Working Beyond RGB
A Report from the Field

Reto Kromer • AV Preservation by reto.ch

4th International Conference Colour in Film
British Film Institute, London, United Kingdom
25–27 February 2019

Summary

Y′CoCg Colour Space

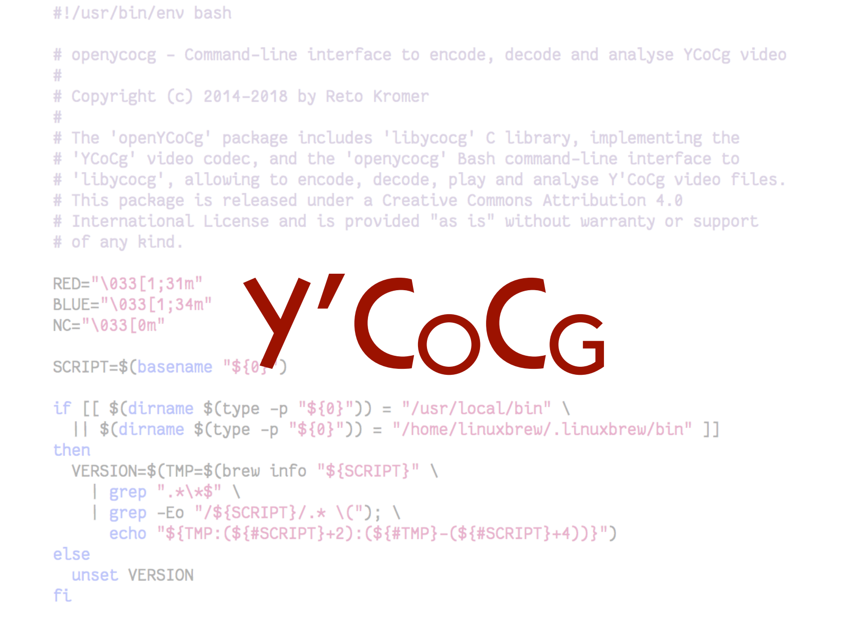
- openYCoCg (= libycocg + openycocg)

Multispectral Moving Images

- openMSMI (= libmsmi + openmsmi)

Moving Image

- openMovIm (= libmovim + openmovim)



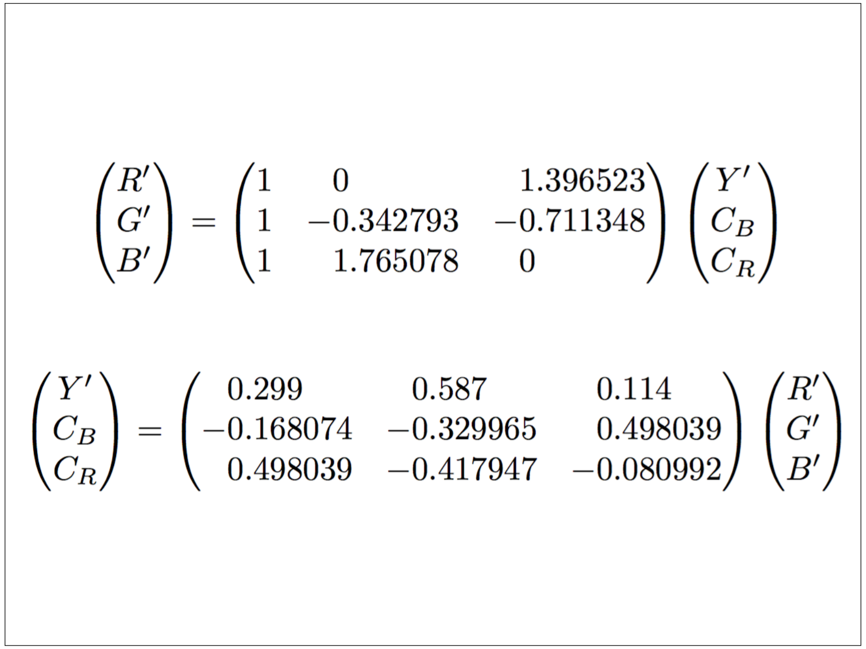
```
#!/usr/bin/env bash

# openycocg - Command-line interface to encode, decode and analyse YCoCg video
#
# Copyright (c) 2014-2018 by Reto Kromer
#
# The 'openYCoCg' package includes 'libycocg' C library, implementing the
# 'YCoCg' video codec, and the 'openycocg' Bash command-line interface to
# 'libycocg', allowing to encode, decode, play and analyse Y'CoCg video files.
# This package is released under a Creative Commons Attribution 4.0
# International License and is provided "as is" without warranty or support
# of any kind.

RED="\033[1;31m"
BLUE="\033[1;34m"
NC="\033[0m"

SCRIPT=$(basename "${0}")

if [[ $(dirname $(type -p "${0}")) = "/usr/local/bin" \
  || $(dirname $(type -p "${0}")) = "/home/linuxbrew/.linuxbrew/bin" ]]
then
  VERSION=$(TMP=$(brew info "${SCRIPT}" \
    | grep ".*\*$" \
    | grep -Eo "/${SCRIPT}/.* \("); \
    echo "${TMP:(${#SCRIPT}+2):(${#TMP}-(${#SCRIPT}+4))}")
else
  unset VERSION
fi
```

