5DtoRGB Batch 1.5

June 26, 2013

http://rarevision.com/5dtorgb/
support@rarevision.com

What's New

5DtoRGB 1.5.15 is a maintenance release. Some bugs have been fixed, and as a result general stability should be improved.

5DtoRGB Batch Basic Usage

5DtoRGB transcodes files in the AVC/H.264 video format to either uncompressed DPX image sequences or Apple ProRes MOV files.

To convert a single file, do this:

1. Click the "Convert..." button in the main window or select "Open..." from the "File" menu.
2. In the file browser window, select the file you would like to convert and click "OK."
3. Specify an output format, decoding matrix and program options, then click "Convert…"
4. Use the file browser to select a save location for the image sequence or MOV file.
5. Once the progress bar reaches 100%, the conversion is complete!

You can add several files to the batch queue by dragging and dropping them directly onto 5DtoRGB's window. You can also use the "Add to Batch..." button to add multiple files.

NOTE: Clicking "Cancel" will terminate the conversion process. If a DPX conversion is in progress when "Cancel" is clicked, 5DtoRGB will finish writing the current frame to disk before stopping. If a QuickTime conversion is in progress, 5DtoRGB will remove the partially converted MOV file.

Program Options

Output Format: Choose the output format to which you'd like to transcode. Selecting one of the ProRes options will output an MOV file with embedded timecode. Choosing "DPX" will output a DPX image sequence with timecode inserted into the header of each DPX file. ProRes output is typically used for optimal editing performance with software such as Apple Final Cut Pro or Final Cut Pro X and Adobe Premiere. ProRes files generated by 5DtoRGB will play back in Final Cut Pro 7 without rendering (no red bar).

DPX output is often used for visual effects purposes or when the absolute highest quality is needed.

Frame Rate Selection: Use this to set your footage play at an arbitrary frame rate. This is useful for "overcranking" effects, like when shooting at 60 fps for 24 fps output. You can also use this for "undercranking" effects or any other special purpose. Keep in mind that this option does not "convert" footage from one frame rate to another. It wouldn't be useful for converting 29.97 fps material to 23.976, for example. It simply sets the playback rate of the video without changing the frame count.

Decoding Matrix: Choose the decoding/transform matrix you want to use here. Normally, you would match the matrix that was used to encode the video to YUV (technically YCbCr). Generally speaking, BT.709 is for HD material and BT.601 is for SD. Canon HDSLRs prior to the 5D Mk.III use the BT.601 matrix, which is selected by default. You can disable decoding altogether by choosing "Raw YCbCr." This copies luminance to the green channel of the output file, and Cb and Cr to blue and red, respectively. This option can be used to access the raw luminance data out of the input files, as the luminance data is unprocessed bit-exact output from the H.264 decoder. You can also specify a custom decoding matrix by choosing "Import Matrix…" from the "File" menu. 5DtoRGB accepts matrix files in Rarevision's dMatrix format (.dmtx files).

Luminance Range: This setting tells 5DtoRGB whether to interpret the color range in your footage as full range (0-255) or broadcast range (16-235). The correct setting will depend on the camera used. Some cameras, like Canon HDSLRs, shoot using full-range color. On the other hand, the Panasonic DMC-GH2 shoots using broadcast range color. If you select "ITU-R BT.709" as your decoding matrix, "Broadcast Range" will be selected automatically.

Chroma Mode: Selects the chroma interpolation method. The options are:

> *Default:* Uses the highest quality chroma interpolation available. The best choice for progressive footage like from Canon HDSLRs.
>
> *Interlaced:* Uses an alternate method better suited to interlaced material. Don't select this when working with Canon HDSLR footage.
>
> *Disabled:* Disables chroma interpolation. With this option, Cb and Cr channels are made full raster using "nearest neighbor" scaling . Only useful for testing and comparison purposes.

Post-Processing: You can use this setting to post-process your footage using the built-in Technicolor CineStyle LUT or a separate GLSL fragment shader. Shaders can be imported by selecting "Import Fragment Shader..." in the "File" menu. You can also have a shader loaded automatically by naming it "default.glsl" and placing it in the same directory as the files to be transcoded. If 5DtoRGB finds a default.glsl shader in the same directory as the file being transcoded, it will automatically load it. More information on this feature can be found in the "Using GLSL Fragment Shaders" section.

Gamma Correction: Specify a decimal value here to adjust gamma during transcoding. This works much like the "Gamma Correction" video filter in Final Cut Pro. For example, to shift gamma from 2.2 to 1.8, use the value **1.22.** The default value is 1.0, which can be considered a "pass through" value that performs no correction.

Override Clip's Timecode: Select this if you want to ignore the timecode either embedded into the file or in an accompanying THM file and specify your own timecode. The timecode value entered into the timecode entry field will be used.

Preferences

The preferences dialog contains options for modifying the behavior of 5DtoRGB. The following preferences can be set:

Don't warn about missing THM files - Disables the warning that is displayed when an input file's accompanying THM file in not present.

Don't automatically load dmtx files – Selecting this will prevent 5DtoRGB from loading dMatrix files automatically when present in the same directory as the input file.
Set default timecode hour to 0 – If a THM file containing timecode is not found when importing a clip, 5DtoRGB uses a default timecode value of 01:00:00:00. Use this option to change it to 00:00:00:00 instead.

