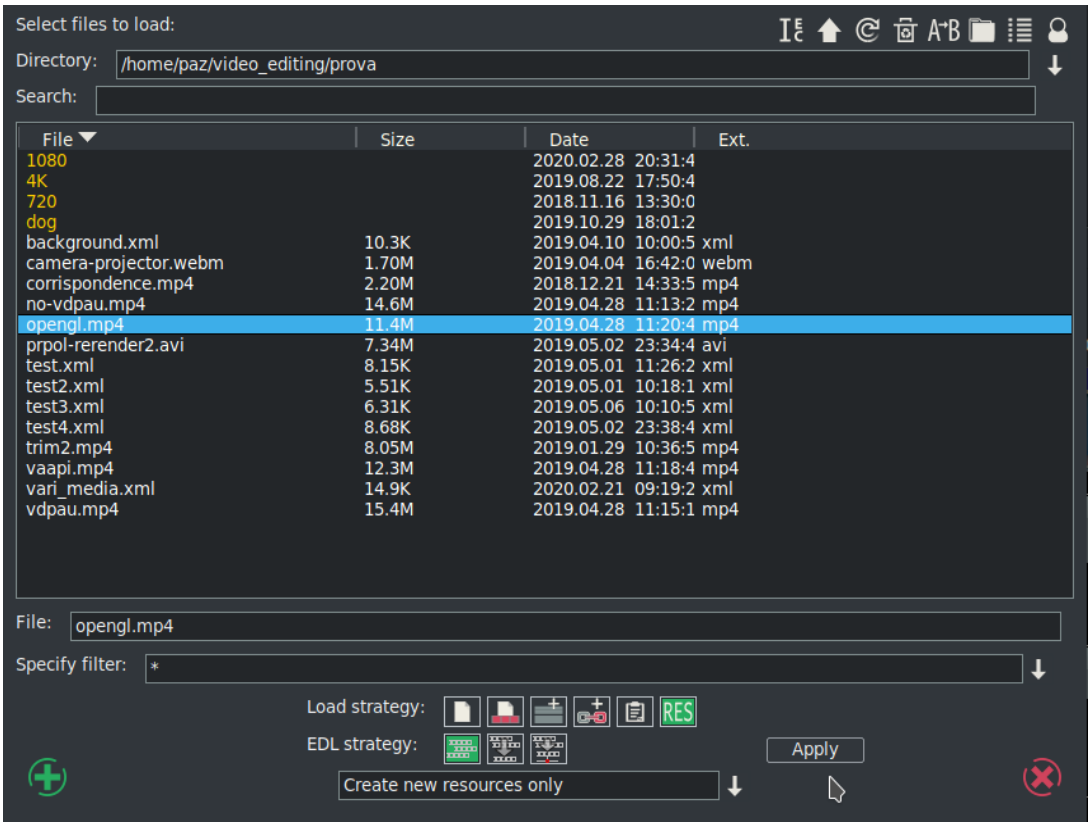


Figure A.2: *Load media window – note the icons top right for more options*

1. Click on *File* for a list of available options and note that in the right hand column are shortcuts for many of the options that will come in handy if you use CINELERRA-GG often.

2. Next click on the second one down – *Load files...* – which brings up the Load menu.

3. Below *Select files to load* on the top left side is a textbox and if you look all the way to the right side of the textbox, there is a down arrow which you use to navigate

your file system. Highlight the desired file system and you will see that directory name appear in the textbox and the files below.

4. Scroll to the media file you would like to work on and highlight that file. When you do, you will see that filename also appear in the textbox below the listing of files. You could have directly keyed in that file in that textbox instead.

5. On the bottom of the Load menu, is a box called *Insertion strategy*. For getting started the default of *Replace current project* is sufficient. But you can click on the down arrow to see what is available for future use.

6. Now click on the green colored checkmark on the bottom left hand side to actually load the file and see it appear on the timeline in CINELERRA-GG's main window and a single frame in the Compositor. The first track will most likely be video thumbnails and the next tracks may be audio waveforms.

7. Press the space bar in the main Program window and your video will start playing and press the space bar again to stop the play. While playing, you should see the video in the Compositor window in the upper right hand side of your screen and if you have your audio hooked up, you will hear the sound. To get back to the beginning of the video, hit the home key on your keyboard.

A.1.4 Choose Output Format

You can skip this step if you want the format of your output to be the same as your input. However, to create output media that is widely viewable on many platforms, to include phones and television, you should set your format accordingly.

1. On the main timeline, use the *Settings* pulldown (about the 7^{th} pulldown from the left side top) and click on *Format* which is the first option in that list.

2. A *Set Format* menu will appear that shows what the current format is for your loaded media in an Audio and a Video tab. In the United States, the Video Frame rate is usually expected to be 29.970 and usually the Color model is only changed if you have a personal preference.

3. The *Canvas size* is probably the only thing you will want to change here in order to get to the most commonly viewable settings. On the right hand side of the Width parameter is a down arrow. Left click the down arrow to see your options.

4. Highlight 1280 × 720 HD for a good common option.

5. Click OK to have this option take effect. When you do, the Compositor window may change to fit this option and may look wrong sized.

6. If the video now looks too small or too large in the Compositor, you will want to *autoscale* it to look correct when the new media is created. To do this, mouse over to

Figure A.3: *Format menu to change settings*

the Resources window in the lower right hand corner and under the word Visibility, highlight *Video Effects* to see some plugins.

7. Mouse over the *Auto Scale* icon, left click to highlight the words underneath the icon, and mouse drag the icon to the timeline video track. When you see a white colored outline show on that track, drop the Auto Scale icon there and you will see that the video may now automatically scale to a new value. Click on the magnifying glass icon on the brown colored line beneath the main timeline video which opens a new window. In that window, again use the down arrow to choose 1280 × 720 HD, then dismiss this window.

8. If not needed, to remove the Auto Scale plugin, right mouse on the brown line and choose *Detach*.

A.1.5 View and Listen

1. On the second line, below the pulldowns, are transport buttons to move back and forth on the timeline and play forward or reverse, fast or slow, or a single frame. When you mouse over one of these buttons, a colored tooltip appears to tell you its function along with a key shortcut inside of parenthesis. When you left click the mouse on the transport button it starts the play and click again to stop it. As you use these buttons, watch the Compositor to watch your video.

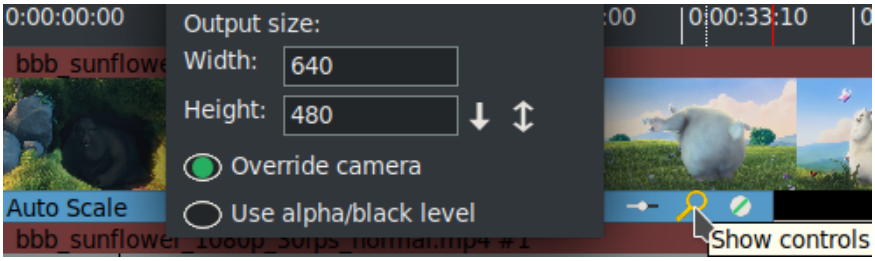


Figure A.4: *Effect blue bar with magnifier*

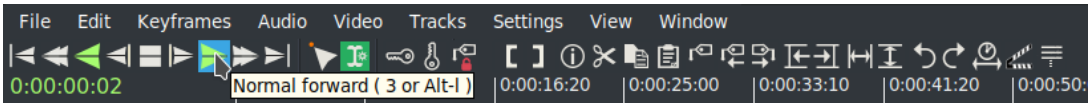


Figure A.5: *Menu pulldowns at the top with Transport buttons below. Note the colored tooltips too.*

2. On the timeline, you only see thumbnails and not every single picture. You may want to use your keyboard's *down arrow* to expand the thumbnails and the *up arrow* to unexpand them – on United States keyboard, the arrow keys are generally together on the lower right hand side of the keyboard, a little to the right of the space bar. This is a more cpu intensive operation and for very large video can be time-consuming.

If you need more detailed information on a button or other particular GUI element than is shown in the tooltip, press Alt/h hotkey and the HTML page in your configured web browser will display the documentation for the item currently under the mouse.

A.1.6 Edit/Compose

There may be sections of your media that you want to delete, or audio that is hard to hear and needs to be enhanced, or there is a need for a descriptive title that you want to add. Here are a few basics. But first be sure that you are in *cut and paste* mode (this is the default) by checking to verify that you see a gold color around the "I" i-beam mode icon as in the figure above. If the arrow to the left is gold, you are in *drag and drop* mode so switch to *cut and paste* by clicking on the "I" instead.

1. You should look at the *Edit* pulldown - 2^{nd} from the upper left on the main timeline to see the most common options to use. The first option in the list is `Undo` followed by a terse comment of the last operation that you performed that can be undone.

2. To delete a section of video/audio is described next. Various ways to do that are available but the easiest is to move your mouse and left click at the beginning of the section you want to delete on the timeline and while holding down the left mouse button, drag to the end of the section to be deleted. When you do this, a white

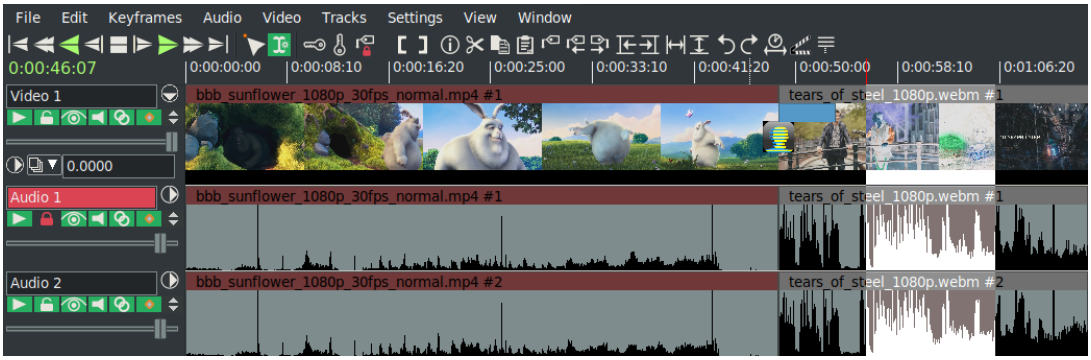


Figure A.6: *From left to right: Audio 1 is disarmed – BandSlide transition in Video 1 – A highlighted section.*

colored highlighted section becomes visible. Use the *Edit* pulldown and choose the *split/cut* option to cut out the highlighted area (note the shortcut of "x"). Remember if you cut the wrong thing out you can always use the Edit pulldown to Undo that.

3. To add a transition where there is deleted section which may make your video look disjointed, do the following. Go back to the Resources window in the bottom right hand corner. Change to *Video Transitions* by highlighting that underneath the word *Visibility*. Highlight a transition like *BandSlide* with the left button mouse click, hold down and drag to the video track and when you see a white colored box around the area that you deleted above, drop the icon. Right mouse click the icon on the track to vary some parameters like length.

4. To insert another clip from a different video, first you have to load the other video on another track. Go to *File* pulldown again and choose the *Load files* option. Type in a directory at the top again and erase any specific file that you may have chosen previously in the bottom 2 textboxes. It is very important to now change the Insertion strategy to *Append in new tracks* or you will write over your current work. But if you make this mistake, you can use the Edit pulldown and Undo that!

    a) Once the new video is on the track below your current work, you want to work with only this new track, so disarm your other tracks by looking to the left of each track's timeline and click the 2^{nd} button beneath the track name, for example Video 1 or Audio 1. The track name textbox will turn red to remind you that the track has been disarmed. The boxed area is called the patchbay.

    b) Move to the area you want to make a clip of on your newly loaded track, hold down the left mouse button and drag the area to be made into a clip which will turn the color white. Remember, you disarmed the other tracks so only this track is relevant at this time. On the second line of the main window to the right of the transport buttons, are action buttons and as you mouse over them a colored tooltip explains its purpose. Find the one that says *To clip* which is on the right hand side of the right bracket symbol.

    c) Click on *To clip* and a small window comes up which you can comment in, but you do not have to, so just click on the green checkmark and now you will

have a clip.

d) Disarm that new track and re-arm your original tracks so you can go back to working on them

e) Move your cursor to the spot in your original video where you want to insert the clip. Make a *Split* with the *Split | Cut* option.

f) Go to the Resources window and under the word Visibility, highlight *Clips* so you can see your recently created clip in the box to the right. Highlight that clip and drag it to where you did the blade cut and drop it in.

Figure A.7: *Compositor + Title menu for setting parameters + the Color Picker.*

5. To add a Title or any wording you will use the *Title* plugin. In the Resources window, under the word *Visibility*, highlight *Video Effects*. In the box to the right, many plugin icons appear. Scroll to the right using the scroll bar at the bottom of the Resources window to locate Title. Highlight the *Title* icon and drag/drop to your video track. By now your video track may be in sections as you deleted, added blade cuts, and inserts so where you drop the Title icon will be surrounded with a white colored box. It will take effect in that entire area so you may want to highlight a section as usual with left mouse click on the timeline and drag to the end of the desired area.

   a) Right click on the brown colored bar that appeared below your video track to get to options and then left click on *Show* to get the Title window to appear.

b) Now for the fun part. First type in some words in the bottom large text box just to see what it does. There are so many variable parameters here and they are a lot of fun to play around with.

c) You can dismiss the Title window when finished BUT be sure to leave the brown colored Title bar on the track. And if you enabled the *drag* feature, you should disable it so you do not forget.

d) Right mouse click on the bottom Text box to see many more interesting parameters.

A.1.7 Back up your work

At this time, or even earlier if you think you might make a mistake or if you are concerned about computer crashes, you should save your work. Use the *File* pulldown, and you can use *Save as* to designate a directory and filename. Then click the green checkmark. You are saving the EDL which is the set of changes that you have made – this file is separate from your original media.

A.1.8 Create your new media

1. Once again in the main Program window, click on the *File* pulldown and highlight/click the `Render` option which is about the 9^{th} option down from the top of the list. A CINELERRA-GG Render menu will appear.

2. First key in the first textbox the file to render to under *Select a file to render to*.

3. For the *File Format*, click on the down arrow and select FFMPEG (because this is the most commonly used format; later you may want to experiment with others). To the right of that box, click on the down arrow and highlight `mp4` – again because this is common. When you click on mp4, notice that if there is an extension to your filename in the *file to render to* above, it may change it to mp4 and if there is none, it will add `.mp4` because that is what is expected.

4. Make sure there is a red colored checkmark next to the words Audio and Video right below if you have/want both audio and video. To the left of that checkmark box, is a symbol that looks like a wrench. Click on this for Audio just to see the default Preset options which are just fine so dismiss the menu. Then click on the wrench for Video and check Pixels by using the down arrow to the right to be yuv420p – this is most commonly usable option. And click on the green checkmark.

5. Check the Insertion Strategy in the Render Menu window. You might want to change that to a different strategy than the default of *Append in new tracks*. If not, then when the Render is done, your new video will automatically be loaded in another set of tracks below your work tracks. Click on the green checkmark in the lower left corner to start the render.

6. As the render is running, you will see the video play by in the Compositor. Rendering is usually slow, especially with plugins added.

A.1.9 Play your new media

The file you created in the Render step should now be playable. You can test this in CINELERRA-GG most easily by going to the Resource window in the lower right corner, clicking on the Media folder, and dragging and dropping the last video to the Viewer window. There is a separate set of transport buttons on the bottom on that screen to use for playing.

A.2 Overview on Formats and Codecs

Here is an overview of the formats (also called containers) and codecs that are used in CINELERRA-GG, by ffmpeg and the internal engine. Roughly speaking these are divided into uncompressed codecs (or codecs with *Intraframe* compression, which can be lossy or lossless) and compressed codecs of *Interframe* type (LongGOP, almost always with lossy compression). The All-I (intraframe) codecs are suitable for editing because a cut or other operation on the timeline corresponds to the exact frame on which you are operating. The interframe types use Groups of Pictures (GOP) and a cut or other operation is accurate (and requires no further calculation) only if it coincides with the beginning of the GOP, and not with an internal frame. There is also color compression: Color Space *bit-depth* and *Chroma-Subsampling* for YUV models. In addition, heavy compression requires the system to do more encoding/decoding work on the timeline. High quality codecs have high bit rates and bit depths but this also affects the performance of the system, not to mention the increased disk space usage. Some formats implement both audio and video streams, others audio only or video only.

A.2.1 Video FFmpeg Formats

FFmpeg supports hundreds of codecs and formats. Some are proprietary and cannot be implemented in FFmpeg or can be voluntarily compiled as non-free; others are proprietary but their use is free; finally there are the Open formats/codecs, fully supported and well documented. We are only describing here a selection of the most well-known and most frequently used ones.

A.2.1.1 High Quality

High quality formats are also called Mezzanine codecs, Digital Intermediate, Preservation codecs or Editing codecs. These have no compression or intraframe lossless or near-lossless compression and are suitable for editing, post-processing, mastering and archiving. They are also used for the interchange of files between different programs. They take up a lot of disk space and require a powerful system.

MKV Open, highly configurable and extensively documented. Can have seeking problems. Belongs to the Matroska family.
Presets: *ffv1, ffvyuv*

MXF Created by Avid. It is probably the best and most advanced container for editing.
Presets: *DNxHR, ffv1, AVC_Intra_100*

MOV Created by Apple. It is a suitable format for editing because it organizes the files within the container into hierarchically structured *atoms* described in a header. This brings simplicity and compatibility with various software and does not require continuous encoding/decoding in the timeline.
Presets: *DNxHR, ffv1, CineformHD, huffyuv*

PRO Different extension, but it is still mov. Prores is proprietary and there are no official encoders except the original Apple one. The engine used by ffmpeg is the result of reverse engineering and, according to Apple, does not guarantee the same quality and performance of the original[1]. Option *vendor=apl0* is used to make it appear that the original Apple engine was used.
Presets: *ProRes; ProRes_ks*

QT Different extension, but it is always mov.
Presets: *DNxHD, magicyuv, raw, utvideo*

MP4 mostly used for General Purpose. It belongs to the large MPEG family.
Presets: *AVC_Intra_100*

RGB Raw format.
Presets: *raw*

YUV Raw format.
Presets: *raw*

AVI Old and limited format (no multi streams, no subtitles, limited metadata) but with high compatibility.
Presets: *ffv1*

A.2.1.2 General Purpose

These are also called Delivery codecs. They are the most used and widespread being suitable for streaming, video sharing, watching TV, smartphones, plus more. Because of lossy compression type Interframe, they produce smaller files with variable quality. They are not suitable for editing, compositing and color correction. Further rendering of these formats worsens the quality exponentially. The most used codecs have hardware support (vaapi, vdpau, nvenc) that make them more efficient.

MOV Created by Apple. It is a suitable format for editing because it organizes the files within the container into hierarchically structured "atoms" described in a header. This brings simplicity and compatibility with various software and does not require continuous encoding/decoding in the timeline.
Presets: *Presets: mov*

[1] https://support.apple.com/en-us/HT200321

QT Different exstension, but it is always mov.
Presets: *mjpeg, DV, Div, CinePack*

MP4 The most popular. Many other formats belong to this family (MPEG);
h264 is actually x264, open, highly configurable and documented; h265/HEVC is actually x265, open, highly configurable and documented. x264-5 is for encoding only.
Presets: *h265, h265, mjpeg, mpeg2, obs2youtube*

WEBM Open; similar to mp4 but not as widespread (it is used by YouTube). It belongs to the Matroska family. In CINELERRA-GG there are specific Presets with `.youtube` extension, but they are still webm. For VP9 and AV1 presets, two pass rendering is recommended, which provides higher quality.
Presets: *VP8, VP9, AV1*

MKV Open, highly configurable and widely documented. It might have seeking problems. It belongs to the Matroska family. For VP9 presets, two pass rendering is recommended, which provides higher quality.
Presets: *Theora, VP8, VP9*

AVI Old and limited format (no multistreams, no subtitles, limited metadata) but with high compatibility.
Presets: *asv, DV, mjpeg, xvid*

MPG Parent of the MPEG family, to which MP4 also belongs. Mpeg is used by CINELERRA-GG as default for proxies and mpeg-2 is the standard for Video DVDs.
Presets: *mpeg, mpeg2*

A.2.1.3 Note on Matroska (mkv) container

Matroska is a modern universal container that is Open Source so there is lots of ongoing development with community input along with excellent documentation. Also derived from this format is the *Webm* container used by Google and YouTube, which use the VP8-9 and AV1 codecs. Although using in CINELERRA-GG is highly recommended, you may have seeking problems during playback. The internal structure of matroskas is sophisticated but requires exact use of internal keyframes (I-frame; B-frame and P-frame) otherwise playback on the timeline may be subject to freeze and drop frames. The mkv format can be problematic if the source encoding is not done well by the program (for example, OBS Studio). For an easy but accurate introduction of codecs and how they work see: https://ottverse.com/i-p-b-frames-idr-keyframes-differences-usecases/.

To find out the keyframe type (I, P, B) of your media you can use ffprobe:

```
$ ffprobe -v error -hide_banner-of default=noprint_wrappers=0 -print_format flat -
    ↪ select_streams v:0 -show_entries frame=pict_type input.mkv
```

-v error -hide_banner: serves to hide a blob of information that is useless for our purposes.

-of: is an alias for *-print_format* and is used to be able to use *default=noprint_wrappers =0*.

-default=noprint_wrappers=0: is used to be able to show the information from the parsed stream that we need.

-print_format flat: is used to display the result of ffprobe according to a *flat* format (you can choose CSV, Json, xml, etc).

-select_streams v:0: is used to choose the first stream (0) in case there are multiple audio and video streams (tracks, in CINELERRA-GG).

-show_entries: shows the type of data collected by ffprobe that we want to display (there are also types: `_streams`, `_formats`, `_packets`, and `_frames`. They are called *specifiers*).

-frame=pict_type: within the chosen specifier indicates the data to be displayed; in this case *pict_type*, that is, the keyframe type (I, P, B) of the frame under consideration.

input.mkv: is the media to be analyzed (it can be any container and codec).

(see https://ffmpeg.org/ffprobe.html for more details)

We thus obtain a list of all frames in the analyzed media and their type. For example:

```
frames.frame.0.pict_type="I"
frames.frame.1.pict_type="P"
frames.frame.2.pict_type="B"
frames.frame.3.pict_type="B"
frames.frame.4.pict_type="B"
...
```

There are also 2 useful scripts that not only show the keyframe type but also show the GOP length of the media. They are zipped tars with readme's at:
https://cinelerra-gg.org/download/testing/getgop_byDanDennedy.tar.gz
https://cinelerra-gg.org/download/testing/iframe-probe_byUseSparingly.tar.gz

We can now look at the timeline of CINELERRA-GG to see the frames that give problems in playback. Using a codec of type Long GOP, it is probably the rare I-frames that give the freezes. To find a solution you can use MKVToolNix (https://mkvtoolnix.

download/) to correct and insert new keyframes into the mkv file (matroska talks about *cues data*). It can be done even without new encoding. Or you can use the `Transcode` tool within CINELERRA-GG because during transcoding new keyframes are created that should correct errors.

A.2.1.4 Image Sequences

The image sequences can be uncompressed, with lossy or lossless compression but always Intraframe. They are suitable for post-processing that is compositing (VFX) and color correction. Note: even if CINELERRA-GG outputs fp32, exr/tiff values there are normalized to 0-1.0f.

DPX Film standard; uncompressed; high quality. *Log* type.

PNG Uncompressed or lossless compression. Supports alpha channel.

WEBP, TIFF, GIF, JPEG, ... Variable compression, size and quality.

A.2.1.5 Old Pro Formats

Some formats, though used in the past in the pro field, are disappearing with the evolution of technologies. DVD is becoming more and more niche, while Bluray is still widespread (also as a backup); DV/HDV remains only as a support for old Camcorders with magnetic tapes. DV is still a quality format, with intraframe compression; HDV is mpeg-2 compressed.