Y′CoCg

Y′CBCR

Y′CoCg

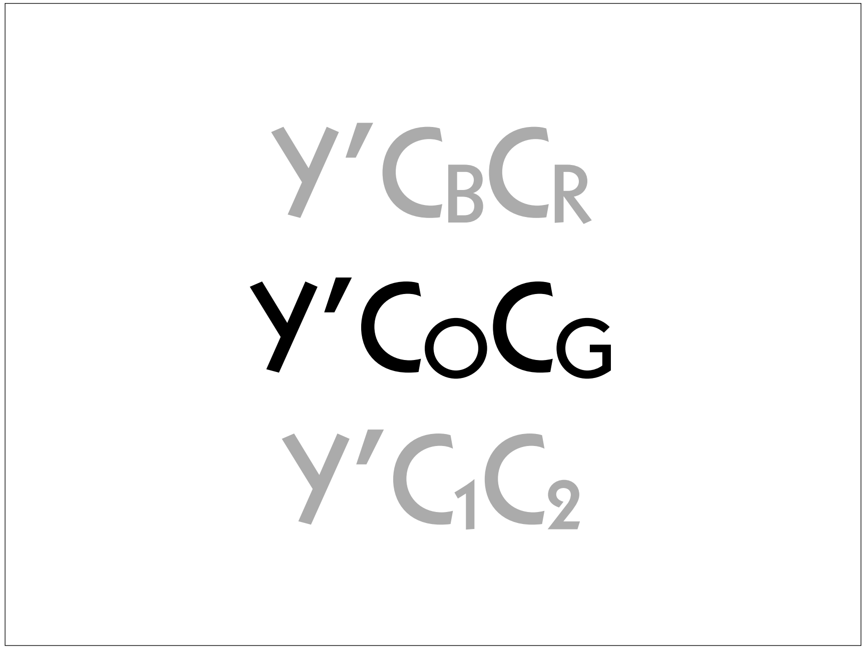
$Y'C_1C_2$



$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1.396523 \\ 1 & -0.342793 & -0.711348 \\ 1 & 1.765078 & 0 \end{pmatrix} \begin{pmatrix} Y' \\ C_B \\ C_R \end{pmatrix}$$

$$\begin{pmatrix} Y' \\ C_B \\ C_R \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.168074 & -0.329965 & 0.498039 \\ 0.498039 & -0.417947 & -0.080992 \end{pmatrix} \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix}$$

