

NAME

openycocg – Command–line interface to encode, decode, play and analyse Y’CoCg video

SYNOPSIS

openycocg (**-e** | **-d** | **-p** | **-a**) **-i** *input_file* [**-o** *output_file*]

openycocg **-h** | **-v**

DESCRIPTION

YCoCg is a video codec, built on the Y’CoCg colour space and specifically designed for restoration purposes.

libycocg is a C library implementing the **YCoCg** video codec.

openycocg is a Bash command–line interface to **libycocg** allowing to encode, decode, play and analyse Y’CoCg video files.

The **openYCoCg** package includes the **libycocg** library and its **openycocg** command–line interface.

OPTIONS

-h, --help

display a help message

-v, --version

display the running version

GENERAL OPTIONS

Select the mode:

-e, --encode

encoding mode: encode an *input_file* to an *output_file*

-d, --decode

decoding mode: decode an *input_file* to an *output_file*

-p, --play

playing mode: play an *input_file*

This mode is beneficial when the **libycocg** library is used as a standalone application and not as an embedded library into another application, like a restoration suite. The author is indebted to **mpv** for the inspiration given.

-a, --analyse, --analyze

analysing mode: analyse the validity of a YCoCg encoded *input_file* and generate a report to an *output_file* if specified or to the Terminal otherwise

Select the file(s):

-i *input_file*, **--input=***input_file*

In encoding mode, all container formats supported by FFmpeg should work.

In decoding or analysing mode, the container formats NUT (.nut), MP4 (.mp4), QuickTime (.mov), AVI (.avi) and Matroska (.mkv) have been tested as wrappers for the YCoCg video codec.

-o *output_file*, **--output=***output_file*

In encoding mode, the container formats NUT (.nut), MP4 (.mp4), QuickTime (.mov), AVI (.avi) and Matroska (.mkv) have been tested as wrappers for the YCoCg video codec.

In decoding mode, all container formats supported by FFmpeg should work.

In analysing mode, the output file format can be plain text (.txt), JSON (.json) or XML (.xml).

ENCODING OPTIONS

The following list is not exhaustive.

--bit-depth={10|12|16|24}

bit-depth can be 10, 12, 16 (default) or 24 per channel

We suggest to digitise at 16-bit per channel and to use this bit-depth for actual restoration work. However, 10-bit and 12-bit are provided as well.

Currently, the 24-bit per channel flavour is primarily meant for research purposes on file formats for the future, because it can hardly be transcoded in current formats.

--compression={no|yes}

compression can be *no* (default) or *yes*

A lossless compression can be applied. However, this makes sense only for conservation purposes, because during the restoration process "raw" is always faster and any compression would slow down significantly the image processing.

The wavelet compression used is possibly a wee bit better than HuffYUV, FFV1 or JPEG2000 in terms of both speed and compression rate.

--endian={big|little}

endianness can be *big* or *little* (default)

OTHER OPTIONS

The following list is not exhaustive.

--lut=*path*

path to an 1D or a 3D LUT to apply (default is no LUT)

A LUT can be applied in each mode to the input file.

For 1D LUT, which transforms e.g. from floating-point scene linear into camera log or a display-referred space, the maximum allowed size is currently 65536, i.e. 16-bit precision.

--report-fmt={json|plain|xml}

report format can be *json*, *plain* text (default) or *xml*

NOTES

The current **openycocg** command-line interface is a work in progress.

COPYRIGHT

Copyright (c) 2014–2018 by Reto Kromer

LICENSE

The **openYCoCg** package is released under a Creative Commons Attribution 4.0 International License.

DISCLAIMER

The **openYCoCg** package is provided "as is" without warranty or support of any kind.