AVI old and limited format but with high compatibility.
Presets: *DV_pal, DV_ntsc, mjpeg*

QT belongs to the Apple mov family.
Presets: *DV, mjpeg*

M2TS format for Bluray (mpeg4). Bluray player devices need a standard Bluray disc structure (bdwrite) for playback[2].
Presets: *AVC422, Lossless, Bluray, hevc*

MP4 Belongs to the MPEG family. Motionjpeg has jpeg compression, then Intraframe, so it maintains good quality and fluidity in editing. It is now an old and limited codec.
Presets: *mjpeg*

A.2.2 Audio FFmpeg Formats

Audio formats and codecs take much less resources and space than video ones, so they are often used without compression for maximum quality. However these are compressed formats and codecs widely used in streaming and sharing.

[2] CINELERRA-GG offers specific functionality for creating DVDs/Blurays

A.2.2.1 High Quality

FLAC Open; used for storing music. It has lossless compression.
preset: *flac*

PCM Raw format that encodes the signal with *modified pulse modulation* (pcm). FFmpeg does not support pcm audio if you use mp4 as a container.
Presets: *s8, s16, s24, s32*

WAV Raw format created by Microsoft. 32-bit addressing leading to the 4 GB recording limit. It is a widely used standard.
Presets: *s24le, s32le*

W64 Wave format created by Sony to override the 4GB recording limit. Poorly supported.
Presets: *s16le, s24le, s32le*

MKA Open, highly configurable and documented. It belongs to the Matroska family. Uncompressed pcm type.
Presets: *s16le, s24le, s32le*

ALAC Apple's codec, free to use but not open source. It is lossless and of high quality but is slower than other similar codecs.
Presets: *m4a, mkv, qt*

A.2.2.2 General Purpose

MP3 Belongs to the MPEG family. The most widely used in streaming and sharing.
preset: *mp3*

OGG Open, highly configurable and documented. It belongs to the Matroska family. Flac has lossless compression; opus is compressed but modern and of good quality, superior to mp3. Vorbis is compressed and dated, but lightweight and compatible.
Presets: *flac, opus, vorbis*

PRO Created by Apple; compressed audio codec, competing with mp3.
Presets: *aac256k*

A.2.3 CINELERRA-GG Internal Engine

FFmpeg is the default engine, but you can also use its internal engine, which is limited in supported formats but efficient and of high quality.

A.2.3.1 Video general purpose

RAW DV supports the DV standard.
Presets: *dv*

MPEG Video highly configurable. Extension `.m2v`.
Presets: *mpeg1, mpeg2*

OGG Theora/Vorbis Open, easily configurable. Theora for video, Vorbis for audio.
Presets: *theora, vorbis*

A.2.3.2 Image Sequences

There are quite a few formats available. Note: even if CINELERRA-GG outputs fp32, exr/tiff values there are normalized to 0-1.0f.

EXR Sequence OpenEXR (Open Standard) is a competing film standard to DPX, but *Linear* type.

Ppm Sequence is RGB Raw.

Tga Sequence is RGB(A) compressed or uncompressed.

Tiff Sequence is RGB(A) or RGB(A)-Float with various compression types.

Jpg, gif Sequences lossy compressed and limited formats.

A.2.3.3 Audio general purpose

AC3 widely used multichannel standard (Dolby Digital). Format with lossy compression.
Presets: *ac3*

Apple/SGI AIFF Created by Apple; is an uncompressed format (pcm type) or with 32/64-bit floating point compression.
Presets: *aif*

Sun/Next AU created by Sun and used in Unix environment, now in disuse. It can be of pcm type or with lossy compression.
Presets: *au*

Flac Open, lossless compression, very good quality.
preset: *flac*

Microsoft WAV created by Microsoft. It can have 16-24-32-bit linear or float compression.
Presets: *wav*

MPEG Audio Very widespread standard. Extension `.mp3`.
Presets: *mp3*

A.3 Overview on Color Management

CINELERRA-GG does not have support for ICC color profiles or global color management to standardize and facilitate the management of the various files with which it works. But it has its own way of managing color spaces and conversions; let's see how.

A.3.1 Color Space

A color space is a subspace of the absolute CIE XYZ color space that includes all possible, human-visible color coordinates (therefore makes human visual perception mathematically tractable). CIE XYZ is based on the RGB color model and consists of an infinite three-dimensional space but characterized (and limited) by the xyz coordinates of five particular points: the Black Point (pure black); the White Point (pure white); Reddest red color (pure red); Greenest green color (pure green); and Bluest blue color (pure blue). All these coordinates define an XYZ matrix. The color spaces are submatrices (minors) of the XYZ matrix. The absolute color space is device independent while the color subspaces are mapped to each individual device. For a more detailed introduction see: https://peteroupc.github.io/colorgen.html A color space consists of primaries (*gamut*), transfer function (*gamma*), and matrix coefficients (*scaler*).

Color primaries : the gamut of the color space associated with the media, sensor, or device (display, for example).

Transfer characteristic function : converts linear values to non-linear values (e.g. logarithmic). It is also called Gamma correction.

Color matrix function (scaler): converts from one color model to another. $RGB \leftrightarrow YUV$; $RGB \leftrightarrow Y'CbCr$; etc.

The camera sensors are always RGB and linear. Generally, those values get converted to YUV in the files that are produced, because it is a more efficient format thanks to chroma subsampling, and produces smaller files (even if of lower quality, i.e. you lose part of the colors data). The conversion is nonlinear and so it concerns the "transfer characteristic" or gamma. The encoder gets input YUV and compresses that. It stores the transfer function as metadata if provided.

A.3.2 HDR

Before 2016, HDR was understood as values outside the range (0 - 1.0f). Since there were no devices capable of achieving these values the only way available was to create digital images (in Blender, for example) or to merge multiple images with various dynamic ranges into a single HDR image. In 2016, with the arrival of sensors and monitors capable of extended dynamic ranges, HDR was standardized and there is the BT. 2100 color space that has gamut and color model like BT. 2020 but different transfer functions (gamma): Dolby Vision (HDR10), PQ and HLG. BT. 2100 considers values above 1.0f as illegal. There are two ways to return to *legal* values: clipping and tone-mapping. The BT.2100 color space is an efficient standard for encoding HDR colors, but it's not well suited for many rendering and composition (blending) operations.

CINELERRA-GG runs internally in 32-bit floating-point and thus can read and process values outside the range $(0-1.0f)$. However, there are plugins and tools that can introduce clipping that must be taken into account to keep the whole pipeline which preserves the original values. The manual indicates which plugins limit pixel values in the range $(0, 1.0f)$. The easiest way to see if a media is out of $(0-1.0f)$ range is to use the `eyedropper` tool (Get Color) in the Compositor window. If we use HDR media and need to do a Tone Mapping to make the pixel values fall within the range $(0, 1.0)$, we can use the `Blender Program` plugin and, inside it, load the function (found in `=>Syslib`) *tone_mapping.bp*. Instructions for using this function can be found here at example 6.

A.3.3 CMS

A color management system (CMS) describes how it translates the colors of images/videos from their current color space to the color space of the other devices, i.e. monitors. The basic problem is to be able to display the same colors in every device we use for editing and every device on which our work will be viewed. The color management operation consists of a color space conversion (matrix and 1DLUT). Colors that exceed the display's target color gamut are numerically clipped. Calibrating and keeping our hardware under control is feasible, but when viewed on the internet or DVD, etc. it will be impossible to maintain the same colors. The most we can hope for is that there are not too many or too bad alterations. But if the basis that we have set up is consistent, the alterations should be acceptable because they do not result from the sum of more issues at each step. There are two types of color management: *Display referred* (DRC) and *Scene referred* (SRC). Display-refered: 1.0f (white point) as the reference white level of the display. Scene-refered: 1.0f as the reference white of the used Color Space (nominal)

- **DRC** is based on having a calibrated monitor. What it displays is considered correct and becomes the basis of our color grading. The goal is that the colors of the final render will not change too much when displayed in other hardware/contexts. Be careful to make sure there is a color profile for each type of color space you choose for your monitor. If the work is to be viewed on the internet, be sure to set the monitor in *sRGB* with its color profile. If for HDTV we have to set the monitor in *rec.709* with its color profile; for 4k in *Rec 2020*; for Cinema in *DCP-P3*; etc.
- **SRC** instead uses three steps:
  1. The input color space: whatever it is, it can be converted manually or automatically to a color space of your choice.
  2. The color space of the timeline: we can choose and set the color space on which to work.

3. The color space of the output: we can choose the color space of the output (on other monitors or of the final rendering).

ACES and *OpenColorIO* have an SRC workflow. Please note that the monitor must still be calibrated to avoid unwanted color shifts.

- There is also a third type of CMS: the one through the **LUTs**. In practice, the SRC workflow is followed through the appropriate 3D LUTs, instead of relying on the internal (automatic) management of the program. The LUT combined with the camera used to display it correctly in the timeline and the LUT for the final output. Using LUTs, however, always involves preparation, selection of the appropriate LUT and post-correction. Also, as they are fixed conversion tables, they can always result in clipping and banding.

A.3.4 Display

Not having CINELERRA-GG a CMS, it becomes essential to have a monitor calibrated and set in sRGB that is just the output displayed on the timeline of the program. You have these cases:

Timeline	Display	Description
sRGB	sRGB	we get a correct color reproduction
sRGB	Rec.709	we get slightly dark colors, because gamma
sRGB	DCI-P3	we get over-saturated dark colors, because gamma and bigger gamut

A.3.5 Pipeline CMS

INPUT → DECODING/PROCESSING → OUTPUT/PLAYBACK → DISPLAY → ENCODING

Input color space and color depth of the source file; better if combined with an ICC profile.

Decoding how CINELERRA-GG transforms and uses the input file (it is a temporary transformation, for usage of the internal/ffmpeg engine and plugins).

Output our setting of the project for the output. In CINELERRA-GG such a signal is `8-bit sRGB`, but it can also be 8-bit YUV in *continuous playback*.

Display as the monitor equipped with its color space (and profiled with ICC or LUT) displays the signal that reaches the user and what we see. The signal reaching the display is also mediated by the graphics card and the operating system CMS, if any.

Encoding the final rendering stage where we set not only formats and codecs but also color space, color depth, and color range.

A.3.6 How CINELERRA-GG works

Decoding/playback: Video is decoded to internal representation (look at `Settings /Format/Color model`). Internal format is unpacked as 3 color values + one alpha value every pixel. CINELERRA-GG has 6 internal pixel formats (RGB(A) 8-bit; YUV(A) 8-bit and RGB(A)_FLOAT 32-bit (see Color Model in Video attributes). The program will configure the frame buffer for your resulting video to be able to hold data in that color model. Then, for each plugin, it will pick the variant of the algorithm coded for that model. CINELERRA-GG automatically converts the source file to the set color model (in a buffer, the original is not touched!). Even if the input color model matches what we set in `Settings/Format/Color model`, there will always be a first conversion because CINELERRA-GG works internally (in the buffer) at 32-bit in RGB. For playback CINELERRA-GG has to convert each frame to the format acceptable by the output device, i.e. sRGB 8-bit. In practice, the decoded file follows two separate paths: conversion to FLOAT for all internal calculations in the temporary (including other conversions for plugins, etc.) and simultaneously the result in the temporary is converted to 8-bit sRGB for on-screen display. See also Automatic "Best Model" Media Load. To review, a *temporary* is a single frame of video in memory where graphics processing takes place. CINELERRA-GG use X11 and X11 is RGB only and it is used to draw the *refresh frame*. So single step is always drawn in RGB. Continuous playback on the other hand can also be YUV for efficiency reasons.

Color range: One problem with the YUV color model is the `YUV color range`. This can create a visible effect of a switch in color in the Compositor, usually shown as grayish versus over-bright. The cause of the issue is that X11 is RGB only and it is used to draw the *refresh frame*. So single step is always drawn in RGB. To make a YUV frame into RGB, a color model transfer function is used. The math equations are based on Color_space and Color_range. In this case, color_range is the cause of the *grayish* offset. The *YUV MPEG color range* (limited or TV) is 16..235 for **Y**, 16..240 for **UV**, and the color range used by *YUV JPEG color range* (full or HDTV) is 0 to 255. The cause is that 16-16-16 is seen as pure black in MPEG, but as gray in JPEG and all playback will come out brighter and more grayish. This can be fixed by forcing appropriate conversions via the ColorSpace plugin. See Color Space and Color Range Affecting Playback

Plugins: On the timeline all plugins see the frames only in internal pixel format and modify this as needed (*temporary*). Some effects work differently depending on colorspace: sometimes pixel values are converted to float, some-

times to 8-bit for an effect. In addition *playback single step* and *plugins* cause the render to be in the session color model, while *continuous playback* with no plugins tries to use the file's best color model for the display (for speed). As mentioned, each plugin we add converts and uses the color information in its own way. Some limit the gamut and depth of color by clipping (i.e. `Histogram`); others convert and reconvert color spaces for their convenience; others introduce artifacts and posterization; etc. For example, the `Chroma Key (HSV)` plugin converts any signal to HSV for its operation. If we want to better control and target this color management in CINELERRA-GG, we can take advantage of its internal ffmpeg engine: there is an optional feature that can be used via `.opts` lines from the ffmpeg decoded files. This is via the `video_filter=colormatrix=...`ffmpeg plugin. There may be other good plugins (lut3d...) that can also accomplish a desired color transform. This `.opts` feature affects the file colorspace on a file by file basis, although in principle it should be possible to setup a `histogram` plugin or any of the `F_lut*` plugins to remap the colortable, either by table or interpolation.

Conversion: Any conversion is done with approximate mathematical calculations and always involves a loss of data, more or less visible, because you always have to interpolate an exact value when mapping it into the other color space. Obviously, when we use floating point numbers to represent values, these losses become small and close to negligible. So the choice comes down to either keeping the source color model even while processing or else converting to FLOAT, which in addition to leading to fewer errors should also minimize the number of conversions, being congruous with the program's internal one. The use of FLOAT, however, takes more system resources than the streamlined YUV. Color conversions are mathematical operations; for example to make a YUV frame into RGB, a color model matrix function is used. The math equations are based on color_space and color_range. Since the majority of sources are YUV, this conversion is very common and it is important to set these parameters to optimize playback speed and correct color representation.

Encoding: Finally, the encoding converts to colorspace required by the codec.

A.3.7 Workflow

Let us give an example of color workflow in CINELERRA-GG. We start with a source of type YUV (probably: YCbCr); this is decoded and converted to the chosen color model for the project, resulting in a *temporary*. Various jobs and conversions are done in FLOAT math and the result remains in the chosen color model until further action. In addition, the temporary is always converted to sRGB 8-bit for monitor display only. If we apply the `ChromaKey (HSV)` plugin, the temporary is con-

verted to HSV (in FLOAT math) and the result in the temporary becomes HSV. If we do other jobs the temporary is again converted to the set color model (or others if there is demand) to perform the other actions. At the end of all jobs, the obtained temporary will be the basis of the rendering that will be implemented according to the choice of codecs in the render window (*Wrench*), regardless of the color model set in the project. If we have worked well the final temporary will retain as much of the source color data as possible and will be a good basis for encoding of whatever type it is.

For practical guidelines, one can imagine starting with a quality file, for example, *10-bit YUV 4.2.2*. You set the project to `RGBA-FLOAT`; the `YUV color space` to your choice of Rec709 (for a FullHD) or BT 2020NCL (for UHD) and finally the `YUV color range` to JPEG. If the original file has the MPEG type color range then you convert to JPEG with the `ColorSpace` plugin. If you want to transcode to a quality intermediate you can use *DNxHR 422*, or even *444*, and maybe do the editing step with a *proxy*. For rendering you choose the codec appropriate for the file destination, but you can still generate a high-quality master, for example *ffv1 .mov* with lossless compression.

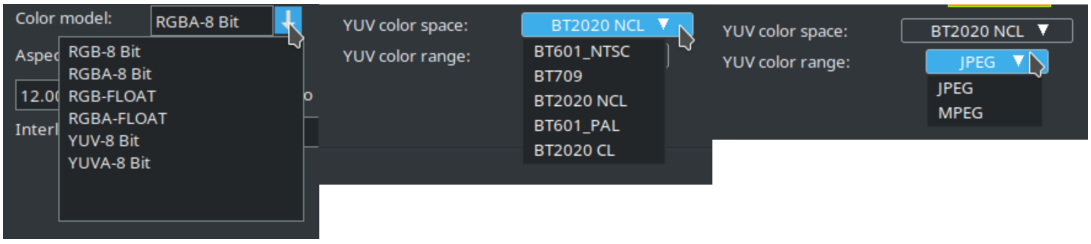


Figure A.8: *Color settings (Settings → Format / Settings → Preferences)*

APPENDIX B

Developer's Section

B.1 How Builds Really Work and Even More Options

This section describes how builds really work if you want to know more about making changes or understanding the process; probably only for a developer or system administrator.

Builds occur in 4 basic steps:

1. unpack/patch source code
2. configure build
3. make build targets
4. installation

So, an example of what happens in 4 steps for a single-user build would be as follows:

1. unpack/patch source code:
   ```
   git clone --depth 1 ‘‘git:/.../target’’ cinelerra5
   ./autogen.sh
   ```
2. configure build:
   ```
   ./configure -with-single-user
   ```
3. make build targets:
   ```
   make 2 > &1 | tee log
   ```
4. installation:
   ```
   make install
   ```

A lot of things can be tweaked to change the results. Mostly these changes are parameters to the configure step, which can change important build related items, like the application name, or where and what the target system directories should be. This makes it possible to have several versions at the same time on the same computer if needed. To see what it is that the makefiles use to build CINELERRA-GG, look at the resulting top-level global_config file which is created by the ./configure step.

Building CINELERRA-GG requires many thirdparty libraries, and it is recommended that you use the static build version included in the git repo. Some of them are patched, and fix significant bugs. It is important to note that because system installation historically has been with as many shared objects as possible, the defaults are that any system library detected during configuration setup will be used, when the package is built *--without-single-user*, which is the default build. To build with static thirdparty libraries for system install to the system /usr area, use:

```
.configure --enable-static-build -prefix=/usr
```

Sometimes, additional package parameters and variables are needed during thirdparty builds. These optional values occur before and after the *configure* and *make* commands during a build. A presentation of the format of the package qualified variable names and how they appear in the build procedure are:

pkg.cfg_vars	prepended to configure
pkg.cfg_params	appended to configure
pkg.mak_vars	prepended to make
pkg.mak_params	appended to make
pkg.cflags	added as CFLAGS+=$(cflags) to pkg.vars
pkg.cppflags	added as CPPFLAGS+=$(cppflags) to pkg.vars

These steps are done for EACH of the packages in the thirdparty build:

```
<pkg.cfg_vars> ./configure <pkg.cfg_params>

<pkg.mak_vars> make <pkg.mak_params>
```

The thirdparty Makefile has a set of default vars and params used to build each of the needed thirdparty packages, but you can specify new or overriding values for these Makefile substitutions. These thirdparty build config changes are specified in the top-level file: *cin_config*. By using this file, you can save the configuration changes made for the current build to use the next time you do a new build. For example, to add an include file path to the giflib build, add this line to *cin_config*:

```
giflib.cflags := -I/usr/local/include/giflib5
```

To have a param/var change apply to all thirdparty builds, use:

```
CFG_VARS, CFG_PARAMS, MAK_VARS, MAK_PARAMS
```

CFLAGS, CXXFLAGS and LDFLAGS are forwarded to the thirdparty build environ-

ment via:

```
CFLAGS=-ggdb ./configure --with-single-user
```

However, there is no guarantee that the thirdparty build will honor the environmental flags.

Finally, there are build controls, which enable/disable and set build features.

A few of the more useful ./configure --parameters are:

--with-jobs=n	where n=number of build jobs; defaults to 1.5*cpus+2
--enable-static-build	build all 3rd party libs; defaults to yes if single-user, else no
--with-single-user	build installs to <build_path>/bin; no system installation

The ./configure command builds *global_config*. The global_config is read by the thirdparty/Makefile to create the recipes and definitions used by the thirdparty build.

There are a lot of different options. Thirdparty library build control is available in the configure step of the build. Thirdparty libraries are built on a demand basis. So if you use:

--enable-libname=auto	the static-build enable, or the lack of a system library causes a build
--enable-libname=yes	this forces the thirdparty build
--enable-libname=no	this forces no thirdparty build

FFmpeg is a *strongly connected* component in the build linkage and widely influences the CINELERRA-GG library demands. It is possible to make small additions to the ffmpeg configuration step using the environment variable `FFMPEG_EXTRA_CFG`. For example, to eliminate the use of libvdpau (an nvidia support library) in the ffmpeg configuration step after you have determined that it is causing an error, use:

```
make clean
autogen.sh
export FFMPEG_EXTRA_CFG=" --disable-vdpau"
./configure ...
```

B.2 Experimental Builds

The main compilation we have seen leads to building CINELERRA-GG with its own internal ffmpeg, which includes its stability and feature patches. Normally ffmpeg is updated to version x.1 of each release. The reasons why it is best to use third-party with their static versions of the libraries and ffmpeg are explained in detail in Using the very latest Libraries and in Unbundled Builds.

You can still compile CINELERRA-GG with ffmpeg-git to enjoy the latest version. This build may lead to feature variation and less stability, but in most cases will work perfectly fine. You have to supply the actual URL location of the ffmpeg git as you can see in this example `bld.sh` script:

```
#!/bin/bash
./autogen.sh
./configure --with-single-user --with-booby --with-git-ffmpeg=https://git.ffmpeg.org/
    ↪ ffmpeg.git
make && make install ) 2>1 | tee log
mv Makefile Makefile.cfg
cp Makefile.devel Makefile
```

Since the procedure for obtaining the latest ffmpeg version is not always kept up-to-date and the line numbers will always change, you may have to create some patch first. Generally those line numbers are only updated by a developer when a new stable version with worthwhile features is actually included in the CINELERRA-GG build. FFmpeg is constantly changing and many times the git version is not as stable as desired.

Finally, it is possible to compile CINELERRA-GG so that it uses ffmpeg which is already installed on the system. This build takes less time to compile and may increase performance in both rendering and timeline manipulation. Again, there may be variations in functionality and less stability. Getting a build to work in a system environment is not easy. If you have already installed libraries which are normally in the thirdparty build, getting them to be recognized means you have to install the devel version so the header files which match the library interfaces exist. If you want to build using only the thirdparty libraries installed in your system, just include `--without-thirdparty` to your configure script. For example:

```
./confgure --with-single-user --disable-static-build --without-thirdparty --without-libdpx
```

The library, libdpx, is just such an example of lost functionality: this build of CINELERRA-GG will not be able to use the DPX format.