5

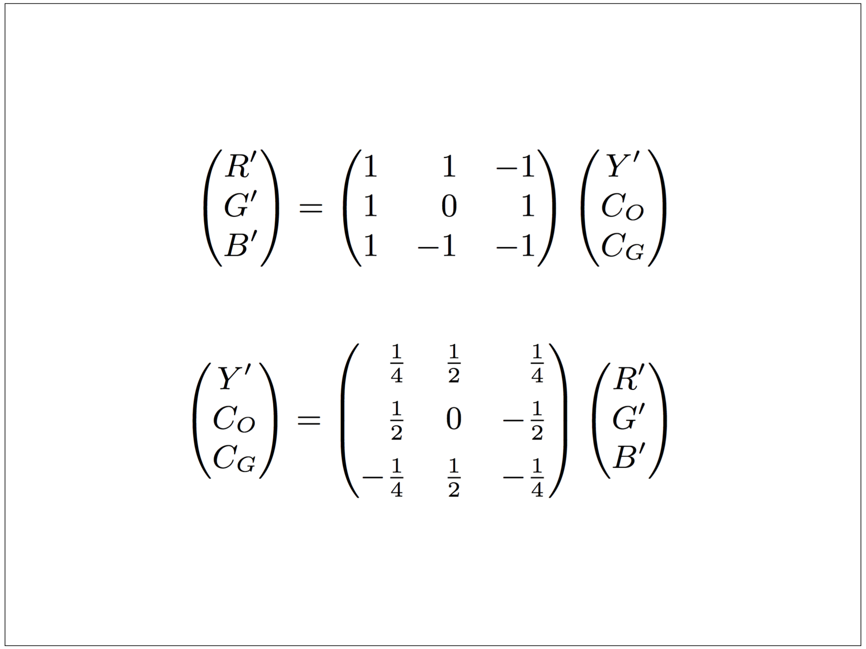


$Y'C_BC_R$

$Y'C_OC_G$

$Y'C_1C_2$

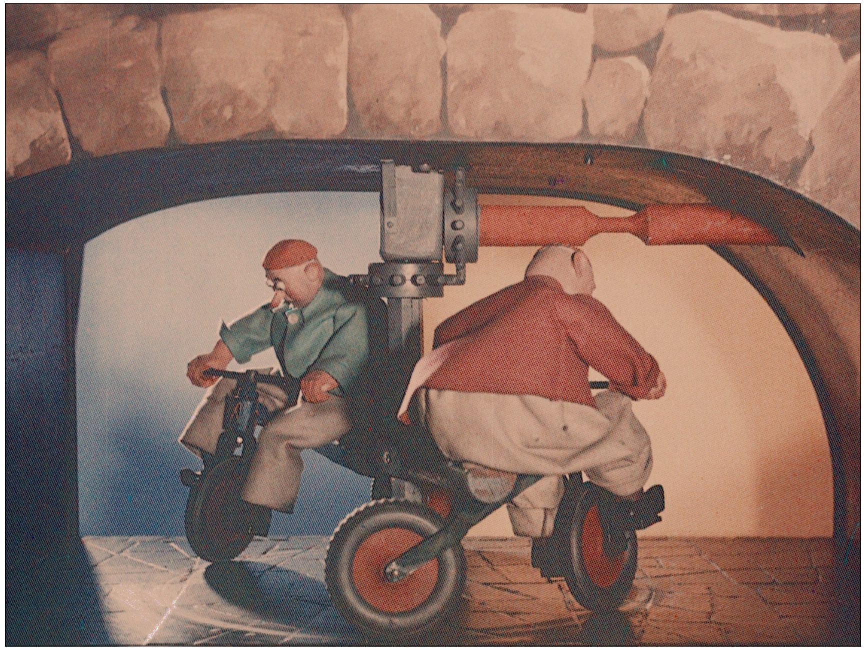
6



$$\begin{pmatrix} R' \\ G' \\ B' \end{pmatrix} = \begin{pmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{pmatrix} \begin{pmatrix} Y' \\ C_O \\ C_G \end{pmatrix}$$

$$\begin{pmatrix} Y' \\ C_O \\ C_G \end{pmatrix} = \begin{pmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & -\frac{1}{2} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \end{pmatrix} \begin{pmatrix} R' \\ G' \\ B' \end{pmatrix}$$

