BBC | Research & Development

Use of Look-Up Tables (LUTs) in FFmpeg

Part of the HDR-TV series. Last updated November 2016.

Introduction

As discussed in "Perceptual Quantiser (PQ) to Hybrid Log-Gamma (HLG) Transcoding and Conversion," available on the BBC R&D HDR web page, an issue exists when using Look-Up Tables (LUTs) with devices that are not aware of the input or output colour spaces in use, e.g. 3G-SDI hardware LUT devices. Most LUTs work in the RGB domain, so a conversion is required between 3G-SDI's YCbCr domain and RGB.

In hardware, this is often fixed to use the ITU-R BT.709 colour gamut. If a different colour gamut is required, the LUT can be designed with a pre-correction to fix this behaviour. This should not be required in software solutions. When using software it should be possible to correctly assign the correct matrices to convert between YCbCr and RGB and vice versa.

A method to correctly assign conversion matrices when using a command line video conversion software package is presented here. Software that is colour-aware, for example non-linear video editing or colour grading software, should have a method of correctly assigning the conversion matrices in use and flag this use in the output video file.

Example Software LUT Application

FFmpeg (and its derivatives) is a tool used for the creation, transcoding and application of filters to media files and which can be configured to correctly use colour conversion matrices and colour metadata. This method should work in recent software releases[1]. In this example BASH script, we use FFmpeg to apply a standard dynamic range to Hybrid Log-Gamma conversion with a change in colour gamut from ITU-R BT.709 to ITU-R BT.2020 Non-Constant Luminance. This is a minimal working example and further command-line flags would be required to set parameters such as the output bitrate and encoding settings.

[1] Tested with git repository cloned 16th November 2016

```
#!/bin/bash

LUTFILE=sdr709_to_hlg2020.cube
FILETYPE=MXF
VIDEO_IN=$1
VIDEO_OUT=${VIDEO_IN%.*}-${LUTFILE%.cube}.$FILETYPE
CODEC=$2

echo $VIDEO_IN

echo $VIDEO_OUT

ffmpeg -y -i $VIDEO_IN -vf lut3d=$LUTFILE,\
scale=in_range=tv:out_range=tv:in_color_matrix=bt709:out_color_matrix=bt2020_ncl\
-codec:v $CODEC -codec:a copy  -color_primaries bt2020 -colorspace bt2020_ncl\
-color_trc arib-std-b67 -pix_fmt yuv422p10le $VIDEO_OUT
```

In order to ensure that the correct input and output matrices are used, we apply a "scale" video filter:

```
scale=in_range=tv:out_range=tv:in_color_matrix=bt709:out_color_matrix=bt2020_ncl
```

This filter sets the data range to narrow (TV), i.e. for 10 bit video the range will be 64-940 and sets the correct input and output colour matrices.

We also set the output colour space, colour primaries and transfer function metadata to be included either in the video data stream or the container:

```
-color_primaries bt2020 -colorspace bt2020_ncl -color_trc arib-std-b67
```

It should be noted that, at the time of writing, not all codecs or container formats have been updated to correctly store this information.

These Look-Up Tables use the Cube LUT specification. Further information can be found in Cube LUT Specification, Version 1.0 by Adobe.