B.3 Configuration Features

A listing of the current configuration features as of January 11, 2020:

```
‘configure’ configures \CGG{} to adapt to many kinds of systems.

Usage: ./configure [OPTION]... [VAR=VALUE]...

To assign environment variables (e.g., CC, CFLAGS...), specify them as
VAR=VALUE.  See below for descriptions of some of the useful variables.

Defaults for the options are specified in brackets.

Configuration:
  -h, --help              display this help and exit
      --help=short        display options specific to this package
      --help=recursive    display the short help of all the included packages
  -V, --version           display version information and exit
  -q, --quiet, --silent   do not print ‘checking ...’ messages
      --cache-file=FILE   cache test results in FILE [disabled]
  -C, --config-cache      alias for ‘--cache-file=config.cache’
  -n, --no-create         do not create output files
      --srcdir=DIR        find the sources in DIR [configure dir or ‘..’]

Installation directories:
  --prefix=PREFIX         install architecture-independent files in PREFIX
                          [/usr/local]
  --exec-prefix=EPREFIX   install architecture-dependent files in EPREFIX
                          [PREFIX]

By default, ‘make install’ will install all the files in
‘/usr/local/bin’, ‘/usr/local/lib’ etc.  You can specify
an installation prefix other than ‘/usr/local’ using ‘--prefix’,
for instance ‘--prefix=$HOME’.

For better control, use the options below.

Fine tuning of the installation directories:
  --bindir=DIR            user executables [EPREFIX/bin]
  --sbindir=DIR           system admin executables [EPREFIX/sbin]
  --libexecdir=DIR        program executables [EPREFIX/libexec]
  --sysconfdir=DIR        read-only single-machine data [PREFIX/etc]
  --sharedstatedir=DIR    modifiable architecture-independent data [PREFIX/com]
  --localstatedir=DIR     modifiable single-machine data [PREFIX/var]
  --libdir=DIR            object code libraries [EPREFIX/lib]
```

```
  --includedir=DIR        C header files [PREFIX/include]
  --oldincludedir=DIR     C header files for non-gcc [/usr/include]
  --datarootdir=DIR       read-only arch.-independent data root [PREFIX/share]
  --datadir=DIR           read-only architecture-independent data [DATAROOTDIR]
  --infodir=DIR           info documentation [DATAROOTDIR/info]
  --localedir=DIR         locale-dependent data [DATAROOTDIR/locale]
  --mandir=DIR            man documentation [DATAROOTDIR/man]
  --docdir=DIR            documentation root [DATAROOTDIR/doc/cinelerra]
  --htmldir=DIR           html documentation [DOCDIR]
  --dvidir=DIR            dvi documentation [DOCDIR]
  --pdfdir=DIR            pdf documentation [DOCDIR]
  --psdir=DIR             ps documentation [DOCDIR]

Program names:
  --program-prefix=PREFIX            prepend PREFIX to installed program names
  --program-suffix=SUFFIX            append SUFFIX to installed program names
  --program-transform-name=PROGRAM   run sed PROGRAM on installed program names

Optional Features:
  --disable-option-checking  ignore unrecognized --enable/--with options
  --disable-FEATURE       do not include FEATURE (same as --enable-FEATURE=no)
  --enable-FEATURE[=ARG]  include FEATURE [ARG=yes]
  --enable-silent-rules   less verbose build output (undo: "make V=1")
  --disable-silent-rules  verbose build output (undo: "make V=0")
  --enable-dependency-tracking
                          do not reject slow dependency extractors
  --disable-dependency-tracking
                          speeds up one-time build
  --enable-a52dec         build a52dec (yes)
  --enable-djbfft         build djbfft (yes)
  --enable-audiofile      build audiofile (no)
  --enable-encore         build encore (no)
  --enable-esound         build esound (no)
  --enable-ffmpeg         build ffmpeg (yes)
  --enable-fftw           build fftw (auto)
  --enable-flac           build flac (auto)
  --enable-giflib         build giflib (yes)
  --enable-ilmbase        build ilmbase (auto)
  --enable-lame           build lame (auto)
  --enable-libavc1394     build libavc1394 (auto)
  --enable-libraw1394     build libraw1394 (auto)
  --enable-libiec61883    build libiec61883 (auto)
  --enable-libdv          build libdv (auto)
  --enable-libjpeg        build libjpeg (auto)
  --enable-opus           build opus (auto)
  --enable-openjpeg       build openjpeg (auto)
  --enable-libogg         build libogg (auto)
  --enable-libsndfile     build libsndfile (auto)
  --enable-libsvtav1      build libsvtav1 (no)
  --enable-libtheora      build libtheora (auto)
  --enable-libuuid        build libuuid (yes)
  --enable-libvorbis      build libvorbis (auto)
```

```
  --enable-mjpegtools     build mjpegtools (yes)
  --enable-openexr        build openexr (auto)
  --enable-tiff           build tiff (auto)
  --enable-twolame        build twolame (auto)
  --enable-x264           build x264 (auto)
  --enable-x265           build x265 (auto)
  --enable-libvpx         build libvpx (auto)
  --enable-lv2            build lv2 (auto)
  --enable-sratom         build sratom (auto)
  --enable-serd           build serd (auto)
  --enable-sord           build sord (auto)
  --enable-lilv           build lilv (auto)
  --enable-suil           build suil (auto)
  --enable-libaom         build libaom (auto)
  --enable-dav1d          build dav1d (auto)
  --enable-libwebp        build libwebp (auto)
  --enable-ffnvcodec      build ffnvcodec (auto)
  --enable-static-build   build static ([auto])
  --enable-x264_hidepth   build x264 10bit ([no])
  --enable-x265_hidepth   build x265 10bit ([no])

Optional Packages:
  --with-PACKAGE[=ARG]    use PACKAGE [ARG=yes]
  --without-PACKAGE       do not use PACKAGE (same as --with-PACKAGE=no)
  --with-jobs             parallel build jobs (auto)
  --with-exec-name        binary executable name (cin)
  --with-single-user      to install cin in bin (no)
  --with-ladspa-build     build ladspa library (yes)
  --with-lv2              lv2 library support (yes)
  --with-cinlib           cinelerra library path (auto)
  --with-cindat           cinelerra share path (auto)
  --with-plugin-dir       plugin install dir (auto)
  --with-ladspa-dir       ladspa install dir (auto)
  --with-config-dir       .bcast config dir ($$HOME/.bcast5)
  --with-nested-dir       nested proxy dir ($$HOME/Videos)
  --with-browser          cin_browser path (firefox)
  --with-git-ffmpeg       git ffmpeg using url (no)
  --with-noelision        use noelision/libpthread (auto)
  --with-booby            window lock trace booby trap (no)
  --with-libzmpeg         build libzmpeg (yes)
  --with-commercial       enable commercial capture (yes)
  --with-thirdparty       use thirdparty build (yes)
  --with-shuttle          shuttle device (yes)
  --with-wintv            usb 2040:826d wintv device (yes)
  --with-x10tv            usb 0bc7:0004 X10 remote device (yes)
  --with-vaapi            video acceleration api (yes)
  --with-vdpau            video decode+presentation api for unix (yes)
  --with-nv               nvenc/nvdec ffnvcodec api (yes)
  --with-cuda             nv cuda plugins (auto)
  --with-clang            use clang instead of gcc/g++ (no)
  --with-gl               use opengl (auto)
  --with-oss              use OSS audio (auto)
```

```
  --with-xft              use libXft (auto)
  --with-alsa             use libasound/alsa (auto)
  --with-firewire         use firewire (auto)
  --with-dv               use dv (auto)
  --with-dvb              use dvb (auto)
  --with-video4linux2     use v4l2 (auto)
  --with-xxf86vm          use xf86vmode (auto)
  --with-esound           use esd (auto)
  --with-shuttle          shuttle dev support (auto)
  --with-shuttle_usb      use libusb-1.0 (auto)
  --with-lv2              use lv2 (auto)
  --with-cuda             build cuda plugins (auto)
  --with-dl               system has libdl (auto)
  --with-opencv           opencv=sys/sta/dyn,git/tar=url (auto)
  --with-numa             system has libnuma (auto)
  --with-openexr          use openexr (auto)
  --with-onevpl           use Intel hardware oneAPI Video Processing Library (no)

Some influential environment variables:
  CC          C compiler command
  CFLAGS      C compiler flags
  LDFLAGS     linker flags, e.g. -L<lib dir> if you have libraries in a
              nonstandard directory <lib dir>
  LIBS        libraries to pass to the linker, e.g. -l<library>
  CPPFLAGS    (Objective) C/C++ preprocessor flags, e.g. -I<include dir> if
              you have headers in a nonstandard directory <include dir>
  CCAS        assembler compiler command (defaults to CC)
  CCASFLAGS   assembler compiler flags (defaults to CFLAGS)
  CXX         C++ compiler command
  CXXFLAGS    C++ compiler flags

Use these variables to override the choices made by 'configure' or to help
it to find libraries and programs with nonstandard names/locations.

Report bugs to <mail@lists.cinelerra-gg.org>.
```

B.4 Thirdparty Parallel Build

The Makefile in the thirdparty build directory employs a set of macros used to create a build rule set of thirdparty library build dependencies. The standard build sequence of [source, config, build] is used to prepare thirdparty products as static libraries. Build package dependency can be specified in the Makefile std-build macro call. These Makefile macro calls define the rules used for each thirdparty build. The expanded rule definitions may be viewed by using:

```
make -C thirdparty rules
```

Individual package libraries can be rebuilt, via:

```
make -C thirdparty <pkg>-clean; make -C thirdparty <pkg>
```

The rule targets create the set of thirdparty packages which are built from local source archive copies of thirdparty source code and patches, if needed. The build rule set of dependencies allows for compiling multiple thirdparty programs simultaneously using maximum computer resources. This parallel build speeds up the process considerably. For example, these are full static build timings on the production build machine (full build includes building all thirdparty programs as well as all of CINELERRA-GG):

1 cpu	=	61 mins
12 cpus	=	7.5 mins
24 cpus	=	2 mins

B.5 Using the very latest Libraries

Using the most current libraries can be a challenge for some of the Operating System distros that use stable compilers, assemblers, and their own libraries. Because they are stable, they frequently do not keep up with the very latest of thirdparty libraries. Consequently, some program constructs may not yet be implemented. CINELERRA-GG tries to maintain stability since it is better to have less features and no crashes. The goal is to make CINELERRA-GG widely available on many platforms rather than dependent on advanced tools that are not supported on some distros.

CINELERRA-GG attempts to upgrade to the latest releases of many thirdparty libraries about every 3-4 months. But it is often difficult to keep some of these thirdparty libraries up to date as their developers switch from tried and true standard tools to newer, less standard tools. As a result in order to build CINELERRA-GG on 2-3 versions of any distro rather than only the most current version, some thirdparty libraries can not be kept up to date and may even be to the point of no further updates. In a lot of cases, the updated releases provide little new capabilities but rather are bug fixes that may not even be relevant to CINELERRA-GG's use.

So as a computer savvy user or a developer, if you would like to build CINELERRA-GG with the latest thirdparty releases for some of the packages here are a few suggestions based on other developer's feedback and experimentation.

dav1d

Status - currently CINELERRA-GG is staying at 0.5.1. This is disappointing because there may be speed gains in later versions that would be beneficial. However,

it is usable for decoding whereas libaom is a lot slower. Unfortunately, it has no effective encoder.

Problem - 0.6 dav1d requires NASM 2.14 (and later versions of dav1d use even later versions of NASM) and uses instructions like vgf2p8affineqb, not exactly an "add" instruction. It also uses meson which is not widely available on all distros. The more recent NASM requirement apparently provides for using AVX-512 instructions (like vgf2p8affineqb, which is more like a whole routine than a simple instruction).

Workaround already in use by CINELERRA-GG - a Makefile was generated to replace Meson usage but has to be continuously updated for new releases. Dav1d 0.5.1 requires NASM 2.13 so at this level the newer distros will work. The availability of meson and nasm are a significant problem on many systems which are still in wide use.

Your workaround - Because a request to dav1d developers to consider changes to ensure their library is more widely usable does not appear to be in their future, since it works for them, you can upgrade NASM to 2.14 to stay up to date. Even then, you will have to build using meson and incorporate it into CINELERRA-GG.

OpenExr

Status - stable at 2.4.1 from February 2020.

Problem - the OpenExr tarball is not a single package but is 2 packages instead.

Workaround already in use by CINELERRA-GG - reworked the packages so that it looks like one package with 2 stubs.

Your workaround - perhaps use the same workaround.

OpenCV

Status - 2 different versions specific for O/S but none for Ubuntu 14, 32 or 64 bit.

Problem - There are really 2 problems here. The first is OpenCV is not really "Open" in that Surf is patented/non-free and there is no actual source available for certain capabilities. The second is that cmake 3.5.1 is required for OpenCV 4.2.

Workaround already in use by CINELERRA-GG - using 3.4.1 for older distros and 4.2 for newer.

Your workaround - upgrade cmake to 3.5.1 for upgrade to 4.2; add non-free to the compile; and use binaries that you do not know what they contain since no source code to compile. Look into opencv4/opencv2/core/types.hpp:711;27.

libaom

Status - currently at version 3.6.0 for older O/S and 3.8.0 for newer O/S.
Problem - requires cmake 3.5 at v3.6.0 and 3.7.2 for v3.8.0.
Workaround already in use by CINELERRA-GG - modify configure.ac to switch from 3.8.0 to 3.6.0 for Ubuntu 16 and delete thirdparty/src/libaom-v3.8.0*.*.
Your workaround - upgrade on some systems to cmake 3.7.2, switch to using 3.6.0 as in last sentence, or add –libaom-enable=no to configure line when building.

x10tv

Status - this is the x10 TV remote control.
Problem - INPUT_PROP_POINTING_STICK not defined error on older distros.
Workaround already in use by CINELERRA-GG - leaving out of Ubuntu14, Ubuntu, Centos7.
Your workaround - look into /usr/include/linux/input-event-codes.h.

lv2 plugins, consisting of 6 routines

Status - currently at version 1.18.0 for lv2 and different for other 5.
Problem - the current versions use cmake but the updated versions now all use meson and CINELERRA-GG is not set up to handle that.
Workaround already in use by CINELERRA-GG - not upgrading at this time.
Your workaround - if you are familiar with meson, you can independently upgrade the 6 routines.

B.6 Find Lock Problems with Booby Trap

A Booby Trap is used in CINELERRA-GG for setting a trap to catch lock problems that might have been missed. It will trap boobies only if compile by adding *--with-booby* on the configuration command line. This is the default if you compile using `./bld.sh` from the GIT repository. It should not interfere with normal execution.

If you have the time and inclination, enable *--with-booby* and send any trap output that you find. Maybe you will catch some boobies and if you do, send a snapshot of any boobies you find.

There are 2 potential traps:

- If you try to unlock a lock when it is not locked
- If you execute a drawing operation without holding the window lock

The trap prints the following in the controlling terminal window:

BOOBY! *<backtrace>*

An example backtrace is below along with a hint below on how to analyze:

```
/home/cin5/bin/./cin(_Z5boobyv+0x3f) [0x557069fa9b2f] /home/cin5/bin/./cin(
    ↪ _ZN13BC_WindowBase9draw_lineEiiiiP9BC_Pixmap+0x3b)0x557069fb9a9b]
/home/cin5/bin/./cin(\_ZN10BC_ListBox11draw_borderEi+0x73)[0x557069f7dc73]
/home/cin5/bin/./cin(+0x9707fb) [0x557069f7e7fb]
/home/cin5/bin/./cin(\ZN10BC\ListBox16center\selectionEv+0x4e)[0x557069f7f2ae]
/home/cin5/bin/plugins/video/sketcher.plugin(_ZN17SketcherCurveList6updateEi+0
    ↪ x1a0)[0x7f1b8002a4c0]
/home/cin5/bin/plugins/video/sketcher.plugin(
    ↪ _ZN18SketcherCurveColor17handle_done_eventEi+0x76)[0x7f1b8002a5f6]
/home/cin5/bin/./cin(_ZN15BC_DialogThread3runEv+0xd8)[0x557069f6fb78]
/home/cin5/bin/./cin(_ZN6Thread10entrypointEPv+0x45)[0x557069fc5995]
/usr/lib/libpthread.so.0(+0x7a9d) [0x7f1b91b4ea9d]
/usr/lib/libc.so.6(clone+0x43) [0x7f1b90accb23]
```

To see which routine is reporting the booby key in:

```
c++filt
```

And then the 2^{nd} line in the backtrace above:

```
_ZN13BC_WindowBase9draw_lineEiiiiP9BC_Pixmap
```

It comes back with the routine as:

```
BC_WindowBase::draw_line(int, int, int, int, BC_Pixmap*)
```

B.7 Valgrind Support Level

Valgrind is a memory mis-management detector. It shows you memory leaks, deallocation errors, mismanaged threads, rogue reads/writes, etc. CINELERRA-GG memory management is designed to work with Valgrind detection methods. This assists in developing reliable code. Use of Valgrind points out problems so that they can be fixed. For example, when this version of CINELERRA-GG shuts down, it

deallocates memory instead of just stopping, thus making memory leak detection possible. An alternative to Valgrind is *heaptrack* which has good documentation and for large programs can run faster. You can find the source and information about it at https://github.com/KDE/heaptrack.

The best way to compile and run valgrind is to run the developer static build. This takes 2 steps and you must already have gdb and valgrind installed:

1. The standard static build:
   ```
   cd /path/cinelerra-5.1
   make clean
   ./bld.sh
   ```
2. run the incremental rebuild for debug objs:
   ```
   CFLAGS=-ggdb make -j8 rebuild_all
   ```

If you frequently make mods and changes in CINELERRA-GG or the thirdparty libraries, do not depend on those compiled versions to have the headers updated; so be sure to fully rebuild as shown in the previous 2 steps before running valgrind or you will get false errors.

Now your CINELERRA-GG obj has all of the debug stuff. Next run valgrind as root for the most useful results:

```
    cd /path/cinelerra-5.1/cinelerra

    valgrind --log-file=/tmp/log --leak-check=full
--num-callers=32 ./ci
```

This runs CINELERRA-GG under the control of valgrind, and produces a log file in /tmp which will list information about any leaks, usually clearly identifiable. Be sure to Quit out of CINELERRA-GG normally instead of Ctrl-C or a SEGV otherwise the program does not have a chance to cleanup and there will be some false alarms. But it runs very slowly, and is basically single threaded, which means that race conditions may be impossible to catch... like one thread deletes memory that another thread is currently using. But overall it is a big help and if you test any new features, please email the log output. A lot of effort when writing the code was put into trying to be sure that all of the object constructors have matching destructors so that the leaks can be identified. There are already several libraries that create predictable memory leaks and valgrind does a good job for most of these.

It is impossible to test everything with valgrind because some things are just too big and slow for a practical test. Occasionally you can find a leak or an illegal

memory access. There are several false alarms that are difficult to avoid *Conditional jump* messages, and *unhandled DW_OP_*, but anything with the word *illegal* in the message is important. Memory leaks that originate in CINELERRA-GG are good to find and fix, but are usually not deadly. The listing of the memory leaks can be quite voluminous so locating the *LEAK SUMMARY* section towards the end of the report is most useful.

Another very useful valgrind run to locate unitialized variables while executing is:

```
    valgrind --log-file=/tmp/log --tool=memcheck
--num-callers=32 ./ci
```

B.8 CFLAGS has -Wall

When compiling CINELERRA-GG Infinity a CFLAGS option used is *Wall* where the "W" represents warnings and "all" means all. This causes the compile to check for simple mistakes that can be detected automatically and issue warnings when the code is questionable. It can also detect situations where the compiler will generate incorrect code, like type-punned pointer. By turning on this flag, when new code is vetted for predictable mistakes, the code can be corrected before becoming manifested in the application.

B.9 Prof2 – A Profiler

Frequently there is a problem with a program running slow and you do not know why. You need a thumbnail analysis of where the program is spending most of its time without all of the overwhelming details. This is when a Profiler comes in handy.

There are many different profilers available – this particular one does not do anything special and is fairly ordinary in that it just characterizes frequency of execution of various regions in the program. However, it has some pretty good strengths as listed next.

1. It takes very little or no preconditioning, i.e. setup
2. The effect it has on the running program is minimal
3. It runs almost at full speed, which is really nice
4. It is not particularly thread aware
5. Reports where it is executing at intervals of 100 times a second

B.9.1 Setup

This profiler works on x86_64 systems and you must be root to compile and run it. Also, you must install your operating system's *iberty* – for example, which would be binutils-devel for Fedora or libiberty-dev for Ubuntu 18.

Go to the top level CINELERRA-GG directory.

Key in: `prof2`

Key in: `make clean all install`

Because *smap* may have to be found in the system if *How to use* below does not work, you will have to do the following:

Key in: `cp -a smap /usr/local/bin`

Later, if you want to remove this from the system,

Key in: `make uninstall`

B.9.2 How to use

The program you are profiling should be compiled debuggable with stack frame and symbols enabled.

To see help, key in: `prof -h`

usage: -h [-o path] [-d] [-e] [-p libpath] [-012] [-u #] cmd args...

- -o profile output pathname, -=stdout, --=stderr
- -d debug otitleutput enabled
- -e child debug output enabled
- -p specify path for libprofile.so
- -0 usr+sys cpu timer intervals (sigprof)
- -1 usr only cpu timer intervals (sigvtalrm)
- -2 real time timer intervals (sigalrm)
- -u profile timer interval in usecs

To execute the profiler, key in:

```
prof -o /tmp/prof_list.txt ./cin
```

where `/tmp/prof_list.txt` is the output file and in this case `cin` is the CINELERRA-GG binary file. The pid of this command will be displayed on the startup window. This comes in handy in the use case where there is a lot of initial load and possible configuration setup inside of CINELERRA-GG and you want to profile plugins and not necessarily all of the setup steps. Then you can use the following command in another window to continue running CINELERRA-GG and obtain the more useful information:

```
kill -USR1 pid
```

Running this command refreshes the memory maps used for profiling. When you are profiling a plugin, you want to run this AFTER the plugin loads.

B.9.3 Results

There are 3 sections in the resulting output file of this stochastic quick analysis.

1. The first section is a histogram of the timer intervals of that sample set. Each function occupies a region of addresses. One hundred times a second, the profiler samples the program address in the region of execution.
2. In the second section, there is another histogram of the cumulative frequency of all things in the call stack and this should point out who is the culprit. For example, a codec calls a bunch of subroutines, the cost of the subroutines is accumulated to the codec parent caller. This makes the actual cpu user much more apparent.
3. The last section is for the library spaces. Each library occupies a region and the profiler adds up the time spent in each of the libraries by frequency.

On the very bottom is a 1 line summary which shows you if there is a bad guy and who it is.

B.9.4 Sample output

— *profile start* —

1020 ticks 43 modules 81412 syms

0.010s 0.1% Autos::copy(long, long, FileXML*, int, int) /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% BinFolders::copy_from(BinFolders*) /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% cstrdup(char const*) /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% XMLBuffer::encode_data(char*, char const*, int) /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% XMLBuffer::encoded_length(char const*, int) /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% PluginClient::send_configure_change() /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% UndoVersion::scan_lines(UndoHashTable*, char*, char*) /mnt0/.../cin
0.010s 0.1% UndoStackItem::set_data(char*) /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% UndoStack::load(_IO_FILE*) /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% BC_Bitmap::cur_bfr() /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% YUV::init_tables(int, int*, int*, int*, int*, int*, int*, int*, int*, int*, int*, int*, int*, int*) /mnt0/build5/cinelerra-5.1/bin/cin
...
...
— *profile calls* —

0.010s 0.1% AutoConf::save_xml(FileXML*) 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% Automation::copy(long, long, FileXML*, int, int) 1.0 /mnt0/.../cin
0.010s 0.1% AWindow::run() 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% Canvas::stop_single() 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% ColorPicker::new_gui() 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% ColorWindow::create_objects() 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% PaletteWheel::draw(float, float) 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% PaletteHex::update() 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
0.010s 0.1% CWindowGUI::draw_status(int) 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
0.010s **0.1%** CWindowCanvas::status_event() 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
...
...

```
0.990s     9.7%  BC_Xfer::xfer_slices(int) 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
1.880s    18.4%  DirectUnit::process_package(LoadPackage*) 1.0 /mnt0/build5/cinelerra-5.1/bin/c
1.880s    18.4%  DirectUnit::rgba8888() 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
3.910s    38.3%  __init_array_end 1.1 /mnt0/build5/cinelerra-5.1/bin/cin
5.450s    53.4%  LoadClient::run() 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
7.890s    77.4%  Thread::entrypoint(void*) 1.0 /mnt0/build5/cinelerra-5.1/bin/cin
7.890s    77.4%  start_thread 1.0 /lib64/libpthread-2.28.so
—
—
0.010s     0.1%/ 0.0%   /lib64/libm-2.28.so
0.010s     0.1%/ 0.0%   /lib64/libexpat.so.1.6.8
0.020s     0.2%/ 0.1%   /lib64/libXext.so.6.4.0
0.020s     0.2%/ 0.1%   /lib64/libXft.so.2.3.2
0.020s     0.2%/ 0.1%   /lib64/libxcb.so.1.1.0
0.040s     0.4%/ 0.2%   /lib64/ld-2.28.so
0.050s     0.5%/ 0.2%   /lib64/libpng16.so.16.34.0
0.130s     1.3%/ 0.6%   /lib64/libX11.so.6.3.0
0.180s     1.8%/ 0.8%   /lib64/libz.so.1.2.11
0.200s     2.0%/ 0.9%   /lib64/libfontconfig.so.1.12.0
0.380s     3.7%/ 1.8%   /lib64/libpthread-2.28.so
1.660s    16.3%/ 7.7%   /lib64/libc-2.28.so
7.480s   73.3%/34.7%   /mnt0/build5/cinelerra-5.1/bin/cin

10.200t 0.001u+0.000s 21.566r 47.3%
— profile end —
```

The summary line above in Bold represents the User time, System time, Real time and the percentage is how much Timer time elapsed over Real time so in this case the measurement covers 47.3% of time.