7



8

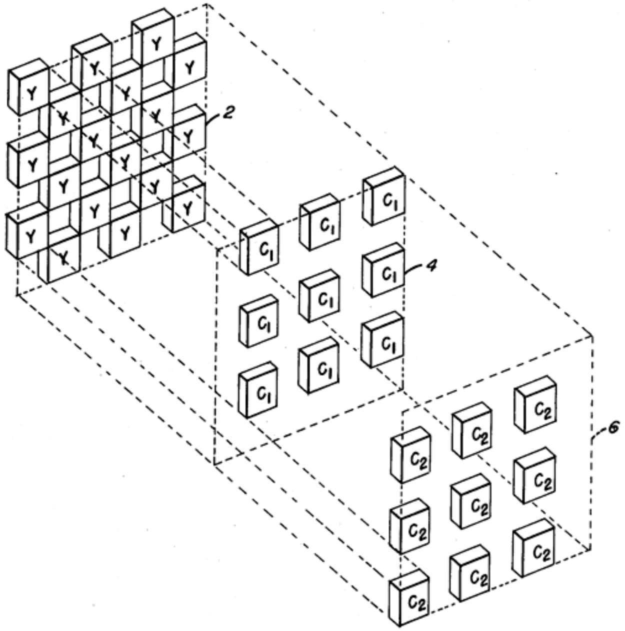
Draft Programme

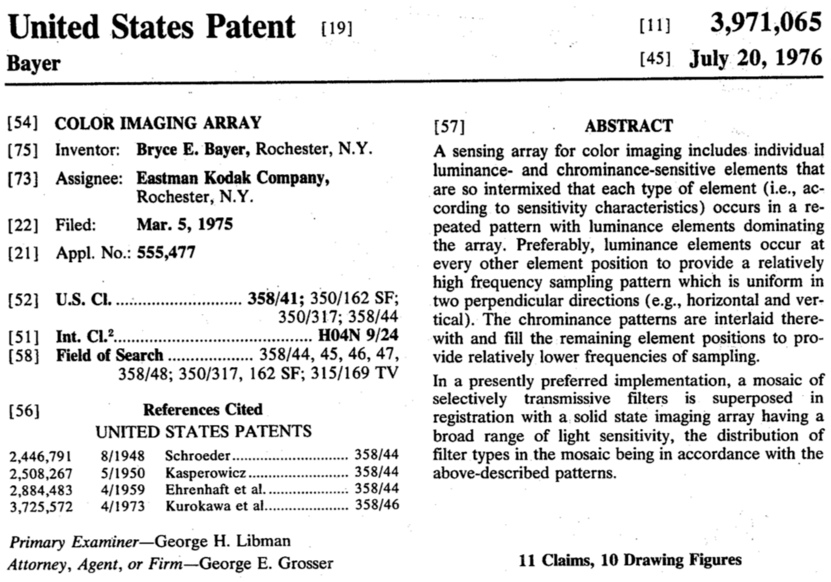
Monday 15	• Just Chemistry • Hands-on #1: Manufacturing of Orthochromatic Gelatine Emulsion • Hands-on #2: Manufacturing of Panchromatic Gelatine Emulsion • Opening Night Reception
Tuesday 16	• Hands-on #3: Manufacturing of Cinécolor • Hands-on #4: Manufacturing of Kodacolor Lenticular • Hands-on #5: Manufacturing of Dufaycolor • Shooting onto the Manufactured Films • Processing of the Exposed Films
Wednesday 17	• ACES for Restorers • Programming Look-up Tables • OpenEXR for Conservators and Restorers • Visit of a Postproduction Company
Thursday 18	• Digitising of Historic Films and New Produced Films • DCP Encoding of Legacy Films • Screening Night on Additive Colour Systems
Friday 19	• Ethical Issues on Digital Conservation and Restoration • Open Forum • Boat Trip on the Lake of Geneva and Closing Night Reception

$Y'C_BC_R$

$Y'C_OC_G$

$Y'C_1C_2$





United States Patent [19]
Bayer

[11] **3,971,065**
[45] **July 20, 1976**

[54] **COLOR IMAGING ARRAY**

[75] Inventor: **Bryce E. Bayer,** Rochester, N.Y.

[73] Assignee: **Eastman Kodak Company,** Rochester, N.Y.

[22] Filed: **Mar. 5, 1975**

[21] Appl. No.: **555,477**

[52] **U.S. Cl.** .... **358/41;** 350/162 SF; 350/317; 358/44

[51] **Int. Cl.**^{2}.. **H04N 9/24**

[58] **Field of Search** .... 358/44, 45, 46, 47, 358/48; 350/317, 162 SF; 315/169 TV

[56] **References Cited**

UNITED STATES PATENTS

2,446,791	8/1948	Schroeder	358/44
2,508,267	5/1950	Kasperowicz	358/44
2,884,483	4/1959	Ehrenhaft et al.	358/44
3,725,572	4/1973	Kurokawa et al.	358/46