Usage Tips

5DtoRGB will perform at its best when reading from and writing to fast storage media. The faster the storage device, the faster 5DtoRGB can perform read/write operations. A SAN or locally attached disk array is preferable.

You can run multiple instances of 5DtoRGB on Mac OS X. To do so, place 5DtoRGB Batch.app in your /Applications folder, and type this command in a terminal:

```
open –n "/Applications/5DtoRGB Batch.app"
```

Each time you enter this command, a new instance of 5DtoRGB will be launched. This makes it possible to do multiple conversions simultaneously. It is also possible to do conversions using a shell script that utilizes 5DtoRGB's command line interface. Third-party scripts are available for download that support batch conversions.

Timecode Support

5DtoRGB will derive start timecode values from the file's embedded timecode or THM file in the same way that Canon's E1 plugin for Final Cut Pro does when using the "Log and Transfer" function. However, unlike the Canon plugin, 5DtoRGB does not require the CF card directory structure to be intact. It only needs the THM file to be in the same directory as the MOV file. In the event that the THM file is unavailable, you can still specify the start timecode yourself by entering an arbitrary value prior to starting the conversion.

Command Line / Shell Scripting Support

5DtoRGB supports basic command line operation when running in CLI (command line interface) mode. Using a terminal, you can pass 5DtoRGB five arguments:

1. An input file path
2. An integer representing the desired output format
3. An integer representing the decoding matrix
4. An integer representing the luminance range
5. A floating point value representing the frame rate (e.g. 23.976)

Here's an example:

```
"/Applications/5DtoRGB Batch.app/Contents/MacOS/5DtoRGB Batch" /Clips/MVI_0001.MOV 2 1 0 23.976
```

The first argument, "/Clips/MVI_0001.MOV," is the path to the input file. There is no option for an output file name at this time. The output file will be written to the input file's directory with the same name as the input file and a "_1.mov" suffix.

The second argument, "2," corresponds to the index of the desired format in the list of output formats. So, if you want to output the file as ProRes 422 (HQ), this number would be 2, since it is the second ProRes option in the list. DPX output is not supported in command line mode, so numbering starts at 1 (Which would be ProRes 4444). The supported output formats are as follows:

1. ProRes 4444
2. ProRes 422 (HQ)
3. ProRes 422
4. ProRes 422 (LT)
5. ProRes 422 (Proxy)

The next two arguments correspond to the list index of the desired decoding matrix and luminance range value, respectively. Unlike the output format list, these indexes are zero-based so "0" corresponds to the first item in the list.

The last argument is the desired frame rate to use for the transcoded file. This is used to retime shots that have been shot at higher frame rates with "overcranking," or slow motion in mind.

Note that when in CLI mode, dmtx files are automatically loaded when present. This overrides the value specified for the decoding matrix on the command line. If you would rather specify a matrix even when dmtx files are present, disable automatic loading of dmtx files in the preferences.

dMatrix Custom Matrix Support

If you've designed a custom decoding matrix for use with 5DtoRGB, dmtx files will be automatically loaded if in the same directory as the footage and with the same file name (e.g. MVI_0123.MOV and MVI_0123.dmtx). The matrix will show in the "Decoding Matrix" list as "dMatrix Custom." If you want 5DtoRGB to ignore this custom matrix, choose another from the decoding matrix list.
dMatrix files can store a gamma correction value as well as matrix values. If you want this custom gamma value applied to the footage, select the "Bake Gamma Correction" option in 5DtoRGB (this will override the default gamma correction value). You may still need to select a gamma flagging option if some applications are not displaying your custom gamma properly.

Using GLSL Fragment Shaders

The ability to import GLSL shaders is a powerful feature that allows users to run their own fragment programs on the video card's GPU. Effects like color correction, noise and sharpening can be implemented with these programs.

User-defined GLSL shaders execute within main() after gl_FragColor. If you want to implement your own shader, you can use `gl_FragColor` as your starting point. For example, this simple program will perform a gamma correction:

```
gl_FragColor.rgb = pow(gl_FragColor.rgb,vec3(1.0/0.818));
```

The gamma correction value in this case is 0.818. Varying this value will vary the amount of gamma correction applied.

5DtoRGB does not currently perform any GLSL syntax validation. Because of this, GLSL code containing errors may cause the fragment shader to fail. This may result in grayscale-only or corrupted video output. You will need to ensure your GLSL code is valid prior to importing it.
For more information on GLSL and OpenGL, check out http://www.opengl.org.

Requirements

5DtoRGB Batch 1.5.15 requires Mac OS 10.7 or later. An OpenGL-enabled video card or motherboard video chipset is required, and the the OpenGL implementation on the system must support the OpenGL extensions used by 5DtoRGB. 5DtoRGB is compiled as a 64 bit Intel-only binary, and will only run on 64 bit Intel processors.

Known Compatibility Issues

The ATI Radeon X1900 graphics card is not supported by 5DtoRGB. Attempting to use 5DtoRGB with this card will most likely result in poor performance and/or severe video artifacts.

Notes on LGPL Compliance

This software links with software libraries from the FFmpeg project under the terms of the GNU LGPL version 2.1. Source code for these libraries can be found at http://www.ffmpeg.org. Full text of the GNU LGPL 2.1 can be found at http://www.gnu.org/licenses/lgpl-2.1.txt.