So why use a Profiler? Because it is the “ls” for executable functions!!

B.10 Working on AppImage

You can work on the appimage file to make changes and fix errors, or you can create a new appimage from scratch containing customizations. For example, you can add new rendering presets, update the Context-Help, change libraries that are no longer supported by the current distro, or make other modifications.

B.10.1 Managing AppImage

A limitation of using AppImage instead of installing the binary or compiling from git, is that there is only a single file without the ability to browse the directory structure or to look for files to edit or check. So if using CINELERRA-GG leads to some errors, it is not possible to investigate and fix the problem. Which means if you want to add the most up-to-date Context-Help or want to introduce some custom presets, that can not be done.

Because the appimage file is nothing more than a compressed file containing the same structure as the installed program plus other libraries that allow the program to run independently from the system, the content can be extracted so that you can work on it as you would have on the normally installed program. To do this you will need the appimage management program. Many Linux distros come with this managment program by default, but others may not. For instance in the case of Arch Linux the `appimagetool-bin` package from AUR needs to be installed.

To work on the appimage, first unpack it using the command[1] (note that you do not have to be root to do any of the following):

```
/{path to appimage}/CinGG-yyyymmdd.AppImage --appimage-extract
```

You will now have a `squashfs-root` folder in your current directory containing `/usr/bin/` as well as other files and directories such as `/usr/lib` and `/usr/share`. `Bin` is the folder similar to the one installed with CINELERRA-GG and contains the files that you can work on. Now it is possible to make changes like adding a custom preset in `/ffmpeg/video` or replacing a library that no longer works with a more recent version by working in `/squashfs-root/usr/lib`.

To start the unpacked program from the bin folder use the command:

```
/{path to appimage}/squashfs-root/usr/bin/./cin
```

After making your changes, to get back to having an appimage file instead of having to run the program from the bin folder, you can take the extra step of recompressing the squashfs-root folder. To do so use linuxdeploy's appimage, a program that can also be useful for creating appimages from scratch (https://github.com/linuxdeploy/linuxdeploy/releases/continuous).

The steps to recreate the appimage are:

[1] Example provided by Glitterball3

1. Copy the linuxdeploy appimage to /{path to appimage} and make sure it is executable.
2. Then use the command:

```
./linuxdeploy-x86_64.AppImage --appdir squashfs-root --output appimage
```

A new appimage will be created like the original but containing the changes.

Alternatively, download the appimagetool version from https://github.com/AppImage/AppImageKit/releases if available for your distro and use the command:

```
./appimagetool --comp /{path to appimage}/squashfs-root /tmpCinGG-yyyymmdd.
    ↪ AppImage
```

Now there will be an appimage called *CinGG-yyyymmdd.AppImage* with the changes that were made.

NOTE: for the **bdwrite** program which is used to create the udfs image to burn Blu-ray media (or any other standalone program), you will find it in the squashfs-root/usr/bin subdirectory. You only need to do the initial unpacking extract step and then just copy bdwrite and the entire squashfs-root/usr/lib directory to a convenient permanent place for usage. Then when running bdwrite, you will have to first export the libary path by typing this command in the window first:

```
export LD_LIBRARY_PATH=/{put your copied usr/lib path here}:$LD_LIBRARY_PATH
```

B.10.2 Build the CinGG.AppImage from scratch

If a developer wants to create an appimage from git, follow these next few steps. An existing automated script is available, bld_appimage.sh, in the directory /{path to cinelerra-5.1}.

This follows four steps:

- Build static version of CINELERRA-GG.

- If desired and needed, build the HTML manual for context-sensitive help.
- Organize code and if present, the manual, in a AppImage specific format *AppDir*. Do this with tool makeappimage, which is built if needed.
- Call an external tool to make the AppDir into an AppImage

Start by downloading the CINELERRA-GG source from Cinelerra's git. The last parameter is a directory name of your choice, the directory must not exist. As example, the name *cinelerra5* is used.

```
git clone --depth 1 git://git.cinelerra-gg.org/goodguy/cinelerra.git cinelerra5
```

The source will be in a subdirectory `cinelerra-5.1` of the directory created by the *git clone* operation.

If context-sensitive help is needed, download the manual sources too, with a different destination directory.

```
git clone --depth 1 git://git.cinelerra-gg.org/goodguy/cin-manual-latex.git cin-manual
    ↪ -latex
```

Then move to the /{`path to cinelerra-5.1`/} directory.

There are two preliminaries to do before running the script:

1- If context sensitive help in the appimage version is required, the source of the manual and the tools (packages) to build it must be on the system. In the bld_appimage.sh script, set the variable `MANUAL_DIRECTORY=$(pwd)/../../cin-manual-latex` to the path of the source of the manual. If the variable is empty, or the specified directory does not exist, CINELERRA-GG will be built without built-in help. The path to the manual source can be an absolute or relative one. An easier method to include the help from the manual rather than having to install a bunch of latex building software, is to simply download the latest tgz version from https://cinelerra-gg.org/download/images/HTML_Manual-202xxxxx.tgz. Then extract the files using "tar xvf" into the cinelerra AppDir/usr/bin/doc directory. This alternative method may not contain the most recent changes to the Manual but rather will contain what had been checked into Git by the date of the tgz file.

2- The script bld_appimage.sh uses a platform specific version of appimagetool so that it can create appimages for *x86_64*, *i686*, *aarch64*, or *armv7l* architecture. We need to add appimagetool-(platform).AppImage to the `/{path to cinelerra-5.1}/tools` directory, or somewhere in your path. You can download the tool for your system (e.g. appimagetool-x86_64.AppImage) from git: `https://github.com/AppImage/AppImageKit/releases`. Be aware of the possibility that an older appimagetool from 2020 might work better on some systems compared to the latest release.

Always remember to make it executable. The four supported platforms are:

- appimagetool-aarch64.AppImage
- appimagetool-armhf.AppImage
- appimagetool-i686.AppImage
- appimagetool-x86_64.AppImage

With the preparations done, let's go back to the bld_appimage.sh script.

Make the script executable if it isn't already:

```
chmod x+u bld_appimage.sh
```

Start the script with:

```
./bld_appimage.sh
```

The first part of the script customizes `configure`, depending on the architecture of our system. The next step compiles and installs CINELERRA-GG in the `bin` subdirectory. The third part creates the manual in HTML; It will be the basis for the context help (see Help and Context Help). The fourth part creates the directory `/AppDir/usr`, which is the basic structure of an appimage, and then populates it with the contents of `/{path to cinelerra-5.1}/bin`. Finally, the fifth step starts a script, called `makeappimage`, which points to *bld.sh* contained in `/tools/makeappimagetool/`.

The script specifies all executables that are part of CINELERRA-GG, so makeappimage can pick up dependencies. Any executable code in other places is not picked

up. makeappimage will populate the AppDir directory, and then call the platform dependent appimagetool found in `/{path to cinelerra-5.1}/tools`.

At the end of the compilation there will be an appimage in the `cinelerra-5.1` directory, which we can move and use.

B.11 How to Create a new Theme

A *Theme* is a base class object that is created and customized as *ThemeName*. It is constructed during program initialization in a theme plugin `PluginTClient`, defined in `plugins/theme_name` source directory.

`theme_name.C` and `theme_name.h` are derived *Theme* class object constructors.

A *Theme* is constructed during initialization in `init_theme` (`mwindow.C`). The theme plugin is accessed using the *name* from preferences and then the theme plugin is loaded which contains the code to construct the theme. A *Theme* object has functions and data that CINELERRA-GG uses to do a variety of customizations, such as `default_window_positions`, and it can modify GUI defaults like `default_text_color` when it is initialized.

The theme plugin contains a *new_theme* function that allocates and constructs a *ThemeName* object with base classes of *BC_Theme* (gui setup), *Theme* (CINELERRA-GG defaults), and *ThemeName*, with definitions and overrides that create the custom theme. To create a new theme, a new plugin is needed:

```
#include "header files.h"
PluginClient* new_plugin(PluginServer *server)
{
  return new NameMain(server);
}

NameMain::NameMain(PluginServer *server)
: PluginTClient(server)
{
}

NameMain::~NameMain()
{
}
const char* NameMain::plugin_title() { return N_("Name"); }

Theme* NameMain::new_theme()
```

```
{
  theme = new ThemeName;
  extern unsigned char _binary_theme_name_data_start[];
  theme->set_data(_binary_theme_name_data_start);
  return theme;
}

Name::Name()
: Theme()
{
}

Name::~Name()
{
  delete stuff;
}
```

When a theme is constructed by `NameMain::new_theme()`, it sets a pointer to a block of data created in the plugin build that contains all of the png data files in the `plugins/theme_name/data` directory. These images may define or override the appearance of gui images, such as *ok.png* (the ok button). There are usually a large number of images that need to be defined. The theme plugin adds them to the theme image data in the `theme → initialize()` function. The best list of theme image setup is probably in SUV (`plugins/theme_suv/suv`).

The easy way to create a new theme is to copy an existing theme and change its name to *ThemeName*, change `plugin_title()` to the new name, and then tweak the definitions until you are happy with the results. The file names and Makefile also need to be updated to the new theme name. The source can by manually rebuilt by invoking *make* in the `plugins/theme_name` directory.

Once the new theme is built into the plugin library, it will automatically be discovered by the plugin probe and it will become an available theme in *Preferences*.

If you are ready to add it to the main build, then *theme_name* should be included in the DIRS targets of the `plugins/Makefile`, and `plugin_defs` needs *theme_name* in the themes list.

Themes usually require considerable time to create from scratch. For example, the SUV theme has over 800 lines in the initialize function, and has over 500 png images in the data directory. Most of these images and data values are required to be initialized by the custom theme constructor; very tedious and time consuming work. Creating a new theme is usually a lot of work.

B.12 How Context Help works in the Program Code

All class methods related to context help start with the common pattern `context_help` in their names. It is easy to get all occurences in the code with the following command (example):

```
grep -F context_help ‘find . -name ’*.[Ch]’ -print‘
```

The base functionality is defined in several *BC_WindowBase* class methods in `guicast/bcwindowbase.C` (search on `context_help`). All BC_Window's and BC_SubWindow's inherit these methods.

For the simplest case of context help definition, it is sufficient to add the call to `context_help_set_keyword()` in the most inclusive widget constructor. If `Alt/h` is pressed with the mouse over this widget's window, its `keypress_event()` method (inherited from BC_WindowBase) will catch this hotkey, fetch the keyphrase defined by `context_help_set_keyword()` and call the `doc/ContextManual.pl` script with this keyphrase. Then ContextManual.pl script does the whole processing of the keyphrase given and calls web browser to display the found HTML manual page. The browser is called in the background to prevent from blocking the calling CINELERRA-GG thread.

An example from *cinelerra/zoombar.C*:

```
ZoomBar::ZoomBar(MWindow *mwindow, MWindowGUI *gui)
: BC_SubWindow(...)
{
        this->gui = gui;
        this->mwindow = mwindow;
  context_help_set_keyword("Zoom Panel");
}
```

If `Alt/h` is pressed with the mouse over some subwindow (arbitrary depth) of the `context_help_set_keyword()` caller, the `keypress_event()` method of that subwindow catches the hotkey. Its own context help keyword, probably, will be empty. In this case the whole widget hierarchy is traced up to *top_level* widget, their context help keywords are checked, and the first nonempty keyword is used for context help.

This approach allows us to define the help keyword common to the whole dialog window with a bunch of diverse buttons with a single call to `context_help_set`

`_keyword()`, without placing such a call into each button constructor. And at the same time, this approach allows to assign different help keywords to GUI elements belonging to the same window but documented in different manual pages.

B.12.0.1 An example with several different help keywords from cinelerra/mwindowgui.C:

```
MWindowGUI::MWindowGUI(MWindow *mwindow)
: BC_Window(_(PROGRAM_NAME ": Program"), ...)
{
        this->mwindow = mwindow;
        ...
        context_help_set_keyword("Program Window");
}
...
FFMpegToggle::FFMpegToggle(MWindow *mwindow, MButtons *mbuttons, int
    ↪ x, int y)
: BC_Toggle(...)
{
        this->mwindow = mwindow;
        this->mbuttons = mbuttons;
        set_tooltip(get_value() ? FFMPEG_EARLY_TIP : FFMPEG_LATE_TIP);
        context_help_set_keyword("FFmpeg Early Probe Explanation");
}
...
StackButton::StackButton(MWindow *mwindow, int x, int y)
: BC_GenericButton(x, y, mwindow->theme->stack_button_w, "0")
{
        this->mwindow = mwindow;
        set_tooltip(_("Close EDL"));
        context_help_set_keyword("OpenEDL");
}
...
ProxyToggle::ProxyToggle(MWindow *mwindow, MButtons *mbuttons, int x,
    ↪ int y)
: BC_Toggle(...)
{
        this->mwindow = mwindow;
        this->mbuttons = mbuttons;
        ...
        context_help_set_keyword("Proxy");
}
```

If the widget we wish to add context help to, overloads keypress_event() of the

base class with its own method, then it is necessary to introduce a small addition to its keypress_event() handler: the call to `context_help_check_and_show()` has to be added in a suitable place in the handler.

An example with `context_help_check_and_show()` from *cinelerra/mbuttons.C*:

```
int MButtons::keypress_event()
{
     int result = 0;
     if(!result) {
          result = transport->keypress_event();
     }
     if(!result) {
          result = context_help_check_and_show();
            }
     return result;
}
```

All the keypress handlers are not equal, therefore we provide different variants of the methods context_help_check_and_show() and context_help_show() (the latter for explicit checking on hotkey).

An example with context_help_show() from cinelerra/editpanel.C:

```
int EditPanelTcInt::keypress_event()
{
          if( get_keypress() == 'h' && alt_down() ) {
                    context_help_show("Align Timecodes");
                    return 1;
          }
          ...further processing...
          return 1;
}
```

A single example of much more sophisticated keypress_event() handler can be found in `cinelerra/trackcanvas.C` (search on context_help). The problem here was to track the particular object drawn on the canvas to figure out whose context help is to be requested. Another rare example of difficulty may appear when context help is desired for some GUI object which is not subclassed from BC_WindowBase.

ContextManual.pl looks for occurence of keyphrases in the following order:

1. In *Contents.html*
2. In *Index.html*
3. In all the *CinelerraGG_Manual/*.html* files via grep

Keyphrase matching is tried first exact and case sensitive, then partially and case insensitive. The special keyword `TOC` shows Contents, `IDX` shows Index. If the keyword starts with `FILE:`, the filename after *FILE:* is extracted and that file is shown. You can look in the ContextManual.pl script text.

For debugging purposes the script can be called from command line. For this to work, the environment variable `$CIN_DAT` has to be set to the parent directory of doc.

B.12.0.2 Providing context help for plugins

In the simplest case, nothing has to be done at all. The plugin context help functionality introduced in `cinelerra/pluginclient.C` automatically assigns context help keyword from the plugin's title for all plugins subclassed from *PluginClient*. For this to work, the manual page documenting the plugin must be entitled identically to the programmatic title of the plugin. A special treatment can be necessary in the following cases:

1. *Rendered effects* are not subclasses of PluginClient and require an explicit context help definition via `context_help_set_keyword()`.
2. Only the main plugin dialog inherits context help functionality from the parent class. If a plugin opens additional dialog windows, they probably require explicit context help definitions.
3. If the plugin title differs from its subsection title in the documentation, either its help keyword or the corresponding title in the manual should be renamed. Another possibility can be an addition to the *%rewrite table* in `doc/ContextManual.pl`.
4. If the plugin title contains some characters special for HTML and/or Perl regular expression syntax, the special characters have to be escaped. For example, the keyphrase corresponding to the title *Crop & Position (X/Y)* should be converted to:

```
Crop & Position \\(X\\/Y\\)
```

Such a rewriting can be done either in the C++ code or in ContextManual.pl.

One additional explanation about previous user information when plugin tooltips are off. Help for the *selected plugin* is shown because in this case it would be difficult to figure out, without a visual feedback, which particular item is currently under the mouse.

B.12.1 Manual Maintenance is now Tightly Coupled with the Program Code

If some section of the CINELERRA-GG manual gets renamed and is used for Context Help, the corresponding help keywords should be adjusted too. All the keywords used can be listed via the following example command:

```
grep -F context_help ‘find . -name ’*.C’ -print‘ | grep -F ””
```

————————-

Note that the keyword does not have to exactly match the section title; it can also be a case insensitive substring of the section title.

If some new CINELERRA-GG window or dialog is created and documented, the inclusion of context_help_set_keyword() calls and/or adaptation of keypress_event() handler (if any) should be included in the program code.

If some new CINELERRA-GG plugin is created, it is best to document it in the section entitled exactly equal to that plugin's title. Then probably its context help will work out of the box. Otherwise, some adaptation to its keypress_event() (if any) or to the

Why the local copy of CINELERRA-GG manual should be used?

1. For context help keyphrases matching, the local copy of *Contents.html* and *Index.html* is necessary anyway.
2. Grepping *CinelerraGG_Manual/*.html* files of the remote manual from the website cannot work per definition.
3. The local copy usage ensures exact matching of the version of the manual to the version of CINELERRA-GG. Otherwise, if one uses for some reason an

older version of CINELERRA-GG with the help fetched from the newer version of the website manual, various incompatibilities can be expected.

4. Packing the manual into AppImage, the current method of CINELERRA-GG packaging, should be much easier than merging two different git branches when building from source packages, as was done earlier.

What about Localization?

For now, CINELERRA-GG context help is not localized at all. There is no complete CINELERRA-GG manual in any language except English. But even should the manual be translated to other languages, the context help keywords must remain unlocalized literal string constants. First, because the set of languages known to CINELERRA-GG itself is not the same as the set of languages the manual will be translated to. If some "translated keyword" is fed to the help system, while the manual in this language does not exist, keyword matching cannot succeed. Second, such a help localization with the translation of all keywords can be done and then maintained much easier and much more reliably inside the ContextManual.pl script rather than in the CINELERRA-GG binary.

More about ContextManual.pl file?

The bin/doc/ContextManual.pl script can be configured further. Look in the script text. You can define your preferable web browser, or redefine it to 'echo' for debug purposes. There are also some predefined HTML pages for Contents and Index, and several explicitly rewritten keyphrases.

After the first invocation of context help the system-wide script, bin/doc/ContextManual.pl, copies itself into the user's config directory, $HOME/.bcast5. If the user needs to modify ContextManual.pl, it should be done on that copy from .bcast5 rather than the system-wide one. On all subsequent context help requests this (possibly modified) .bcast5 version of the script will be called. If later the CINELERRA-GG package gets upgraded containing a newer, not 100% compatible version of ContextManual.pl script, and/or not compatible structure of the HTML manual itself, the new system-wide version of the script will be copied into .bcast5 again to ensure context help functionality. The older version of the script will be backed up (with the .bak suffix) as a reference of the modifications done by the user.

B.13 Unique Blend plugins workflow

The uniqueness of the Blend Algebra/Blend Program plugins, different then most of the other plugins, provides a universal tool for creating new effects and may

be handy in unexpected cases. These 2 plugins provide the advanced feature of combining and modifying color pixels and transparencies of several tracks according to a mathematic algorithm written entirely by the user in the form of compact and simple piece of code. Such user defined algorithms (blend functions) are compiled, dynamically linked and immediately executed inside CINELERRA-GG on the fly without the need to restart the application, reload the project, reattach plugins, and so on. Because of this uniqueness, the Blend Algebra and Blend Program plugins workflow is documented here to explain how they work. Additional helpful information for developers can be found in the section on the Blend Algebra/Blend Program plugins themselves at Blend Algebra and Blend Program. You will most likely need to read parts to get a better understanding.

B.13.1 Preparation phase

As in any realtime plugin, the `process_buffer()` method in Blend Algebra/Blend Program gets a set of frames from the tracks the plugin is attached to. Then the following events take place. It is checked if the configuration (the parameters) got changed. There is one parameter which requires special treatment, the *user function name*.

In order to prevent resource consuming recompilation of the functions on each new frame, the plugin maintains the successfully compiled and attached functions in cache. If at some keyframe the function name gets changed, the plugin searches if this function is already known and cached. In addition to important function related objects such as entry points, there is a timestamp representing when this function was last checked to be up to date.

If the current function name is empty, it means a function is not used. Nothing else has to be done; all tracks are fetched and passed along in the processing pipeline unchanged. If the function is not empty and seen for the first time, or its timestamp is older than the global timestamp, it is checked as follows.

1. File lock is placed on the function source file to prevent concurrent modification of object files in case of several simultaneous compilations.
2. Compilation script `BlendAlgebraCompile.pl/BlendProgramCompile.pl` is started. The script checks if the resulting shared object file exists and is newer than the source and recompiles it if not.
3. The plugin checks if the shared object timestamp became newer than the timestamp of this function in cache (if any). If the cached version of the function in memory is up to date, it stays there. If not, the outdated function is detached from the plugin, the updated one is reattached, and its entry

points are fetched and put into cache. The function's timestamp in cache is set to the current time since the function has just been checked.

While recompiling or dynamic linking, various things may go wrong.

1. If in the unlikely scenario where the given function file does not exist, the program does nothing, same as for an empty function. No error message is shown in this case in order to prevent a possible deadlock.
2. If recompilation was unsuccessful because of a syntax error, the error message is shown. More detailed diagnostics from the compiler can be seen in the terminal window in which CINELERRA-GG was started.
3. If compilation succeeded, but dynamic linking did not, the error message is shown. In case of any error, the failed function is marked with the current timestamp in cache so that the error messages appear only once before the global timestamp gets updated.

Updating global timestamp forces all cached functions to be checked for recompilation when first accessed. The global timestamp is updated when the following events occur: changing function name, pressing `Edit...` or `Refresh` button, or exiting the `Attach...` dialog with the `OK` button.

If the current active function is up to date and attached, the plugin fetches video frames from all of the affected tracks with the important parameters like frame width and height. Then the `INIT` phase of the function is executed (once for each frame). Several important parameters are requested as defined by the function. They are:

- Working color space needed inside the function. If it is not the same as the color space of the project, then color space conversions have to be done.
- The required number of tracks the function works on. If less than the required number of tracks is available, an error message is shown and the function is not executed.
- Whether the function supports parallelizing or not. If the function does not claim parallelizing support, it will be executed sequentially even if the *PARALLE_REQUEST* checkbox is ON in the plugin GUI.

B.13.2 Processing phase

After the preparation phase, the processing itself takes place. If running sequentially instead of parallel, the following is done for each frame pixel individually.

For each input track, the corresponding pixel is split into color components according to the actual color space of the project. All color components are converted to float (type of the C language) in the ranges [0.0..1.0] for R, G, B, A, Y or $[-0.5.. + 0.5]$ for U, V. If the project color space has no alpha channel, the alpha component is set to A=1.0.

If the function uses a different color space than the project uses, the required conversions are performed. The key color components (selected in the plugin GUI) are also converted to the function color space in the same manner.

For Blend Algebra, the values for output are preinitialized from the track which the output is to go to. All the other tracks are cleared if the corresponding checkbox in the plugin GUI is checked. For Blend Program, this step is not needed.

B.13.3 User function phase

The user function is called with the parameters: actual *number of tracks*, *4 arrays for the 4 color components* (dimensions are equal to the number of tracks), *4 key color components*, *current pixel coordinates* (x, y: upper left frame corner is (0,0), lower right is (width-1,height-1)), and *width and height*. In addition for Blend Algebra, there ae *placeholders* for the result.