Primary Examiner—George H. Libman
Attorney, Agent, or Firm—George E. Grosser

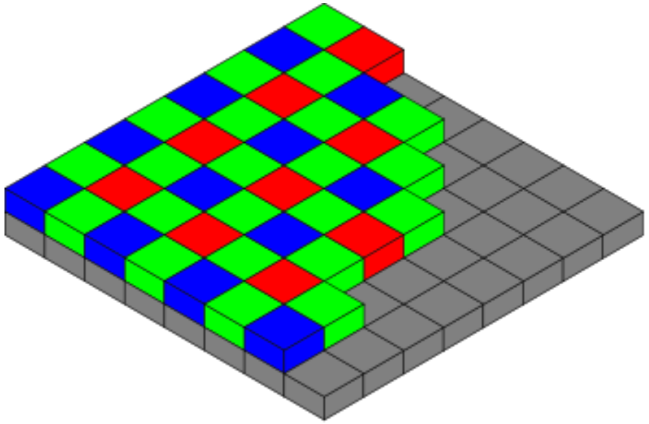
[57] **ABSTRACT**

A sensing array for color imaging includes individual luminance- and chrominance-sensitive elements that are so intermixed that each type of element (i.e., according to sensitivity characteristics) occurs in a repeated pattern with luminance elements dominating the array. Preferably, luminance elements occur at every other element position to provide a relatively high frequency sampling pattern which is uniform in two perpendicular directions (e.g., horizontal and vertical). The chrominance patterns are interlaid therewith and fill the remaining element positions to provide relatively lower frequencies of sampling.

In a presently preferred implementation, a mosaic of selectively transmissive filters is superposed in registration with a solid state imaging array having a broad range of light sensitivity, the distribution of filter types in the mosaic being in accordance with the above-described patterns.

11 Claims, 10 Drawing Figures

Bayer



13

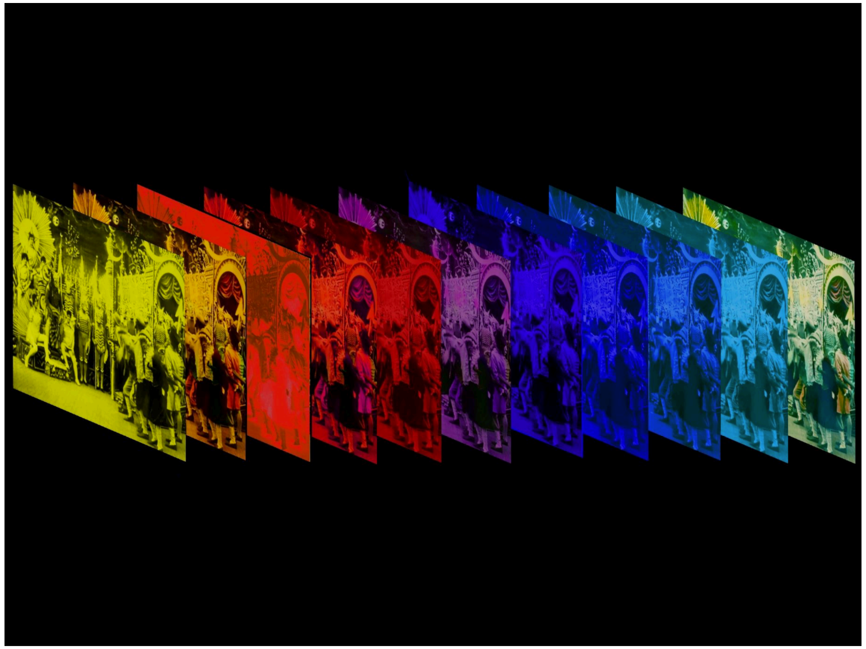
```
float bayer_matrix[16] {
    0, 128,  32, 160,
  192,  64, 224,  96,
   48, 176,  16, 144,
  240, 112, 208,  80
}
```