The user function returns. First of all, the color components of the relevant pixels are clipped to range if the corresponding checkbox in the plugin GUI is on. Relevant for Blend Program are pixels of all the tracks (all tracks can be modified); for Blend Algebra the result only.

After optional clipping, the color components are checked for not being NaNs. If so, the pixel is replaced with the substitution color, then backconversion to the project color space is performed.

If the project has no alpha channel, but the function returned something not equal to 1.0, alpha channel is simulated as if an opaque black background were used.

If the project color space is not FLOAT, unconditional clipping followed by 8-bit transformation takes place.

For Blend Algebra, the result is placed in the right output track. For Blend Program, this step is unnecessary since all tracks are modified in place.

Then the loop is repeated for the next pixel in a row, next row in a frame, next frame in the whole sequence, and so on.

If the function is to be run parallelized, the necessary number of threads is created (as defined in `Settings → Preferences → Performance; Project SMP cpus`). Running parallel on 1 cpu is not exactly the same as running sequential because an extra processing thread is created, while sequential execution takes place inside the main plugin thread. This could induce some subtle differences if the function uses static variables inside or something else which is thread unsafe.

Appendix C

Auxiliary Programs

C.1 Using Ydiff to check results

Delivered with Infinity CINELERRA-GG and in the CINELERRA-GG path, there is a file `ydiff.C` This program compares the output from 2 files to see the differences . Do: `cd cin_path` and key in `make ydiff`.

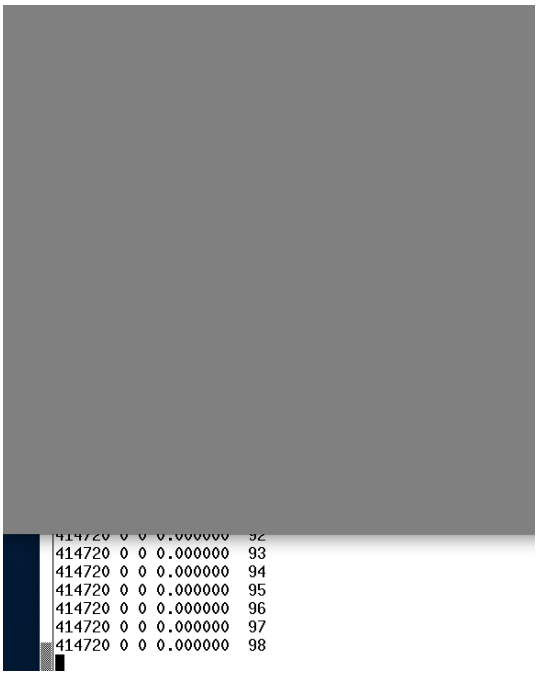


Figure C.1: *Exact match*

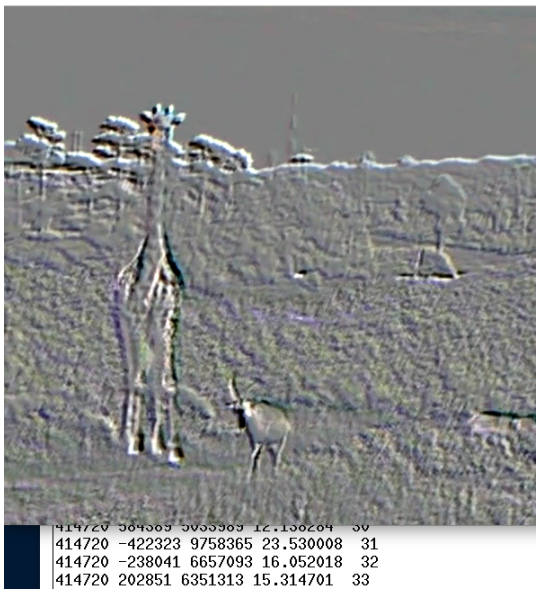


Figure C.2: *"Giraffe" artifacts on 2 files spaced differently*

You can now use this to check the quality differences of various outputs. For example, in this same directory key in:

```
./ydiff /tmp/yourfile.mp4 /tmp/yourfile.mp4
```

Since you are comparing a file to itself, you will see a clean looking white window in the left-hand corner and columns 2,3,4 will be all zeros. Run this same

command with a 3rd spacing parameter of -1 as shown below, and you will see artifacts of comparing 2 files starting in a different position.

```
./ydiff /tmp/yourfile.mp4 /tmp/yourfile.mp4 -1
```

Now render yourfile using different quality levels and run ydiff to compare the 2 results. You will see only noise difference which accounts for the quality level. Columns 2,3,4 might no longer be exactly zero but will represent only noise differences. The ydiff output is debug data with lines that show frame size in bytes, sum of error, and sum of absolute value of error. The frames size is sort of useless, the sum of error shows frame gray point drift and the abs error is the total linear color error between the images. At the very end is the total gray point drift and total absolute error on the last line.

C.2 Image Sequence Creation

Example script to create a jpeg list sequence file is next. It can be modified to create a list for *exr*, *gif*, *ppm*, *png*, *tga*, or *tiff* sequences instead by changing JPEGLIST to be EXRLIST, GIFLIST, PPMLIST, PNGLIST, TGALIST, or TIFFLIST.

```
#!/bin/bash
out="$1"
dir=$(dirname "$out")
shift
geom=$(anytopnm "$1" | head -2 | tail -1)
w="$(echo $geom | cut -d " " -f1)"
h="$(echo $geom | cut -d " " -f2)"
exec > $out
echo "JPEGLIST"
echo "# First line is always format_typeLIST"
echo "# Frame rate:"
echo "29.970030"
echo "# Width:"
echo "$w"
echo "# Height:"
echo "$h"
echo "# List of image files follows"
while [ $# -gt 0 ]; do
  f=./`basename "$1"`
  echo "$f"
  shift
done
```

To use this script, you will have to install the package on your operating system that includes *anytopnm* which is ususally *netpbm*. Example usage of this script follows:

```
./imagelist.sh outfile infiles*.jpg
```

where *imagelist.sh* is just the name chosen for this script which could be anything but has to be executable, *outfile* is the sequence list that is created, *infiles*.jpg* are the format_type files such as *jpg* in this example. The xxxtopnm messages that show up can just be ignored such as: `jpegtopnm: WRITING PPM FILE` and `jpegtopnm: Error writing row`. You may have to edit this script to suit your needs or to include specific directory locations. See also Filelist format.

C.3 Details about .bcast5 Files

The following extensions of files in CINELERRA-GG's `.bcast5` directory are explained below.

.dat	represent saved *data* for perpertual sessions and color palettes; maybe others
.idx	original *index* files that were created for loaded video to speed up seeking
.mkr	ffmpeg specific *marker* index file that is created for each video to aid seeks
.rc	rc stands for *run commands* so basically represents a script
.toc	toc is *table of contents* file for MPEG video files (a type of index)
Cinelerra_plugins	a list of the currently loaded plugins available in your CINELERRA-GG session
Cinelerra_rc	the user's preferences and settings are saved in this file to be used on startup. This file can be carefully edited to change startup values for certain parameters, but if you inadvertently set up something incorrectly, you may end up crashing the program.
ContextManual.pl	the user's configurable version of the Perl script which drives the Context Help feature
ladspa_plugins...	list of currently loaded ladspa plugins for each version of CINELERRA-GG being used

layout#..._rc	user-defined window layout setup with the layout name as part of the file name
.xml	used for various backups or for the current settings of plugins that you have used
.png	thumbnails of files in Resources so they do not have to be created over and over

C.4 Focusing the 4 main windows as a group

When working with multiple programs it is often the case where a set of windows needs to be minimized or maximized temporarily while working with a different program. This can be a little awkward with CINELERRA-GG because there are 4 distinct windows that are treated individually. To improve the workflow so that the 4 are minimized or maximized together as a group, you can use a routine called *xdotool*. This will help to automatically focus all 4 of the windows as one window while still letting the user reposition the 4 screens. The "focusing windows as a group" option makes it so that you can quickly move back and forth between this program and other programs. Here is how to do this.

Install xdotool https://www.semicomplete.com/projects/xdotool/

Iconifying all Cinelerra windows at once:
xdotool search –name Cinelerra windowminimize %@

Reactivating all Cinelerra windows at once:
xdotool search –name Cinelerra windowactivate %@

Appendix D

Editing: real-world usage cases

This appendix includes descriptions of various real life workflows used by Cinelerra-GG users. There may be links to some video explanations in various languages. You can only follow these links from the online pdf version.

D.1 Workflow with OpenEDL and Nested Clips

This is a real world usage case[1] that provides an excellent example of how OpenEDL has been a revolution for Cinelerra-GG. Advantages include: editing speed, clarity, ease of finding a specific item, and movement of a subject block via *Nest to media*. The main concept of editing using OpenEDL in this real world case is to reduce a large quantity of videos and shots to a few big clips by using an OpenEDL session for each clip. Using the OpenEDL timeline is similar to using the normal timeline: you can add effects, cut and drag, add or delete tracks, etc.

For more information on OpenEDL and Nested clips see sections 6.3 and 6.6.2.

Scenario setup consists of making one hour video using 3 different cameras:

- *Footage 1:* A person uses camera 1 to shoot a general view of the subject.
  a) Barcelona (e.g. a jazz group concert)
  b) Lisbon (e.g. Spanish concert)
- *Footage 2:* Another person uses camera 2 to film a detail of a specific subject.
  a) Barcelona (e.g. saxophonist or trumpet player)
  b) Lisbon (e.g. guitarist or double bassist)
- *Footage 3:* Contains images of the journey between the subjects (action cam on the windshield of a moving car, on the helmet of a biker, on the handlebars of a bicycle, or on a boat).

[1] credit fary54

Note: Camera 1 and 2 are both filming the same subject so the number of hours for the footage from these two cameras easily reaches 20 hours, but the final movie can only be one hour maximum. For this case, the final movie consists of all the subjects separated by camera 3 footage.

Workflow: OpenEDL is used throughout the editing process with the timeline reserved as the workspace. Basically, every time you have to work on a clip (consisting of one or more tracks), you use OpenEDL instead of the classic method of bringing each clip to the timeline, before doing the necessary Drag and Drop or Cut and Paste operations between different tracks. When finished with each OpenEDL clip session, save it outside of the OpenEDL layer at the top level of the main timeline.

See Video 1 (using French locale).

Pre-sorting of the footage

- Load all the shots from camera 1 and create a single clip; do the same with cameras 2 and 3.
- Watch each clip and sort the images to keep (via OpenEDL); delete the blurred footage and the images you do not want to keep.

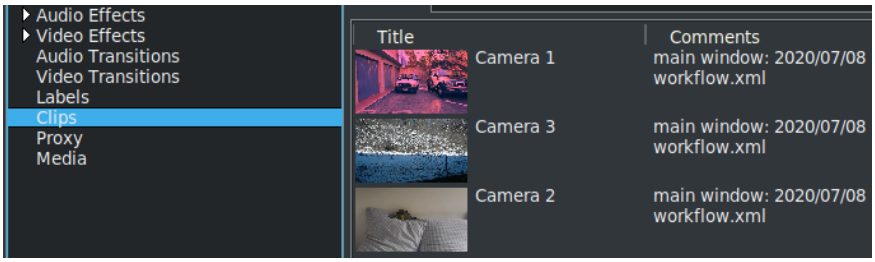


Figure D.1: *three clips modified with OpenEDL*

Camera 1 contains shots of Barcelona and Lisbon concert made by Camera 1.

Camera 2 contains shots of Barcelona and Lisbon concert made by Camera 2.

Camera 3 contains the images of the journey between the first subject and the second (e.g from Barcelona to Lisbon).

Extraction of the subjects

Next extract each common subject from the camera 1 & 2 footage and create a clip, for example - a place, a city, or a subject. In this case scenario, it is a concert in a

specific city. You end up with as many clips as there are subjects (e.g. Barcelona & Lisbon). See figure D.2.

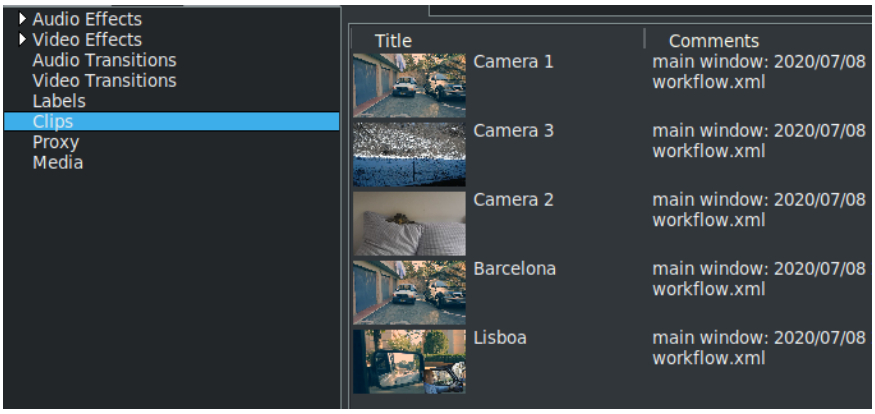


Figure D.2: *a new clip for each subject: Barcelona and Lisbon*

Each subject selected contains 2 video tracks and 4 audio tracks. See figure D.3 of a Barcelona clip.

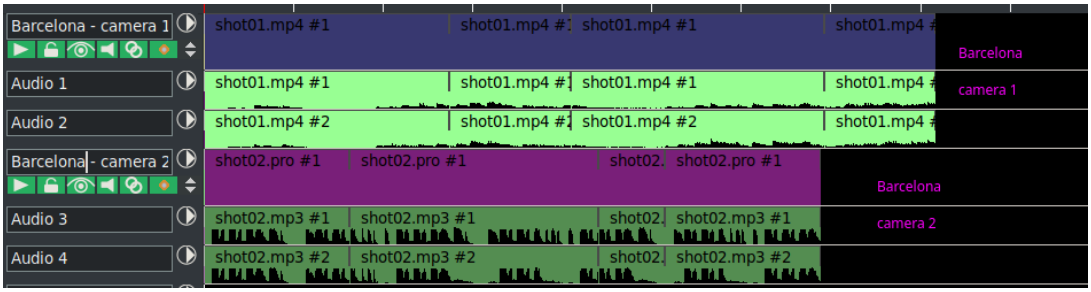


Figure D.3: *Barcelona clip: 2 video tracks and 4 audio tracks*

Editing of the subjects

Then edit each subject via OpenEDL. From the 2 tracks 1 & 4 (camera 1 & 2) you create a a single track (+ 2 audio tracks) as seen in figure D.4.

Figure D.4: *Barcelona clip: 2 video tracks to 1 video track*

Add sound effects and video/audio effects

Next add the sound effects (such as birds or a water fountain) and video / audio effects for each subject; again via OpenEDL.

Compaction

Each subject is converted to a nested clip by using the *Nest to media* option, making it a group.

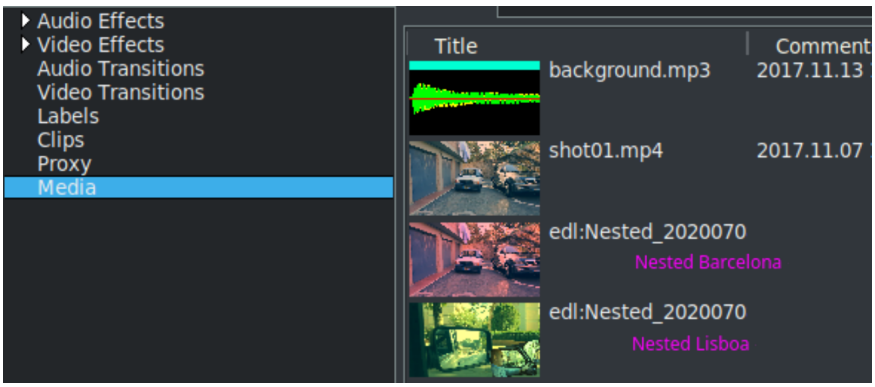


Figure D.5: *two new nested clips: Barcelona and Lisbon*

The subject converted to a nested clip is automatically moved from the Clip folder to the Media folder. See figure D.5.

See Video 2 (using French locale).

Final assembly

Import each subject (Nested clip) on the main timeline. Converting using Nest to Media makes it easy to move and position each subject. You can move one subject after or before another.

Figure D.6: *Barcelona and Lisbon nested clips on timeline*

Insert the retained shots coming from *camera 3* between the different subjects.

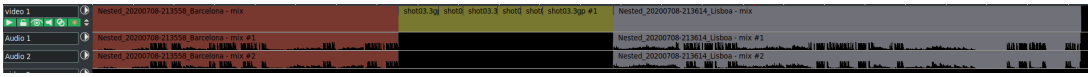


Figure D.7: *camera 3 footage between Barcelona and Lisbon nested clips*

See Video 3 (using French locale).

Add background music and comments.

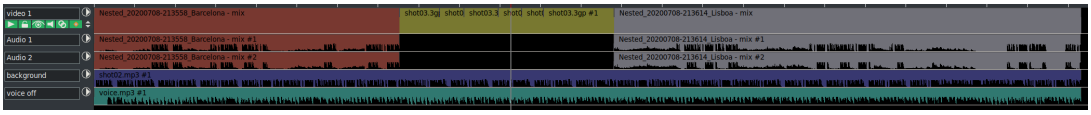


Figure D.8: *completion of the work*

More editing

At this stage, each subject can still be edited and undergo any modification via OpenEDL (again ... and always)

Render your movie

All that remains is mastering the video with a render.

Ordinarily it it is very difficult to set up such a project but using tools such as OpenEDL and Nested clips makes it quite easy. These tools, in many cases, are essential and become standards for the development of important projects using CINELERRA-GG.

Note Once converted to nest to media the clips disappear from the clips folder to end up in the media folder. You can always, if you wish to recover it in the clips folder do so via the *EDL to clip* menu. In this case, the name of the clip can change but the reference of the clip can be found in the comment of the clip.

See Video 4 (using French locale).

D.2 Workflow with multi-cam and external audio

More on the multi-cam can be found in section Multi-Camera / Mixer

Let's take the case of a professional magician filmed in multicam while performing[2]. Camera 1 records the magician's face; camera 2 the whole person and camera 3 the detail of the hands. There is also an external audio recorder to record the artist's voice in mono. So we will have three video tracks with their respective embedded audio tracks, plus an external audio track of good quality.

You can find the files to test the workflow that is described next at the following address: `https://cinelerra-gg.org/download/testing/cinelerra-Manual-reference.zip`

[Media files are licensed under a CC 4.0 BY, NC, SA license.]

[2]credit Armandux.

Start CINELERRA-GG to create an empty video track and two audio tracks (default). If you want you can create a new project (using the shortcut "n") with the desired settings or change the default one (`Settings → Format`).

1. Load multimedia files ("o").
2. `Settings → proxy settings`; mark *use scale*, select scale 1/2, and check *Auto proxy/scale loads*.
3. Wait until the progress bar at bottom right of the main window finishes to create and load proxy clips in `Resources → proxy`.
4. Move to the Resources and Multimedia sources window. Next select the sources, RMB click and *open mixers*. This creates new tracks and loads it. Be sure that the Proxy clips are in use by checking that the word "proxy" is included on the Timeline title bar.
5. `Audio → add track` (my master audio track is mono, because there are only voices and no music). Let us suppose this new track is audio 9.
6. Drag your master audio track to the new audio track (audio 9).
7. Open `Windows → mixers → Mixer Viewer`.
8. Click in the new mixer to select it (you will see a white box drawn inside that window) and afterwards expand the master audio track (audio 9).
9. Click into the expanded track (audio 9) and, while maintaining the new created mixer selected, click on the arrow on the left (*Mixer*) that appears when expanding the track, and click on it to get it pointing upwards. That hooks the new mixer with the audio 9 track (in our example). Next lock this track.
10. Select with the In/Out points tool ("[]") an audio fragment in audio 9 and afterwards click on "a" to select all the tracks.
11. With all the tracks selected (only the three top ones that are empty are disarmed), `Windows → mixers → align mixers`.
12. Select as master track your main audio track (audio 9 in our example). It is important to note the position of this track because you want to move all the other clips to the right to get aligned with the main audio track (9). In short: the selected fragment should be more to the right than the rest of this same fragment on the other clips/tracks. Move this track to the right if you need to. The idea behind this is to allow the other tracks to move to the right in order to get them aligned.

13. In the align mixer dialog select first *adjust*. Wait for the program do its work. You can see the progress bar at the bottom right of the main window.

14. Check the correctness of the alignment from column *R*; the values of the various tracks must be close to 1. Then click on the *apply* button. The tracks will move to the right and align.

15. In the main timeline in the audio 9 track, click on the *Play Track* button that is beside the *Arm track* button to enable it (if it is not already). Then enable audio and you can hear the audio while editing.

16. Position the windows mixers in a position you are comfortable with and next in the main window move the cursor to the beginning of the timeline. You can also resize the Mixers windows to your liking.

17. Click on play (space bar) and stop when you want to insert the portion of the track you want in the main track. You do this by double clicking in the mixer.

18. Go on starting, stopping and double clicking to insert new portions of the clips in the main track.

19. When you finish, arm only the main video track and your main audio 9 track.

20. Go to `tracks` → `delete tracks` and you will have only one video track and one audio track (in this example, the main audio track is mono).

And that is it.

One more useful thing to keep in mind is that you can edit the new main track and adjust the beginning or ending of every clip (Trim) without disturbing the audio and the position of the other clips. For example, if you are not happy with the cut between the three and the fourth clip.

So perhaps you want to have five more frames in clip three and five less in the fourth clip. Position the cursor arrow between the two clips and when the arrows shows, click the middle button of your mouse and drag in the direction you want to enlarge that particular clip and shorten the other one (trim mode: *Slip*). If you also press the shift button, in the Compositor window you see a double view of the neighboring edits on which the trim acts allowing for greater precision and usability.

To render, do not forget to uncheck the P (proxy) button in the main program window.

Note: If our files have Jam-syncing timecodes, we can speed up the workflow (steps 4 - 15) with the simple command : `Tracks → Align Timecodes`.

More info can be found in Align Timecodes.

D.3 Workflow with Keyframes and Plugins

One of the most powerful functions of CINELERRA-GG is the use of keyframes, which can be managed and applied for virtually anything you can do in the timeline, for example using them with plugins. See **??** for their in-depth description. Since they can be used in many ways, including very sophisticated ones, we present a workflow that can serve as an approach to the basic operation of keyframes applied to plugins[3].

Effects keyframing only works on the effect you are using, they are independently set for each effect you have added.

1. Put a multi-edits movie segment on the timeline, hit the `Ctrl-End` keys to get to the end and drag the effect(s) on to the whole timeline from the Resources window, then hit the `Ctrl-Home` keys to go to the beginning.
2. Enable keyframes autos button (`Generate keyframes while tweeking`) or hit `"j"`, make an adjustment to the effect and a keyframe will be generated automatically.
3. Move the cursor until it becomes an arrow at the junction with the next clip and left click to put the play cursor there.
4. Backspace one frame to the end of the first clip (hit `"4"` or `Alt-u` or click the `left frame reverse` transport button).
5. Adjust one parameter minutely and return it to its setting, one click on the mouse wheel rotation in each direction will do it, to set another keyframe which will hold the settings for the length of the clip. If you don't the settings will change as you adjust to set the next keyframe.
6. Go forward to the first frame of the next clip, reset the effect to set another keyframe, make any necessary adjustments.
7. Repeat as for the first clip and do that all the way down the segment. Remember to disable keyframes afterwards or you might inadvertently change settings while continuing to work.

[3] credit to DeJay.

It is always best to use the File pulldown and the option *Save As* after each operation e.g. Color Correction using (for instance) `Histogram`, `Saturation` and `Unsharp` plugins, then there is a point to return to if the next operation fails, or you want to make changes.

The final result is to have the first clip with a given plugin setting; the next edit with other settings; and so on throughout the timeline. The advantage is that you have only used the plugin once instead of adding a plugin to each edit.

It is preferable to put all the effects on the whole timeline and adjust them for each clip together, so you can see the overall result before moving on to the next clip. Effects are read from the top to the bottom of the stack and it is usually necessary to disable the effects lower down the stack while making adjustments. Using the example of Histogram, Saturation and Sharpen, disable Saturation and Sharpen while adjusting Histogram, or the Videoscope readings will be incorrect and so will the resulting settings, then enable Saturation and adjust that before moving to the next clip. Usually Sharpen is used as a *track effect*, meaning you don't keyframe it for each clip, keeping it disabled until the rest is complete.

D.4 Create a frame for a PiP

In CINELERRA-GG it is quick and easy to create *Picture-in-Picture* (PiP) using the Camera and Projector tools. You can read about how to use these at Camera and Projector.

PiPs created with Camera and Projector, however, do not have a frame. You can add it with the *Sketcher* plugin that allows for drawing on the canvas of the Compositor window. Such a frame remains fixed in the position where it was created and can only be moved manually, with a movement that must be in accordance with the movement of the PiP. Unfortunately, this is very complicated and imprecise work. A convenient workaround is to use the *Sketcher* plugin together with the *Crop&Position* plugin. Here is a description of how to do this.

Consider an example of a main video (`bbb_sunflower_1080p_30fps_ normal.mp4`) and a still image example (`Rodents.png`). You want to insert a crop of the image surrounded by a colored frame inside the video. Later you can move the PiP with its frame by using keyframes.

1. First, create a project with 3 video tracks. From top to bottom name them: `V3_Frame 4Picture`, `V2_Picture`, `V1_Video`.
2. In V1_Video put the video file. In V2_Picture put the image (extending it to the length of the video by simply dragging the right edge) - or you can put

another video instead of a still image. The V3_Frame4Picture track is left empty.

3. In V2_Picture, if needed put the *Scale* plugin, and then 2 separate *Crop&Position* plugins. The *Scale* plugin may be needed to scale the image - maybe to use it with the Camera tool, however it is not needed for the creation of the frame and PiP. Depending on its use, you can put it in the first or second position in the plugin stack. The first *Crop&Position* plugin (the topmost one, which is processed first in *Temporary* and whose output becomes the input to the next plugin) will be used only to make the desired crop for our image (Crop Only). The second *Crop&Position* plugin will be used to place the PiP in the desired position (Position Only).

4. In V3_Frame4Picture, select the length of the video track and apply the *Sketcher* plugin. Either highlight the region in which to apply the plugin or assign *In/Out Points*, because you can not add plugins to an empty track. Now add the *Shared Effects* of the *Crop&Position* plugin, taking care to choose the second plugin (Position Only) which is further down the stack (for shared effects see: Shared Effects and Shared Tracks).

This is what the timeline should look like:

```
-V3_Frame4Picture
        -> Sketcher
        -> Shared V2_Picture:Crop&Position (Position ONLY)
-V2_Picture
        -> Rodents.png
        -> Crop&Position (Crop ONLY)
        -> Scale
        -> Crop&Position (Position ONLY)
-V1_Video
        -> bbb_sunflower_1080p_30fps_normal.mp4
```

as you can see in figure D.9.

All that remains is to create the frame on the image that has already been cropped to the desired size. Drawing the frame with the *Sketcher* plugin is not difficult but requires precision. You can draw the 4 corner points imprecisely (`Shift + LMB`) and then arrange the created points by dragging them to exactly the right coordinates (`Ctrl+LMB`), or you can impose the coordinates on the 4 corner points by entering the exact numbers in the `X` and `Y Input` windows. You can also adjust the thickness of the frame and the color (see Sketcher).

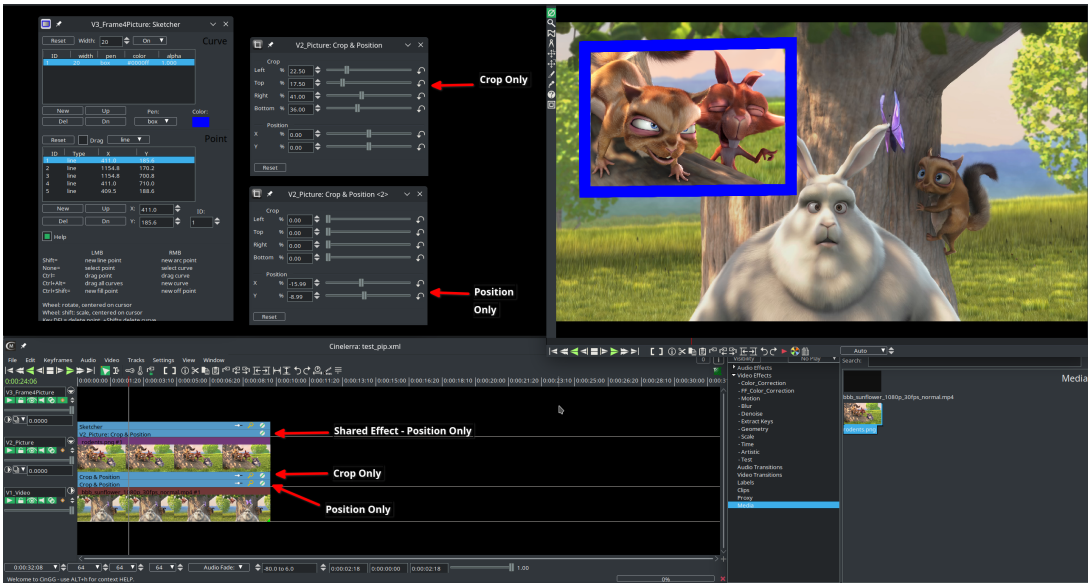


Figure D.9: *Setting effects*

If you want to move the PiP to another position you have to go back to *Crop&Position* (Position Only) and change the coordinates. Now you can see that the frame also moves along with the image thanks to the Shared effect present in the V3_Frame4Picture track, which returns the same coordinates to the *Sketcher* plugin.

Finally, you can animate the position of the PiP as well as the thickness and color of the frame throughout the video using the keyframes (see Keyframes).

D.5 Using Screen Capture on slower CPUs

Some results with different settings when working on slower CPUs follow:

- You can enable "loopback mode" in alsamixer, set xmms (with ALSA output) to play (any alsa-enabled app should work), and then you can record both video and audio. If your motherboard has no loopback switch for its integrated audio, you can use a specialized .asoundrc file set up as an alsa loopback instead – reference `https://bbs.archlinux.org/viewtopic.php?id=147852` for usage.
- If you leave recording settings to their default value of input frequency = 48000, you may get strange one-core cpu overload in kernel space. This will show up in the color orange if using the system monitor software, gkrellm (GNU Krell Monitors). So you most likley will want to set the input frequency to 44100 and then everything should work smoothly.

- Attempting to record 1440*900*24bit*30fps with cpu (AMD FX 430) set to its lowest frequency of 1.4Ghz, usually results in video being shorter than audio, so try slowing video down to different values, such as 0.68 or so via speed curve, and then just clip the few last silent frames.
- If you set the CPU up for performance and if you can rev it up to 4Ghz, then audio and video tend to be much more aligned in terms of their length. In one particular case with generally good results, the codec was mjpeg444 / s16le into a mov container on tmpfs.
- Specifically for screencapture on slow CPU, running short pre-session capture will be useful to see if you can get same length tracks with given resolution/fps/codec. And if not, either drop down recording fps or try to speed up encoder settings.
- In trying other positioning methods, apart from software timings, such as check/uncheck add/drop frames checkboxes and setting different number of audio samples ... , there seems to be no algorithm/code to intellectually duplicate frames that are too late in their encoding. And setting buffered frames in the device to absurdly high value like 50, was also not working for screencapture driver and short recordings like 20-25 seconds long. In conclusion, no amount of buffering will save you if you are chronically late.

D.6 Workflow for render yuvj (color range: jpeg) to yuv (color range:mpeg)

For historical reasons, almost every piece of video software uses *limited* (mpeg or TV) range. But sometimes you have to deal with files encoded utilizing the *full* (jpeg) range. For example, some video cameras encode full range files. The simplest way to deal with them is just to re-encode in limited range[4]:

```
    ffmpeg -i input.mp4 -c:v libx264 -preset fast -crf 19
-c:a copy -pix_fmt yuv420p output.mp4
```

However, this is not an option for you if you're either a perfectionist or trying to fight color banding. Color banding is a video artifact that could be visible in smooth areas of the video.

Here's an example: https://youtu.be/KL8CIZej19o

If you don't see the ugly background, then you don't need to worry about the whole issue. Color banding is different from compression artifacts. You can't fix it just by

[4] credit to Tarantas.

increasing the bitrate of your video. Color banding comes from color correction performed too much or in inaccurate way. And it accumulates, so if you have it, then you should get rid of dynamic range compression every step of the way.

CINELERRA-GG has a native full color range support, and you can enable it via:

`Settings → Preferences → Appearance → YUV color range: JPEG`

Also see: Color section and Color Space and Color Range Affecting Playback.

Note how everything will lose contrast in the Compositor window and become grey-ish. But, in this mode, CINELERRA-GG will also render full range video files. There are two reasons why it could be a bad thing:

- Some video players don't understand what full range is and will play these resulting videos with the increased contrast.
- Some image processing software (like YouTube) will re-encode full range videos into limited range by compressing the dynamic range and making the color banding issue even worse.

That's why you should make a couple of corrections:

- First, after your project is done, apply *F_colorspace* filter at the end (bottom) of the stack with these options: `range: 1` and `irange: 2`.
- Second, in `File → Render → Video wrench`, add this line at the end of the big textfield: `color_range=mpeg`

Since you're already fighting color banding, you should choose 32-bit colors via `Settings → Format → Color model`, the `RGB(A)-FLOAT option`. Note this could slow down your playback and rendering.

Appendix E

FAQ, Known Problems and Workarounds

Some workarounds for issues and a few known problems that have not yet been fixed, are described here. More workarounds can be found here.

Workaround for access to modifiable text files when using AppImage. This includes such files as FFmpeg video or audio files, fonts that you want to add for usage in the Title plugin, icons that you wish to modify, standalone programs, and any other text file for customization. To get access, first extract the files from the AppImage, find the files to add, edit, or modify, and then run the extracted binary following the steps shown here.

- /path-to-appimage/CinGG-20220131-x86_64.AppImage --appimage-extract
- edit or add the files you want to modify in the *bin* or *lib* subdirecctory
- /path-to-appimage/squashfs-root/usr/bin/cin

For more detailed information on working with AppImages, see B.10.1.

Workaround for the *alpha channel not working as intended when using an EDL* inserted as a reference or when an EDL is nested. Using the Projector on the EDL within a Master Project follow these next steps.

- Open your EDL with *Open EDL* by using the MMB (Middle Mouse Button) on the Nested edit on its track.
- In this nested EDL, insert a new Video track below the last video track.
- Add the Alpha plugin to the new Video track for the entire length of the nested EDL.
- Change the value of the Alpha to 0.00. The Alpha Plugin may be disabled or left enabled.

- Close the EDL.

This solves the alpha channel not working as expected and the Master Project now looks correct.

Workaround for using a transition, like Dissolve, between 2 edits with the plugin Motion51. In this case, the Motion51 plugin and the Dissolve transition are not seen during the transition phase and is stopped at the cut point between 2 edits. The left edit's plugin should be in effect until the end of that clip and should include the dissolve. Instead, there is a jump inside the dissolve, as if the plugin does not exist. There are 2 possible workarounds. For example, extend the Motion51 plugin to cover both clips rather than having a separate plugin for each clip. Or another workaround is to use two tracks and have the plugin cover the cut point and use a Fade auto instead of the Dissolve transition.

The Fade auto in certain circumstances produces an unwanted black flash. When you insert an effect on an empty track the fade fades as if it were on a black background; it interprets the alpha as black where there is transparency, which is what produces the fade. So the problem in the case of text is that the letters appear as a black flash briefly in the first few frames, then goes back to transparent, then the fade starts normally with the white color. To avoid the unwanted effect produced by the fade over black (because there is no content) leave the fade in/out of the effect at 0 and use the fade line to perform the intended results to avoid the problem with the alpha channel. The same problem occurs when using transparent PNG images with fade video transitions and can be resolved in the same manner. However this workaround can be quite tedious and complicated if you want to fade many snippets of text and transparent PNGs on the same track, with always the same video transition filter set to the same duration.

Just a side note here. Fade in/out in the Title plugin works fine if there is a clip/image above the plugin in the same track without an alpha channel but not if there is no clip/image. And the Transition effect works fine when there is a transition between two clips/images without an alpha channel.

Detailed explanation:[1]

If we apply the `Dissolve` transition, what should actually happen mathematically? For channels r, g, b the contribution of the 1-st clip should gradually fall from 100% to 0%, and the contribution of the 2-nd rise from 0% to 100%. If we have no alpha channel, this is quite simple. If we have, perhaps the alpha channel should gradually change from the a-value of the 1-st clip α_1 to the α-value of the

[1] Thanks to sge

second (α_2). For example, if the 1-st clip is a PNG image with opaque letter over some considered pixel, and the 2-nd clip is a PNG with a completely transparent background, the α-value should gradually fall from 100% to 0% for that pixel.

We can easily look with our eyes which color has an area with complete opacity ($\alpha = 100\%$). r=g=b=100% is white, 0% is black, 100/0/0% is red, etc. But what color is hindered under complete transparency? When $\alpha = 0\%$, we will always see black. But actually it could be red, or even bright white, but completely masked out by 0% alpha.

Now let's apply our mathematics with channels. Somewhere in the middle of the Dissolve transition we should get about 50% of alpha, if $\alpha_1 = 100\%$ and $\alpha_2 = 0\%$. But what for rgb?

If rgb_1 was 100/100/100% (white), and rgb_2 was also 100/100/100% (fully transparent white), then we will have rgb=100/100/100%, $\alpha = 50\%$. This will look as medium grey, 50% of white.

If rgb_1 was 100/100/100% (white), but rgb_2 was 0/0/0% (fully transparent black), then we will have rgb=50/50/50%, $\alpha = 50\%$. This will look as rather dark grey, only 25% of white.

If rgb_1 was 100/100/100% (white), but rgb_2 was 100/0/0% (fully transparent red, but visually still looking as black on black background), then we will have rgb=100/50/50%, $\alpha = 50\%$. This will look not as grey anymore, but a bit reddish.

So, the 0% alpha channel can hinder any actual color of the masked picture, which can become visible if alpha rises above 0% due to applying some transition of effect.

Do not change Color Model Format in session just before loading backup. Under certain circumstance if you change the Color Model in Settings->Format, quit the session, and then "Load backup" when you start again, it will crash. Workaround is to avoid this scenario.

Audio tracks align with no video track, and in working with Mixers. The use of `Open Mixers` always requires the existence of a video track. See But, I want to use only the first set of audio tracks... We cannot act on audio tracks only, for example to align them. A workaround to get the audio tracks aligned would be:

1. You create/open the *Mixer Viewer* with `Open Mixers` in Resources window for the Videos with Audio. It will create, in the timeline, one Video track and two Audio tracks for each Media (Video/Audio), and its Mixer Viewer.

2. Add a new Video track (`Shift+T` shortcut) and two Audio tracks (`t` shortcut) for your audio (Left+Right channels?) without video: you are adding an empty Video track because `Align Mixers` feature needed a video track.

3. Move the Video track down until it is on the top of the two Audio tracks.

4. Insert your Audio into the two Audio tracks in the usual way.

5. Open/Create a Mixer Viewer from `Menu` → `Window` → `Mixer...`→ `Mixer Viewer` (`Shift+M`).

6. Assign that Mixer Viewer (a white frame is shown when selected) to the one Video tracks (empty) and two Audio tracks, of the #2 point, pressing the arrow icons (`Mixer`) you find on the Patchbay on the left of the Main window. When assigned, you must to see the 3 arrow (for 1video + 2Audio) are pointing up.

7. Now, you can use `Align Mixers` feature as usual. See Audio/Video sync using Waveforms/Timecodes

8. If you use the Viewer to make a clip and then try to "Open EDL" on that clip in the Resources Clip folder window, it will crash. Always use Open EDL with media on the timeline Main Program window.

Glossary

8 / 10-bit images images that were quantizated at 8 or 10 bits per channel. The tone range for each channel is from 0 to 255 for 8-bit and from 0 to 1023 for 10-bit.

Algorithm set of instructions, typically to solve a class of problems or perform a computation.

Aliasing incorrect sampling of a video or audio signal that leads to unwanted artifacts because it cannot distinguish values too close. There is Temporal Aliasing (audio) and Spatial Aliasing (video).

Alpha channel an extra channel in color models for storing information about transparency. If a pixel has a value of 0% as its alpha channel, it is completely transparent or invisible; a value of 100% as its alpha channel provides a fully opaque pixel.

Arm when you *arm* a track, it is enabled so fully affected by editing operations. Tracks are armed with an `Arm Track` button in the patchbay of the timeline.

Arrow mode formally known as Drag and Drop editing mode. Drag and drop editing is a quick and simple way of working in CINELERRA-GG, using mainly the mouse. In this mode, you move *things* by dragging and dropping these resources or objects.

Aspect ratio in reference to an image, this is the geometry of the data as in the ratio of its width to its height. There is the display aspect ratio which is the physical display and may not always be square. Then there is the pixel aspect ratio or geometry of the picture.

Assets the representation of media loaded in the Resources Window and on the timeline. These assets are the objects of the editing. You can think of an asset as the original file that was loaded from your operating system's disk but it also represents clips and nested clips. These are data objects in the program that contains all the parameters it takes to read or write that media.

Audio offset audio displacement applied to the playback position to synchronize playback with other tracks.

Automatic keyframes every time you tweak a key-framable parameter it "automatically" creates a keyframe on the timeline. Since automatic keyframes affect many parameters, it is best enabled just before you need a keyframe and disabled immediately after. Effects are only keyframable in automatic mode because of the number of parameters in each individual effect. Enable automatic keyframe mode by enabling the automatic keyframe toggle on the timeline editpanel.

Autos short term for automation keyframes. Autos are created by clicking on an "automation curve" represented by a colored line, to establish the time position for the new

keyframe anchor point. They are usually drawn as a colored square or a symbol on a media track.

B-spline (basis spline) polynomial curves characterized by nodes and control points to obtain smooth curves.

Banding Incorrect display of a color gradient, due to a narrow tonal range, which leads to the appearance of color bands instead of fading tones.

Bayer array is a color filter array upon a sensor to get an RGB image. The filter pattern is 50% green, 25% red and 25% blue.

Bezier curve a parametric curve used in computer graphics. In image manipulation programs, bezier curves are used to model smooth curves.

Bicubic filter is a mathematical interpolation to resample an image. It produces a smoother result than the nearest-neighbor or the bilinear filter.

Bilinear filter is a mathematical interpolation to resample an image. It produces a smoother result than the nearest-neighbor.

Bit depth Quantization of an audio or video signal. Color depth is the number of bits used for each color component of a single pixel. Audio bit depth is the number of bits of information in each sample.

Bitrate represents the amount of information that is stored per unit of time of a recording. The higher the bitrate, the higher the quality. When setting Bitrate in rendering, be sure to not set any Quality or CRF values because they take precdence.

Black point the part of the image with the darkest value that can be displayed on a device. At the limit is 0 (pure black).

Brightness human perception of the amount of light emitted by a source; overall lightness or darkness of an image. In video signals it is represented by luma. The measure of brightness is value.

Buffer a region of memory used to temporarily hold output or input data, used when there is a difference between the rate at which data is received and the rate at which it can be processed, or in the case that these rates are variable. Buffers are allocated by various processes to use as input queues, etc. A simplistic explanation of buffers is that they allow processes to temporarily store input in memory until the process can deal with it.

Bug a defect in the program that causes it to not function correctly thus creating some kind of error. There are basically 2 kinds of bugs – doing something it should not or not doing something it should.

Canvas the place on the compositor where the final video is displayed. It can be imagined as the canvas of a painter or as the screen of a movie theater.

Chroma the signal used in many video systems to carry the color information of the picture separately from the accompanying luma signal. Luma represents the achromatic image without any color (like a black and white picture), while the chroma components represent the color information.

Clipping to set a data value above an threshold to be equal to that threshold.

Clip a short segment of media which was usually part of a longer recording. A section of a project consisting of a set of timeline selected edits treated as an independent object.

Codec contraction of **co**der and **dec**oder; used to generally compress media and decompress to reproduce it. Results are a trade-off between bitrate and quality.

Color correction Same as color grading. Some people use it with slightly different meanings: minimizing error between the image and the original object. Corrects exposure and color dominants. There is the primary CC, which covers the entire frame; and the secondary CC, which isolates and intervenes only on specific parts of the frame.

Color grading Same as color correction. Some people use it with slightly different meanings: Correct the color and contrast of a shot to personalize it and make it more nicer or communicative. An example is the Teal & Orange, which is widely used in Hollywood movies.

Color model an abstract mathematical model describing the way colors can be represented as an ordered list of numbers, typically as three values or color components (for example, RGB).

Color space organization of the colors of a color model, limited to the gamut of a particular device. Examples are sRGB and rec 709.

Color timing the process of adjusting color balance making it consistent in every scene. We also talk about color matching scene to scene or shot matching.

Composite the combination of two or more video layers to obtain a single composition.

Compression in CINELERRA-GG's dialogs means compression format. See Codec. In a media video the compression can be Intraframe (within each frame) or it can be about Groups Of Pictures (GOP and Long GOP) between frames, i.e. Interframe. Also, the compression algorithm can be lossy or lossless. Lossy and Interframe are also fine for editing, but for Color Correction, Chroma-Key and Rotoscoping, lossless and Intraframe compression is required.

Concatenate in File→Load insertion strategy, means to load from disk or copy from the timeline edits belonging to different media onto the same set of tracks, one right after the other.

Context menu a pop-up menu.

Contrast is the difference in luminance of parts or elements of an image that makes them distinguishable. The greater the difference, the greater the contrast. It is the horizontal range shown by the histogram or the vertical range shown by the waveform.

Control point as applied to Bezier curves, either the *end* points or *way points*. Controls the derivative at that end or way point of that curve. Vector of the curve point derivative.

CR2 the RAW image format of Canon's digital cameras. In CINELERRA-GG, it is used loosely to refer to any brand of camera raw images or still pictures, as handled by Dave Coffin's included code.

Crop rectangularly trim edges to remove unwanted material.

Cut and Paste editing an editing mode on the timeline. You select the cut and paste editing mode by enabling the I-beam toggle on the control bar at the top of the main program window. The I-beam cursor pointer on the timeline is why this mode has the nickname of I-beam. In this mode you can copy edits in the same track, copy from different tracks in the same instance, start a second instance of CINELERRA-GG and copy from one instance to the other or load a media file into the Viewer and copy from there. There are many other operations that can be done.

Cut delete or trim to remove material or insert a break so that the media on the timeline is separated into 2 pieces. Cut, when used as a noun can mean the same as an edit.

Deinterlace the process of converting interlaced video (a sequence of fields) into a non-interlaced form (a sequence of frames). Deinterlacing downsamples/blends the refreshes that reconstruct images with the least motion damage. See the definition of Interlace for an explanation of interlacing.

Denoise is the process of removing digital noise from a video. The three main types are based on statistical methods, transform wavelets and temporal averaging.

DeSpill process to remove background color contamination from the edges of the subject in foreground, during a chroma key.

Disarm when you "disarm" a track, it is disabled so not affected by editing operations. Disarming a track protects it from most changes and operations. Tracks are disarmed with an "Arm Track" button in the patchbay of the timeline.

Drag and Drop editing drag and drop editing is a quick and simple way of working in CINELERRA-GG using mainly the mouse. In this mode, you move "things" by dragging and dropping these resources or objects. To enable, click on the arrow icon in the timeline editpanel. This arrow is why this mode has the nickname of arrow mode.

Dynamic range is the ratio between the largest and smallest values (luminance) that an image can assume. The larger the size, the better we can distinguish details in the dark and light parts.

Edit Decision List (EDL) an XML text file that contains all the project settings and locations of every edit. Instead of media, it contains pointers to the original media files on disk. If you open your .xml project file with a text editor, you will see some of the terminology used in CINELERRA-GG in reference to the editing you perform.