14

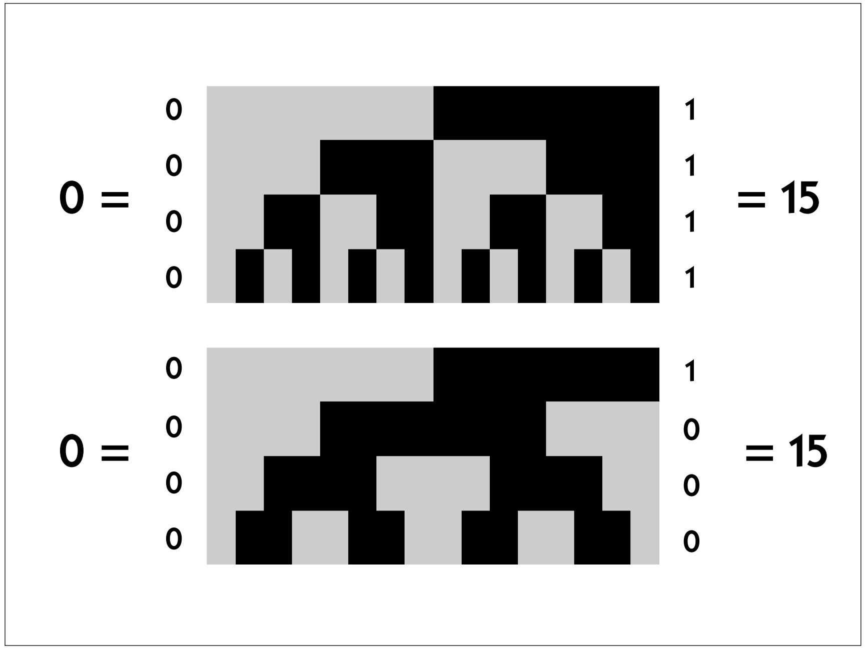
Multispectral

```
#!/usr/bin/env bash

# openmsmi - Command-line interface to encode, decode and analyse
#            multispectral moving images
#
# Copyright (c) 2017-2018 by Reto Kromer
#
# The 'openMSMI' package includes 'libmsmi' C library, implementing the 'MSMI'
# video codec, and the 'openmsmi' Bash command-line interface to 'libmsmi',
# allowing to encode, decode, play and analyse multispectral moving images.
# This package is released under a Creative Commons Attribution 4.0
# International License and is provided "as is" without warranty or support
# of any kind.

RED="\033[1  m"
BLUE="\033[ ;3  "
NC="\033[0m

SCRIPT=$(basename "${0}")

if [[ $(dirname $(type -p "${0}")) = "/usr/local/bin" \
  || $(dirname $(type -p "${0}")) = "/home/linuxbrew/.linuxbrew/bin" ]]
then
  VERSION=$(TMP=$(brew info "${SCRIPT}" \
    | grep ".*\*$" \
    | grep -Eo "/${SCRIPT}/.* \("); \
      echo "${TMP:(${#SCRIPT}+2):(${#TMP}-(${#SCRIPT}+4))}")
else
  unset VERSION
fi
```