Edit panel the second line on the timeline which shows icons that represent transport controls and editing functions. It is sometimes called the Control Bar.

Edit(s) is a fragment of media on a single track of the timeline.

Exposure the exposure is the amount of light per unit area reaching an image sensor, as determined by shutter speed, lens aperture and scene luminance.

Field in interlace, the scan of every second line. An interlaced video frame consists of two sub-fields taken in sequence, each sequentially scanned at odd and then even lines of the image sensor.

File format generally refers to a block network stream format that multiplexes multiple audio/video streams to create presentation output.

Filter a program to process a data stream, often it is used in referring to effects or plugins. Many people use this in referencing the included FFmpeg plugins since that is what it is called in ffmpeg.

Floating point real numbers with decimals. They allow for greater precision in calculations than integers, but they require more processing power.

Footage in filmmaking and video production, footage is the raw, unedited material as it had been originally filmed by a movie camera or recorded by a video camera, which usually must be edited to create a motion picture, video clip, television show or similar completed work.

Frame any image in a sequence of images that form an animated video.

Gadget a composition of widgets. See widget definition.

Gamma color and contrast shaping using an exponential math function to enhance or retard color contrast, either color or brightness.

Gamut In color reproduction, the gamut is a certain complete subset of colors. The larger the gamut of a device (associated with a color space) the more colors can be displayed.

GDB Gnu DeBugger, a Linux program for testing and some help in finding the errors in another program.

Git is a distributed version control system for tracking changes in source code during software development. It is fast and easy to work with git repositories for collaboration among several developers on the same software project.

GPU stands for the Graphics Processor Unit of a computer graphics board. For CINELERRA-GG, direct decoding or encoding using the GPU via hardware acceleration often reduces CPU usage.

GUICAST CINELERRA-GG's GUI library, made from scratch by Heroine Virtual Ltd.

GUI Graphical User Interface allows people to interact with the computer by manipulating graphical icons, visual indicators or widgets, along with text labels or text navigation to represent the information and actions available to a user. This is in contrast to Command Line Interface (CLI).

Handle is a graphical grab point to adjust any number of different graphical parameters, such as an edit position or a mask curvature. The handle becomes active when approached within a certain range, then you will generally see a modified cursor indicator. This is the hot point to grab and click to use it.

HDTV (high definition TV) standard characterized by a 16:9 aspect ratio, various frames rates and scan modes and with a resolution of at least 1080.

HSV Hue, Saturation, and Value is a color model that is often used in place of the RGB color model. In using this color model, a color is specified, then white or black is added to easily make adjustments.

Hue that aspect of a color described with names such as yellow, red, blue. Hue also defines mixtures of two pure colors like "red-yellow" or "yellow-green".

I-beam mode formally known as Cut and Paste editing mode where a lot of the operations performed on your video are with various copy commands. See Cut and Paste editing definition.

Image list a text file with a specific format containing a list of absolute paths for the still images of a sequence plus additional information like file format, framerate and image resolution. Image lists are human readable and editable. Once loaded in the timeline, image lists behave like a video clip. In CINELERRA-GG they can be used to load multiple images belonging to the same scene as a single video asset. CINELERRA-GG can render video clips to image lists (a text file + multiple still images).

Index file an .idx, .mkr or .toc file built by CINELERRA-GG in the .bcast5 directory of your home folder in order to quickly, and with less cpu involved, seek into big media files, for skipping, faster playback and drawing waveforms and picons. Index files are not human readable. If an index file for an asset is already built, it is not recreated – this saves a lot of time. However if you switch from a native format to using ffmpeg, or vice versa, you should always rebuild the index. The number of index files to keep can be set by the user and you can easily delete all of them at once from the Settings→Preferences, Interface menu.

Insertion point a flashing hairline mark that vertically spans the timeline. It marks the starting place of the next operation to be performed.

Interlace (extended definition included here because it is no longer a commonly used technique). Originally media was conditioned for presentation on a cathode ray tube (CRT), like an old-fashioned television set. The image sample was scanned using an image orthicon tube and vacuum tube oscillators. The result is very imprecise and too slow for the human eye. Interlacing, which is scanning the image by fields, first all the even and then all the odd lines, uses phosphor persistence to create a smooth presentation effect and reduce the required signal bandwidth. Today we use a progressive scan at a higher framerate to avoid these issues.

Intermediate The production process of a movie in which you switch from film to a digital intermediate for the various stages of processing and then return to the film. (ADA = Analog, Digital, Analog). The most used formats are Cineon/DPX and OpenEXR. We also talk about DI= Digital Intermediate. Nowadays it is often used as a synonym for Mezzanine codec (see).

Interpolation a method of inventing data using keyframe way points on a curve. The shape of the curve represents the type of the interpolation, for example piecewise linear or bezier. In CINELERRA-GG interpolation is the process that inserts many new values in between the user defined keyframes to give a smoother result.

Jitter broken sound or video, frequently due to quantimization encoding, resulting in data loss.

Keyframe a blob of parameter data associated to a position on the timeline. It represents a certain value set by the user at a certain point in the timeline and is used as input to some edit functions such as a fade, an effect, or a compositing parameter. Keyframes are described in detail in the Keyframes section, chapter 8.

Keyframing a very convenient technique for creating smooth dynamic changes by assigning values to parameters at specific moments in time and letting CINELERRA-GG interpolate the values in between.

Lanczos algorithm for high quality resampling video signal. It is also used in case of upsampling, weak point of other similar filters.

Latency refers to a short period of delay, usually measured in milliseconds, between when an audio signal enters a system and when it emerges.

Letterbox blacks bars on the top and bottom side of the frame. They are due to a smaller frame size than the one set in the project (see also Pillarbox).

Lossless term describing a compression method that allows the exact original data to be reconstructed from the compressed data, without any changes. Lossless compression is used for text and data files, but also for multimedia when quality is more important than file size.

Lossy term describing a compression method where compressing and then decompressing retrieves data that may well be different from the original, but is close enough to be useful in some way. Lossy compression is most commonly used to compress

multimedia (audio, video, still images), especially for streaming. Repeatedly compressing and decompressing the file will cause, for most lossy compression formats, to progressively lose quality. MP3 is an example.

Luma/Luminance Luma represents the achromatic image without any color (like a black and white picture), the part of a video signal that includes information about its brightness. Luma is typically paired with chroma - luma represents the image without any color, while the chroma components represent the color information. Luma is designated with the letter Y.

LUT, 3D LUT (LookUp Table) used to map one color space to another. CINELERRA-GG uses them through ffmpeg filters. There are downloadable collections or there are specific ones provided by hardware manufacturers.

Media generic term for audio, videos and images on some kind of storage. This can include items as a short movie recorded on your phone, photos from your camera, MP3 songs, or a movie trailer.

Mezzanine codec Transcode digital footage files into a high quality codec that is more efficient on the timeline and retains all initial information (is a DDD process, i.e. Digital, Digital, Digital). Generally such codecs are little or nothing compressed; if compressed they are lossless and intraframe and the chrominance is not less than 4:2:2. Very used codecs are DNxHD; ProRes; ffv1; HuffYUV; etc.

Nearest neighbor is the simplest method of resampling an image. It is fast and resources saving, but produces less smooth results.

NLE Non Linear Editing. A modern editing method used by CINELERRA-GG that records the decisions of the editor in an edit decision list (EDL) without modifying the original source files.

NTSC stands for National Television System Committee; a standard that defines a video with canvas size of 720x480 and a framerate of 29.97 fps. This was originally based on United States broadcast television.

On the fly consumed as produced in real time.

PAL stands for Phase Alternating Line; a standard that defines a video with canvas size of 720x576 and a framerate of 25 fps. This is based on a European standard for broadcast television.

Panning in video technology, panning refers to the horizontal scrolling of an image wider than the display. In CINELERRA-GG it is done (together with other camera movements) with the camera tool.

Patchbay the area on the left of the timeline that contains the controls to enable features specific to each track.

Patch an incremental change of any kind of program source.

Picons miniature images, also called thumbnails, of the video. In the Resources Window they represent the first frame of the asset. On the Timeline they imitate a physical film and are derived from the video data in the media file.

Pillarbox blacks bars on the left and right side of the frame. They are due to a smaller frame size than the one set in the project (see also Letterbox).

Pillarbox blacks bars on the top and bottom and/or left and right side of the frame. They are due to a smaller frame size than the one set in the project.

Pixel the smallest independent unit of a digital image; a minute area of illumination on a display screen, one of many from which an image is composed. Word was invented from picture element. In the 3D world, this is call a voxel.

Plugin program fragment that is loaded on demand as in "plugged in". In CINELERRA-GG, these are often called "effects" and are used to provide a specific function on demand.

Pop-up menu a menu that pops up by clicking the right mouse button. Also called a context menu.

Progressive scan a method for capturing, storing, displaying or transmitting moving images in which the lines of each frame are drawn in sequence, in a path similar to text on a page - line by line, from top to bottom. It is in contrast to the "interlace" used in traditional television systems.

Project consists of the EDL, media, and other associated data objects to comprise the entire session for the purpose of rendering/creating the final result.

Proxy file a copy of an original media file but with low resolution or quality, used as temporary media for editing with lower CPU load or I/O. Rendering is then done with the high quality original.

Raw image an image file containing the unprocessed data from the image sensor of a digital camera or a scanner. Raw images from many different cameras can be loaded in CINELERRA-GG. There are quite a wide-range variety of raw formats in existence.

Rec 709 (BT.709 or ITU 709) is the standard color space of high-definition television, having 16:9 aspect ratio, scan modes and frame rate of HDTV. It can decode at 8 or 10 bits per channel. The gamut is the same as for sRGB from which it differs for the 2.4 gamma.

Record the acquisition and storage of some media

Render farm a set of computers that work closely together for rendering.

Rendering the process of applying the instructions contained in the edit decision list (EDL) to produce an audio and/or video file. Also any processing of video, such as in rendering effects to the compositor.

Resize to reduce, enlarge or reshape the outline of an image, thus preserving the relative measures of the distances inside it.

Resolution the size of a digital image, width and height, measured in pixels.

Resource(s) the actual media and support items available for your projects which includes assets, clips, transitions and effects. These are available in the Resources window when needed.

Retiming is the change of the speed of a edit by interpolating the original frames to a new in between frame.

RGB (Red, Green, Blue) a color model in which red, green and blue lights are added together in various combinations to reproduce all the colors. RGBA is the same color model with extra information for transparency in the Alpha channel. This mimics the receptors in the human eye – that is why we use it

Saturation the intensity of a specific color. It measures the distance of a color from a neutral gray.

Scale to reduce or enlarge an image proportionally, preserving the ratio of the distances inside it. See resize.

SEGV Segmentation Fault Violation - a page fault that fails to read or write memory. For example, attempting to write to a read-only location, or to overwrite part of the operating system.

Shell a file containing a series of commands that provides an interface for users. In everyday use it indicates the Command Line. CINELERRA-GG has a "shell cmds" icon on the upper right corner of the main timeline where user written shell scripts can be added and easily accessed without exiting.

Shot in filmmaking and video production, a shot is a series of frames, that runs for an uninterrupted period of time. In film editing, a shot is the continuous footage or sequence between two edits or cuts. Loosely used to refer to a single camera image, i.e. a shot.

Solo to activate only one function of a functional set.

Source a file containing media that has been saved on your operating system disk in a file. Anything that provides data; origin of data.

Splice to unite edits by lapping the two ends together or by inserting an edit between two edits or in the middle of an edit.

Split to divide or break up an edit into two parts.

sRGB is an RGB color space used on monitors, printers, and the Internet. It can decode at 8 or 10 bits per channel. The gamut is the same as for Rec 709 from which it differs for the 2.2 gamma.

Streaming a method of receiving audio or video media while they are still being delivered, so that it is possible to watch video or listen to audio without waiting for an entire file to download. Streaming can be live or on-demand.

Stream a source of data that is usually accessed only sequentially, as in audio or video.

Subsampling is the practice of encoding Y'CbCr images by implementing less resolution for chroma information than for luma information. Depending on the amount of information deleted we have: 4:4:4 (lossless); 4:2:2 (HDTV); 4:2:0 (DVD, smartphone) or 4:1:1.

Subtitle the text of a video displayed on the bottom of the screen, often used for language translations. Subtitles are also the graphics displayed on top of video content, stored in a separate stream such as done for DVD menus.

Telecine a method of sampling media for both color and sample rate in order to prepare for presentation on a broadcast signal. In the United States video is broadcast at 29.97 frames per second but film uses 24 frames per second. For the film's motion to be accurately rendered on the video signal, a telecine must use a technique called the 2:3 pulldown, to convert from film's 24 frames per second to 29.97 frames per second.

Thread a single stream of program instruction fragments that may or may not be executed in parallel.

Thumbnails miniature images of the video. In the Resources Window they represent the first frame of the asset. When drawn on the timeline they imitate a physical film. See picons.

Timebar the part of the timeline that marks the time passing in selectable time units.

Timecode is a sequence of numeric codes generated at regular intervals by a timing synchronization system and recorded into audio and/or video tracks. It is used for synchronization audio and video clips.

Timeline the part of the program window that contains video and audio tracks and displays the edits as they occur in time.

Title bar presentation of a title for a given data object.

Title identifier applied to a data object; it is used to refer to the bar in the upper part of an asset that contains the name of the source file. It can be shown/hidden using the View menu. In the XML project file, TITLE is the name of the track, for example "Video 1".

TOC acronym for Table of Contents; this is one of several kinds of an index file used to accelerate media access. It may contain compressed audio waveform data to accelerate timeline update.

Toggle a program switch.

Transitions rendered output.

Trimming edit boundaries to lengthen or to shorten the duration of the edit in the timeline. Over the edit boundary and during the trimming operation the mouse pointer changes shape.

Tumblers a diamond shaped button composed of two arrows - one arrow upward and one arrow downward. It is used to set values in text boxes using the mouse either by clicking on the up/down arrow or using the wheel with the pointer over the tumbler.

Tweaking to make minor adjustments to. Also, a general term from when automatic generation keyframes are armed.

Two screen editing this refers to an editing method where you mark a portion of your source material with In and Out points and insert or overlay at a specific and marked point in your timeline, called the insertion point.

Underrun a state occurring when a buffer used to communicate between two devices or processes is fed with data at a lower speed than the data is being read from it. This requires the program or device reading from the buffer to pause its processing while the buffer refills.

Value the measure of the brightness of a color. In video signals it is represented by luma.

Vicons stands for Video Icons, which are animated thumbnail presentations of video media.

Waveform the visual image of the form of the audio signal.

White point is the clearest part of an image that can be displayed on a device; at the limit is 1.0 value (pure white).

Widget a unitary graphical object that performs a specific set. It is usually a single subwindow.

XML the language CINELERRA-GG EDLs are written in. Extensible Markup Language (XML) is a general-purpose language that combines text and extra information about the text and allows users to make modifications. It creates a text representation of a data object that is designed to be relatively human-legible.

YCbCr is a color space based on the YUV color model, widely used in broadcast productions. It separates the luma part (Y) from the chroma part (Cb and Cr).

YUV is a color model which splits luma from chroma, similarly to human sight.. Colors are stored as absolute luma representation and the difference signal between luma and chroma complement components. This color model is used by PAL and NTSC standards. Y stands for the luma component and U and V are the chroma components. U and V are actually color difference components (respectively R-Y and B-Y). In fact YUV signals are created from an original RGB source. The weighted values of R, G, and B are added together to produce a single Y signal, representing

the overall luma. The U signal is then created by subtracting the Y from the blue signal of the original RGB, and then scaling; V is created by subtracting the Y from the red, and then scaling by a different factor. Previous black-and-white systems used only luma (Y) information and color information (U and V) was added so that a black-and-white receiver would still be able to display a color picture as a normal black and white picture.

List of Tables

1.1 List of thirdparty builds 17

2.2 Examples 88

5.3 ShuttlePROv2 key default arrangement for CINELERRA-GG / Composer / Viewer 168
5.4 ShuttleXpress key default arrangement for CINELERRA-GG / Composer / Viewer 169

6.1 list States of the Proxy icon 172

10.1 Sketcher controls 402
10.2 Titler attributes 410

List of Figures

1.1 Screencast of the website Download page for installing CINELERRA-GG for various O/S. . . . 4
1.2 Screencast of an Android tablet running CINELERRA-GG using TERMUX. . 29

2.1 The four windows (cc-by-sa Olaf) . . . 33
2.2 Patchbay | Timeline with pulldowns, navigation icons, Video/Audio tracks | bottom Zoom Panel . . . 34
2.3 Goto position menu . . . 36
2.4 Insertion point is at 0:00:25:10 in Hr:Mn:Sec:Frames . . . 40
2.5 I-beam + in/out + labels . . . 40
2.6 An example of the Autocolor assets . . . 45
2.7 Window Layouts . . . 47
2.8 Shown are 4 panes that have split in X and Y the main track canvas . . . 48
2.9 Left hand side are the toolbar functions / bottom bar has many control functions . . . 50
2.10 Compositor window zoom slider bar and scroll bars . . . 53
2.11 Note the black outlines showing the safe regions. Also note the Ruler menu 55
2.12 Compositing pipeline . . . 58
2.13 Color3way on Temporary . . . 59
2.14 Camera and Projector tool . . . 61
2.15 Compositing pipeline with mask . . . 63
2.16 Mask options window . . . 65
2.17 Crop menu and outlined crop rectangle on the right side . . . 72
2.18 Viewer Window - the red arrow "Play" button is left of the Videoscope button 73
2.19 Viewer window at the top displaying same 5 seconds as seen in the Resources window thumbnail. At the bottom of the screen is the audio loaded on the timeline. . . . 74
2.20 The mouse cursor, above the green colored 'play forward' transport button, is on the timebar. Further to the right we see the red "indicator line". . . . 75
2.21 A left-facing arrow on the right side of the reddish slider bar is used to drag the bar. . . . 76
2.22 Here you can see the right-facing arrow used to drag the other end of the slider bar. The black area is the actual preview area. . . . 76
2.23 Note the double-headed fat arrow in the black preview area used to move the selection over. . . . 77
2.24 Folders are in the first column with contents of that folder on the right hand side . . . 78
2.25 Info windows with the 'Detail' box . . . 81
2.26 New folder menu . . . 82
2.27 Folder name . . . 82
2.28 The 'master' Media folder . . . 82

2.29 Modify Folder . . . 83
2.30 The available choices for each field. Value field missing. . . . 84
2.31 Note "Full Play" mode and Vicons and Aicons in Media folder . . . 89
2.32 Draw Vicons | Screenshot display various audio file lengths; red is shortest. 89
2.33 The location of the Preview/Draw Icons mode. . . . 91
2.34 Note the line through the mode. . . . 91
2.35 Example of copy file list . . . 92
2.36 Snapshot menu and choices . . . 93
2.37 Grabshot reticle & orange box . . . 93
2.38 Show Overlays window on the left with the Color ball window to the right to set color . . . 96
2.39 Lines are colored here on the timeline as designated in Show Overlays . . . 97
2.40 Original Settings — cool spot . . . 98
2.41 Note Titles box hot spot . . . 98
2.42 Cam/Proj XYZ toggle to fine tune . . . 98
2.43 Sound Level Meters Window . . . 98

3.1 Set Format window - note the Audio Channel positions . . . 100
3.2 New Project dialog window . . . 101

4.1 Screenshots for RAW images . . . 113
4.2 Load file menu. Note the green checkmark for OK and the middle Apply option . . . 113
4.3 Load - Sort by File name, sort by file Size, and within Extension after a previous Size sort . . . 117
4.4 Load windows with various Numeric Sizes . . . 118
4.5 Three example of Probes window . . . 119
4.6 Multiple program streams and Asset Detail . . . 120
4.7 Export Project option popup and the 3 available options. . . . 122

5.1 Timeline editing session . . . 127
5.2 Patchbay . . . 129
5.3 Video Overlay, audio Pan and Nudge. . . . 131
5.4 Gang None: only the video track is master. We see all tracks both master and non-master (default) . . . 134
5.5 Gang Channels: only the video track is master. Let's see the master tracks and the first of the non master tracks . . . 135
5.6 Gang Media: only the video track is master. We only see the master tracks 135
5.7 Blade cut . . . 143
5.8 Inter-View mode: white bar→source; red bar→timeline . . . 153
5.9 Inter-View mode and the timeline . . . 155
5.10 Edit Length window . . . 156
5.11 Align edits . . . 157
5.12 Selected area for Reverse Edits . . . 157
5.13 Results of the Reverse Edits . . . 158
5.14 Shuffle edits: the edits are permutated . . . 158
5.15 Shown here are 3 Viewer windows and the *View in new window* popup . . . 159

5.16 ShuttlePROv2 and ShuttleXpress 160

6.1 Proxy settings dialog 174
6.2 Once you have an Open EDL, the easiest way to close it. 181
6.3 Default choices for mouse: Ripple for LMB; Roll for MMB; Slip for RMB . 184
6.4 The 5 types of Trim: note the different lengths of the results. 187
6.5 J-cut to left and L-cut to right 187
6.6 Split compositor screen showing the result of the Trim feature 188
6.7 Nested clips in Timeline and Resources window 190
6.8 Using Mixer capability in Cin for multiple cameras 192
6.9 A Mixer window showing available options with the white X designating no playing 197
6.10 Mixers... pulldown 198
6.11 Mixer patchbay 199
6.12 Align Mixers window 200
6.13 Aligned mixer window after "Match". Note the Nudge amounts above. . . 202
6.14 Match setup for aligning by audio. Note that [ ] are set over a sample waveform highlighted selection that includes that. 203
6.15 An audio Match is complete. Note the waveform is aligned. 204
6.16 Before sync 205
6.17 Clapperboard button 206
6.18 After sync 206
6.19 Timecode via MMB on track and via Resources windows/Info 208
6.20 Subtitle window 210
6.21 Subtitles on timeline 211

7.1 Example of the Render menu 213
7.2 Audio and Video tracks automatically checked for Pro file type 214
7.3 Render menu displaying a PNG *one frame* option 215
7.4 Render & Video Preset menus displaying Pixel choices 216
7.5 Render menu showing where Samples is 217
7.6 Example of the Batch Render menu 229
7.7 New Buttons with Unsafe GUI in batchrender 233
7.8 Batch render with the 4 ghosted buttons on the right side + the Warning message below 234
7.9 Settings Background Rendering 236
7.10 Timeline with the top red bar 236
7.11 Settings: Preferences: Performance tab, menu to set up your Render Farm 239

8.1 Automation Audio Fade menu with tumbler 251
8.2 Fade curve with pink colored box Keyframe and Slider bar 252
8.3 Fade Auto with *Always* set for easy to see numeric values 253
8.4 Fade Auto with Controls 253
8.5 Ganged Speed Auto on all tracks and Speed auto with orange Slider bar . 255
8.6 range from left keyframe to right keyframe 256
8.7 Bump autos creation 257
8.8 Speed autos slider 257

8.9 Keyframe Parameters window for a Color 3 Way plugin added to the video track . . . 259

9.1 The three icons of *FF Probe* . . . 264
9.2 FFmpeg wrench, video preset, view and format options . . . 272
9.3 FFmpeg wrench, audio preset, view and format options . . . 273

10.1 Screencast of the native Video plugins in the default Cinfinity icon set. . . . 275
10.2 Cinfinity2 audio plugins . . . 276
10.3 Screencast showing the screen to change your plugin icons set . . . 276
10.4 Screencast showing the menu where Editing Effects behavior is changed. . . 278
10.5 Dragging the Color 3 way effect . . . 279
10.6 Screencast shows 4 Factory presets as preceded by an *. . . . 281
10.7 Effect Info for Color 3 Way . . . 282
10.8 Remove Deinterlace-CV plugin . . . 283
10.9 Screenshot showing on the left hand side the Visibility box with Audio Effects highlighted. . . . 287
10.10 Screenshot showing the Visibility categories of plugins with all toggled on and audio highlighted. . . . 287
10.11 Down facing triangle, Right facing triangle = expander; "-" = options . . . 288
10.12 GUI of configuration for Chorus plugin . . . 290
10.13 GUI of configuration for Compressor plugin . . . 291
10.14 GUI of configuration for Compressor Multi plugin . . . 293
10.15 Graphic Equalizer audio plugin . . . 295
10.16 EQ Parametric audio plugin . . . 296
10.17 The 3 dials of Echo plugin . . . 296
10.18 GUI for EchoCancel with crosshair and mode set to ON . . . 297
10.19 GUI of configuration for Flanger plugin . . . 298
10.20 GUI of configuration for Reverb plugin . . . 301
10.21 GUI for Synthesizer . . . 302
10.22 GUI of configuration for Tremolo plugin . . . 303
10.23 Reload plugin index in yellow and Auto start lv2 gui unchecked . . . 306
10.24 Screencast of simple text interface in the middle of the screen for a Calf LV2 plugin . . . 306
10.25 Screencast with a Calf plugin glitzy window that appears when clicking the simple interface UI button. . . . 307
10.26 Blend Algebra and Blend Program GUI . . . 310
10.27 File Selection window . . . 311
10.28 Screencast showing the BlueBanana plugin control . . . 323
10.29 How it works Brightness and Contrast . . . 333
10.30 C41 - Control window and compositor window in action . . . 336
10.31 Chroma Key control window . . . 336
10.32 Windows config for Chroma Key (Avid/HSV) . . . 338
10.33 Various parameters in the 'Color' and 'Key Parameters' sections. . . . 339
10.34 Windows config for Chroma Key (HSV) . . . 343
10.35 Color 3 Way control window . . . 343
10.36 ColorSpace control window . . . 345

10.37 three active point created in CriKey . . . 347
10.38 The screenshot shows the compositor with some default settings in the controls window. . . . 348
10.39 same screenshot with moving Threshold . . . 348
10.40 Crop tool and Crop & Position plugin compared . . . 349
10.41 DeScratch control window . . . 350
10.42 Various parameters of DeScratch . . . 350
10.43 Original video with scratch; Option Mark selected and Final video . . . 351
10.44 Pulldown menu . . . 352
10.45 Control window of the DeNoise plugin . . . 353
10.46 Difference key and its problematic output . . . 353
10.47 settting *Maximun* to 0.6900 . . . 358
10.48 Setting *Maximun* to 0.6100 and *Gamma* to 0.3300 . . . 358
10.49 Control window and split screen . . . 360
10.50 Master Histogram and RGB Histogram . . . 361
10.51 Histogram Bezier / Curves . . . 365
10.52 Gain Up/Down; clamp; S-Shaped curve and Luma Key . . . 367
10.53 Holographic messages in CinGG! . . . 368
10.54 For clarity of presentation only 2 fields are shown . . . 371
10.55 Possible combinations with both Horizontal and Vertical checked . . . 373
10.56 Motion51 plugin window with its default options set. . . . 375
10.57 Motion plugin GUI . . . 378
10.58 perspective control window . . . 390
10.59 Clockwise: Nearest; Linear; OpenGL and Cubic . . . 391
10.60 Bad Misaligned color and after color aligned . . . 393
10.61 For clarity of presentation only 2 fields are shown . . . 393
10.62 Rumbler control window . . . 397
10.63 Control window and Inkscape . . . 398
10.64 Many parameters of scale ratio . . . 399
10.65 STA control window . . . 400
10.66 Sketcher control window and sketch on Compositor . . . 401
10.67 GUI for Time Average . . . 405
10.68 Temporal bands for Timefront . . . 407
10.69 GUI of the Title plugin . . . 408
10.70 Pulldown Attributes . . . 409
10.71 Final result of the exemple . . . 411
10.72 Screencast showing the Color Picker menu. . . . 413
10.73 Kerning in action . . . 416
10.74 Gui for Tracer plugin . . . 417
10.75 Rug in the top picture is traced in order to black it out or replace later . . . 418
10.76 GUI of the Videoscope. You see Histogram, RGB Parade and Vectorscope . . . 420
10.77 Pull-down Settings . . . 421
10.78 Colortest 75% with Waveform RGB (left) and Waveform (right) . . . 422
10.79 Examples of waveform (with crosshair and the coordinates' box), waveform RGB and waveform Ply . . . 423
10.80 Vectorscope (with H/S pop-up box) and VectorWheel (with 3 default grids) 424

10.81 Balancing a yellow color cast . . . 425
10.82 Before and after YUVShift adjusting . . . 427
10.83 For clarity of presentation only 3 fields are shown . . . 427
10.84 Mandelbrot Cuda plugin . . . 428
10.85 N_Body Cuda plugin . . . 429
10.86 Timeline; control window and compositor for FindObj . . . 436
10.87 FlowObj; GaborObj (before and after) and PuzzleObj . . . 438
10.88 Pulldown of Stylize mode on the original image . . . 439
10.89 Clockwise: Edge Smooth and Recursive; Detail Enhance; Stylization; Color and Pencil Sketch . . . 440
10.90 GUI for FFmpeg plugins . . . 440

11.1 Resources window displaying the Video Transitions. . . . 461
11.2 GUI for Wipe transition . . . 462
11.3 Example of the Shape Wipe -> Star . . . 465

12.1 Patchbay pulldown with Porter Duff and Graphic Art overlays expanded . 467
12.2 Normal and Arithmetic overlays . . . 468
12.3 Porter Duff overlays . . . 469
12.4 Logical overlays . . . 469
12.5 Graphic Art overlays . . . 470
12.6 Normal and Arithmetic overlays . . . 477
12.7 Logical overlays . . . 477
12.8 Graphic Art overlays . . . 478
12.9 Offset source videos to illustrate Porter Duff overlays . . . 478
12.10 Porter Duff overlays . . . 479

13.1 Actual Recording Session - note the bright green boundary box and the big arrow cursor within the "Video In" window . . . 481
13.2 Recording window with typical settings . . . 484
13.3 new feature V4L2 mpeg . . . 488
13.4 Example of the Recording settings . . . 490
13.5 Clockwise: Channels window; Video In (behind); Recording and Scan confirm . . . 491
13.6 Scanning in progress (31%) . . . 491
13.7 Channels Info window with many TV channels buttons . . . 493
13.8 Recording in real-time a TV news report . . . 494
13.9 Realtime TOC check red button . . . 495
13.10 Dual screen - initial view on the left monitor and showing part of the right monitor with 2 terminal windows . . . 497
13.11 Dual screen - CINELERRA-GG and desktop on different screens . . . 497
13.12 Dual screen - CINELERRA-GG compositor on second monitor . . . 498
13.13 Dual screen - CINELERRA-GG compositor on second monitor in full screen mode . . . 498
13.14 Ati-x10 Remote . . . 498
13.15 Interface tab shows Android Remote Control . . . 500
13.16 A Smart Phone with Android Remote Control . . . 501

13.17 Example of the Preferences menu with the Recording tab highlighted . . . 504
13.18 Example of the Recording, Video In, and Channels, along with the Add option, menus. . . . 505
13.19 Recording menu and Video In screen while capturing media. Note the Transport stop option. . . . 506
13.20 Recording setup for EasyCap device . . . 507
13.21 Two possible capture devices using USB . . . 508

14.1 When choose PAL, values get changed in window to reflect PAL . . . 510
14.2 BD Render and DVD Render . . . 513
14.3 check Deinterlace; Histogram and Resize Tracks . . . 516
14.4 Batch render for DVD creation . . . 519
14.5 Batch render for BD creation . . . 519
14.6 Video options for bluray yuv422p . . . 521
14.7 Video options for bluray yuv422p10 . . . 522
14.8 Title menu for DVD/BD . . . 535
14.9 Choose NTSC or PAL for DVD creation . . . 535
14.10 Set Scale Ratio to Cropped . . . 536
14.11 Scale Ratio plugin . . . 537
14.12 Better scale on compositor . . . 538
14.13 Error in Batch Render . . . 538

16.1 Settings of default Preferences with the Appearance tab selected . . . 563
16.2 Multi-screen Playback example useful for watching CINELERRA-GG run on the *big screen* . . . 565
16.3 Index file setup for your preferred configuration for Render Farm sharing or anything . . . 576
16.4 Shows the Cakewalk theme (courtesy Olaf) on Preferences window with list of themes . . . 577
16.5 Default warning when you click on FF icon in main window . . . 580

17.1 "Default" mode with red cursors . . . 589
17.2 "Always show next frame" mode with white cursors . . . 589
17.3 Color space and Color range . . . 590

18.1 Some windows used to manipulate Shell Commands scripts . . . 606

19.1 Render menu setup to encode using GPU with vaapi . . . 620
19.2 Use Audio→Render effect to set render parameter values and then that effect can be varied. . . . 627
19.3 Using the Compressor plugin graph to create a ringtone . . . 633

A.1 Clockwise: Viewer; Compositor; Resources and Main/Program/Timeline . 649
A.2 Load media window – note the icons top right for more options . . . 650
A.3 Format menu to change settings . . . 652
A.4 Effect blue bar with magnifier . . . 653
A.5 Menu pulldowns at the top with Transport buttons below. Note the colored tooltips too. . . . 653

A.6 From left to right: Audio 1 is disarmed – BandSlide transition in Video 1 – A highlighted section. 654
A.7 Compositor + Title menu for setting parameters + the Color Picker. 655
A.8 Color settings (Settings → Format / Settings → Preferences) 669

C.1 Exact match 704
C.2 "Giraffe" artifacts on 2 files spaced differently 704

D.1 three clips modified with OpenEDL 709
D.2 a new clip for each subject: Barcelona and Lisbon 710
D.3 Barcelona clip: 2 video tracks and 4 audio tracks 710
D.4 Barcelona clip: 2 video tracks to 1 video track 710
D.5 two new nested clips: Barcelona and Lisbon 711
D.6 Barcelona and Lisbon nested clips on timeline 711
D.7 camera 3 footage between Barcelona and Lisbon nested clips 711
D.8 completion of the work 712
D.9 Setting effects 718

Index

./configure, 14, 17
.bcast5, 706
1080 to 480, 307
1080 to 540, 308

absolute/relative path, 121
acale ratio, 398
active region, 127, 150, 212, 214, 230
Aging TV, 308
aicon
    audio icons, 88
Align cursor on frames, 39
align cursor on frames, 142
align edits, 155
align timecodes, 205
Align to samples, 39
Alpha, 308
alpha channel, 104
alpha slider, 38, 45
always show next frame, 581, 588
Android, 28
appimage, 1
    creating, 689
    management, 688
    vaapi/vdpau, 613
arm track, 130
arrow, 40, 127, 143, 150
aspect ratio, 57, 103
asset
    color, 44
    dump, 602
    info, 80
    RMB options, 79
attach effect, 38
audio, 46
    attributes, 101
    channels, 102
    device, 567
    driver, 567
    gain, 567, 571
    map, 132
    map 1:1, 132
    map 5.1:2, 132, 566, 571
    plugins, 290
    pulldown, 143
    pulldowns, 134
    record driver, 570
    time stretching, 633
audio offset, 566
audio/video sync, 200
audio/video sync via timecodes, 205
audio/video sync via waveform, 200
audioscope, 290
Authors, 642
auto rotate ffmpeg media, 581
auto scale, 308
autocolor assets, 580
autos, 37
    speed, 49
    zoom, 37

background color, 579
background rendering, 235
    output, 573
    toggle, 235
    use, 573
backup, 116, 123, 181
band slide, 464
band wipe, 464
bar color, 39, 44
batch rendering, 228
    more options, 232
bcast5, 563
blend algebra, 309
blue banana, 322
blur, 331
BluRay, 509, 520
    AVCHD, 530
    camcorders, 530
    debugging, 524
    multiple output, 532
    partial output messages run, 523
    partial output messages writing, 524
    troubleshooting, 526

without re-encoding, 530
boxblur, 332
brightness contrast, 332
build, 9, 10, 15, 17, 20, 22
booby trap, 680
configuration, 674
experimental, 673
more options, 670
prof2, 683
thirdparty, 677
use latest library, 678
valgrind, 681
burning TV, 333

C41, 333
cache, 624
cache size, 572
camera, 54, 58, 60–62
cancel operation, 38
canvas size, 102
capture, 481
commercials database, 501
digital broadcasting, 487
EasyCap model, 506
Hauppage WinTV, 508
remote control for DVB, 497
chorus, 290
chroma key, 336
chroma key Avid, 337
chroma key HSV, 342
clears before toggle, 145, 581
click to activate text focus, 581
click to deactivate text focus, 581
click to play, 51, 75
clip, 138, 141, 143
RMB options, 80
clip
older, 179
codec, 657
color
management, 663
model, 103, 591
range, 590
space, 590
color 3 way, 343
color balance, 345
color bar aicons, 88
color management, 106
color space, 345
column selection, 150
command line, 609
compositing, 55
compositing pipeline, 64
compositor
controls, 51
RMB options, 51
toolbar, 52
window, 50
compressor, 291
compressor multi band, 292
configuration, 563
environment variables, 582
context help, 22, 594, 595, 694
contour shuttle, 595
plugins, 595
Contributors, 642
control point handle, 251, 253
copy/paste, 585
copy/paste from/to resources window, 91
crash dumps, 598
crash handling tips, 625
CriKey, 346
crop, 54, 71
crop & position, 349
crossfade, 464
CUDA, 621
mandelbrot, 428
n_body, 429
plugin, 428
curve type, 62
cut and paste, 40, 127, 140
cygwin, 25

data storage, 228
dc offset, 294
decimate, 351
default A/B display, 565
default lv2 path, 575
deinterlace, 352
deinterlace-cv, 352
delay audio, 294
delay video, 352
delete clip thumbnails, 576
denoise, 294

denoise video, 352
denoisefft, 294
descratch, 350
despike, 295
difference key, 353
digital intermediate, 178
disable hardware synchronization, 566
dissolve, 464
dolby pro logic, 628
dotTV, 354
downsample, 354, 395
drag and drop, 40, 41, 127, 143
    MMB options, 145
drag handles, 183
DVD, 509, 520
    debugging, 524
    partial output messages run, 521
    partial output messages writing, 524
    subtitles, 527
    troubleshooting, 526
    yuv420p interlace mode, 527, 574

echo, 296
echo cancel, 297
edge, 354
edit, 46
    cut with snap, 147
    drag with snap, 148
    dragging group, 149
    grouping, 148, 151
    length, 155
    pulldown, 141
    selection, 145, 148–150
edit tools, 155
EDL, 101, 108, 121, 128, 179, 182
EDL dump, 602
effects/transitions info, 80
EQ graphic, 295
EQ parametric, 296
eyedropper, 54

fader slider, 131
ffmpeg, 263
    audio plugin, 441
    build index, 576
    decoder options, 265
    early probe, 263
    encoder options, 265
    image2 streams, 272
    notes, 274
    option file format, 266, 271
    options files, 264
    plugins, 440
    private options, 271
    raw input options, 273
    try first, 263
    try last, 263
    video plugins, 446
    wrench, 271
fields to frame, 354
file, 46
file by reference, 182
file format, 108, 170, 178, 213, 570, 574
file pulldown, 143
find in resources, 39
flanger, 298
flash, 464
flip, 354
folder add filter/search, 81, 85
folder contents, 78
folder modify, 81, 82
folders, 78
    RMB options, 79
force single processor, 573
foreground, 355
format, 100, 657
frame per job, 573
frame per pre-roll, 573
framerate, 102
framerate achivied, 568
frames to fields, 355
freeverb, 299
freeze frame, 356
full play, 90

gain, 299
gamma, 357
gang fader, 130
gang tracks toggle, 134
git, 9, 10, 12, 17, 19–22
goto, 36
grabshot, 93
gradient, 358
guide lines, 42

hard edge, 142
hardware
    acceleration, 611
    decoding, 612
    encoding, 619
    nvenc, 620
    special opts files, 615
    vaapi, 619
highlighted region, 127
histEQ, 359
histogram, 360
histogram Bezier, 365
holographicTV, 368
HSL Qualifier, 322
hue saturation, 368
HW Device, 573

i-beam, 40, 127, 140, 150
image sequence, 110, 111, 227, 705
in/out point, 39, 41, 127, 138, 141, 151
index file
    delete, 576
    number, 576
    path, 575
    size, 576
insertion point, 39, 41, 42, 128
insertion strategy, 114, 137, 143, 215
installation, 1
inter-view, 153
interface, 33
interlacing, 104, 630
interpolate, 299
interpolate bayer, 368
interpolate video, 368
inverse telecine, 369
invert audio, 299
invert video, 370
inverted colors, 579
iris square, 464

J-cut/L-cut, 186
job, 229

keyframe reticle, 574
keyframes, 249
    auto gang, 253
    automatic mode, 260
    autos, 249
    bump autos, 255
    compositor, 260
    default keyframe, 258
    editing operations, 261
    MMB options, 252
    plugin, 258
    pulldown, 258
    reticle, 252
    spanning, 261
keyframes reticle, 278
keypad, 50

label, 39, 42, 141
    RMB options, 80
ladspa, 303
layout scale, 578
lens, 370
linear blur, 371
linux distro, 31
live audio, 299
live video, 371
load media files, 108, 113, 137, 143
locale/language, 577
loop audio, 300
loop video, 372
LUT, 628
LV2 - Calf, 304
LV2 gui, 575
LV2 path, 305

marking selection, 150
mask, 53, 63
master track, 134
match format, 80
match output size, 39
media files filter/search, 116
media folder, 179
mezzanine codec, 178
Mirror, 372
mixers, 192
    align mixers, 198
    drag tile mixers, 197
    insert mixers, 196
    mix masters, 198
    open mixers, 193
    recover mixer window, 198

tile mixers, 197
toggle, 131, 194
viewer, 193, 196, 197
viewer RMB options, 196
mkv, 659
Motion, 377
Motion 2 Point, 388
Motion Blur, 389
Motion51, 374
mouse over, 90
multi-camera, 192, 200
multi-pane, 47
multi-session, 158
multi-viewer window, 159
multiple instances: copy/paste, 148
mute, 130, 281

nested assets, 189
nested clips, 190
nested EDL, 180, 189
nested proxy path, 575
no play, 90
nudge, 131

oil painting, 389
openCL, 621
openCV, 429
find object, 433
stylizeobj, 438
openEDL, 179, 191
output size, 56, 102
over sharpened footage, 456
overlay audio, 300
overlay pulldown, 131
overlay video, 389
overlays, 467
arithmetic, 473
graphical art, 475
logical, 474
normal, 472
Porter-Duff, 473
overwrite, 138, 142

panning box, 132
PAR, DAR, SAR, 57
patchbay, 34, 128, 134
perpetual session, 123, 580
perspective, 390
picons, 130
pitch shift, 300
pkgs, 5
play every frame, 568
play track, 129
playback -X11 direct, 623
playback buffer samples, 565
plugins, 275
attach effect, 280
attch effect, 277
change icons, 283
copy paste effects, 279
dragging, 279
dumps, 602
editing, 278
icons, 275, 577
preset edit, 259, 277, 281
reload index, 305, 575
rendered effects, 458
RMB on toolbar, 279
shared effect, 280
shared tracks, 280
short description, 282
show controls, 259, 277
speed-up via options, 289
toolbar, 276
turn on/off, 277
visibility, 286
polar, 392
popups activate on button up, 581
posterize, 392
pre-roll, 573
preferences, 563
about, 581
appearance, 576
interface, 574
performance, 572
playback_a_b, 564
recording, 570
preview mode, 154
preview region, 75
probe order, 118, 574
problem FAQ, 599
program streams selection, 120
program window, 34

project, 214, 229
project attributes, 100
project export, 122
project save, 121
projector, 54, 58, 59, 61, 62
proxy, 170, 196
    1:1, 170, 178
    error, 175

quickstart guide, 648

radial blur, 393
raw image, 112
rebuild index, 79
record audio tracks, 570
record video tracks, 570
recording, 481
    transfer VHS/DVD media, 503
    web media in real time, 482
reframeRT, 394
remove asset, 80
remove gaps, 300
render farm, 235, 237
    multi core CPU, 241
    parameters, 239
    setup, 241
    troubleshooting, 247
rendering, 212
    command line, 226, 235
    ffmpeg options, 215
    ffmpeg two-pass encoding, 219
    lossless, 218
    operation light, 51
    profile, 215
    single file, 212
    VP9 parameters, 224
    youtube preset, 223
report problem, 597
repository, 19, 20, 22
reroute, 396
resampleRT, 300
resize track, 39
resources preview, 89
resources window, 77, 137, 143, 154
reverb, 300
reverse audio, 302
reverse edits, 157
reverse video, 396
RGB, 104
rgb shift, 392
rgb-601, 392
rotate, 396
ruler, 53
rumbler, 397

safe region, 54
sample rate, 102
sample read from device, 570
samples read from recording, 571
save media files, 108, 121
scale, 398
scaling equation, 568
search resources, 78
seek, 178, 249
    issue, 589
    playing, 587
selective temporal averaging, 399
sequences, 139
set input focus, 581
settings, 46, 100, 563
shape wipe, 464
sharpen, 400
shell commands, 575, 606
shift interlace, 401
shortcuts, 540
    alternative, 560
    compositor buttons, 552
    compositor keys, 553
    compositor window, 551
    main window, 540
    main window buttons, 545
    main window keys, 548
    main window pulldowns, 540
    other windows, 557
    other windows buttons, 557
    other windows keys, 558
    resources window, 556
    viewer keys, 555
    viewer window, 554
    viewer window buttons, 554
show edit, 39
show ovelays window, 95, 249
shuffle edits, 158
shuttle pro v2 and shuttlexpress, 159

single-user build, 12, 15, 21
sketcher, 401
slide, 466
SMP cpus, 573
snap, 147
snapshot, 93
    path, 574
sort media files, 117
sound level, 302
sound level meters window, 97
source target, 90
spectrogram, 302
speed percent, 404
sphere cam, 404
splice, 138
split, 142
still image, 109
stretch, 394
subtitles, 209
SVG via Inkscape, 397
swap channels, 404
swap frames, 404
sync, 145
synthesizer, 302

tars, 4
temporary, 55
theme, 576
    create new theme, 692
thresold, 405
time average, 405
time format, 36
time stretch RT, 302
timebar, 39, 41, 42, 75, 145, 579
timeblur, 406
timecode, 205
timefront, 406
timelapse helper, 407
timeline, 34, 127, 139, 144, 153, 179, 193
    optimization, 142
    selection, 37
timeline rectify audio, 580
title, 407
tool info, 54, 61, 257
tracer, 416
track, 46
    ganged mode, 134
    height, 130
    popup menu, 38
    pulldown, 133
    size, 56
transcode, 178
transitions, 461
    audio, 463
    video, 464
translate, 418
transport buttons, 35, 48, 51, 73, 94
tremolo, 303
trim, 183, 278, 574
    edge, 185
    ripple, 184
    roll, 185
    slide, 185
    slip, 185
    split view, 186
two-screen, 127, 137

undo dump, 602
unicode insertion, 593
unsharp, 419
use thumbnails in resources window, 581
user clip bin, 81
user folders/bins, 81
user media bin, 81
user title, 39

valgrind
    heaptrack, 682
vicon
    color mode, 578
    quality, 578
    video icons, 88
video, 46
    attributes, 102
    driver, 569
    plugins, 307
    pulldown, 134, 143
    record driver, 571
videoscope, 51, 73, 419
    histogram, 426
    vectorscope, 424
    waveforms, 422
view, 46

view follows playback, 566
view thumbnais size, 578
viewer
    RMB options, 73
    window, 73, 137
viewport, 60

warnings
    bluray no root, 580
    file by reference, 580
    EDL version, 580
    ffmpeg probe, 579
wave, 426
waveform, 37, 74, 88, 130
whirl, 426
window, 46
    layout, 46
windows 10, 24
wipe, 466
workarounds, 721
wrench, 213, 570, 574

Ydiff, 704
yuv, 104, 426
yuv color range, 579
yuv color space, 579
yuv shift, 426
yuv411, 426

zoom, 95, 466
    menu, 51
    panel, 36, 251, 252
    slider, 52
zoom blur, 427