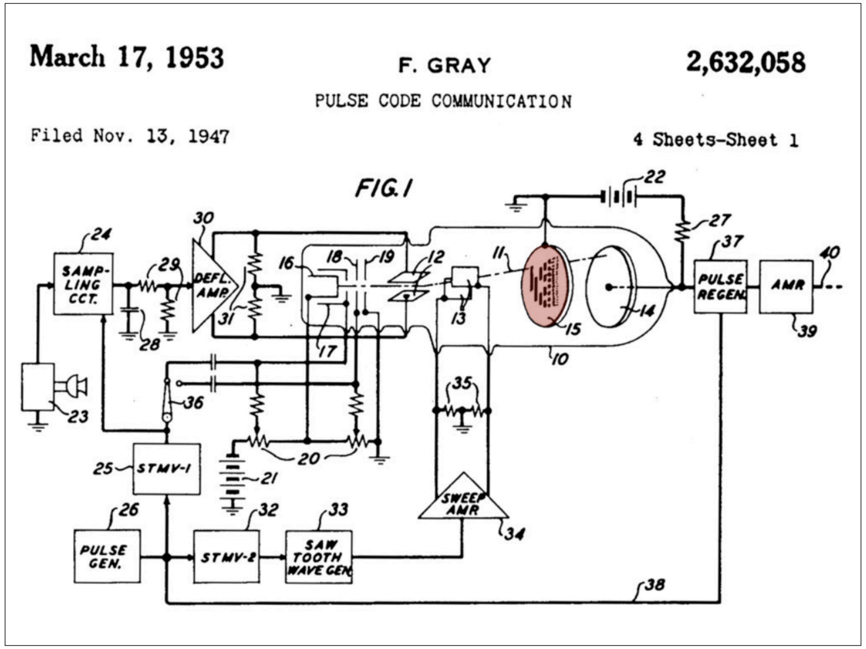
15



16



17

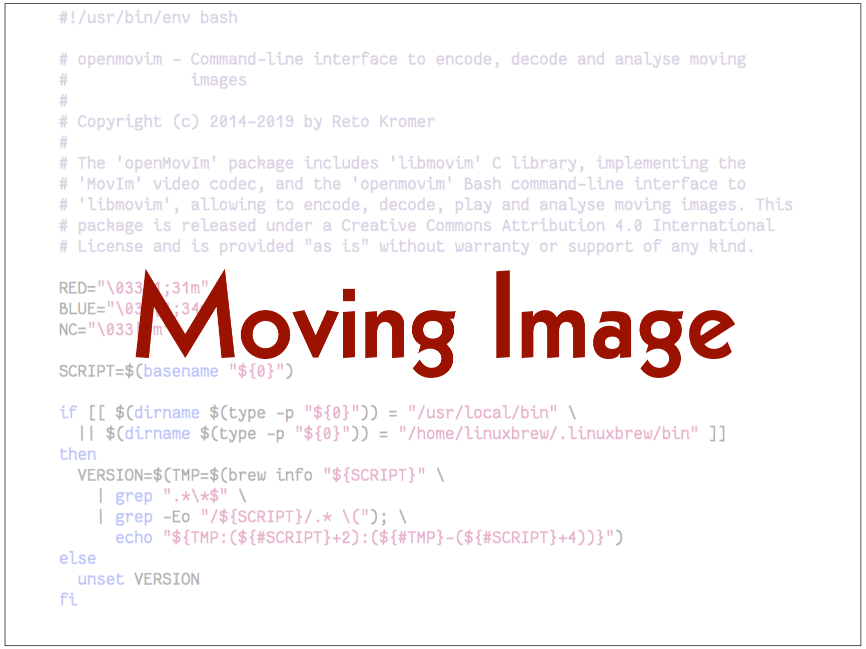


March 17, 1953 F. GRAY 2,632,058

PULSE CODE COMMUNICATION

Filed Nov. 13, 1947 4 Sheets-Sheet 1

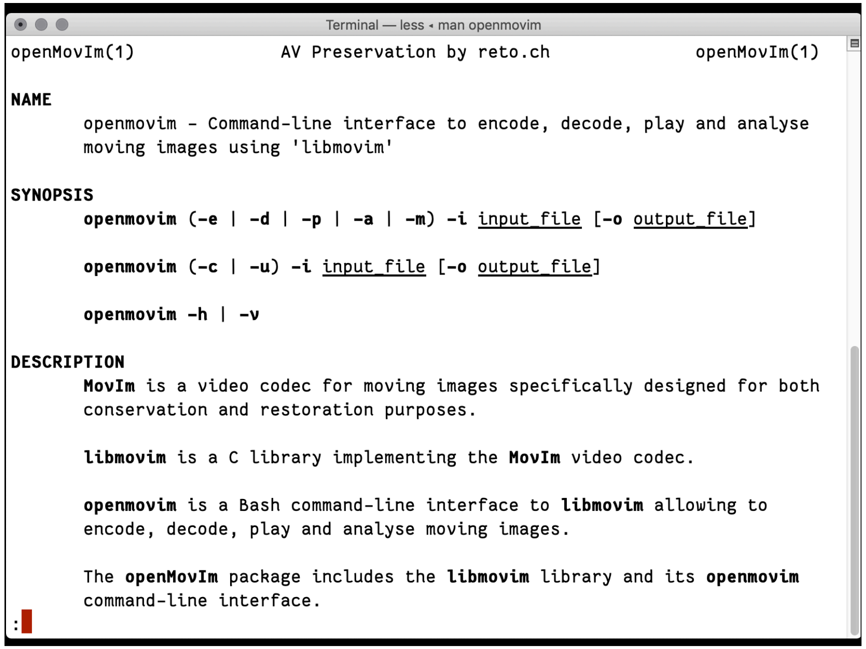
18



```
#!/usr/bin/env bash

# openmovim - Command-line interface to encode, decode and analyse moving
#             images
#
# Copyright (c) 2014-2019 by Reto Kromer
#
# The 'openMovIm' package includes 'libmovim' C library, implementing the
# 'MovIm' video codec, and the 'openmovim' Bash command-line interface to
# 'libmovim', allowing to encode, decode, play and analyse moving images. This
# package is released under a Creative Commons Attribution 4.0 International
# License and is provided "as is" without warranty or support of any kind.

RED="\033[1;31m"
BLUE="\033[1;34m"
NC="\033[0m"

SCRIPT=$(basename "${0}")

if [[ $(dirname $(type -p "${0}")) = "/usr/local/bin" \
  || $(dirname $(type -p "${0}")) = "/home/linuxbrew/.linuxbrew/bin" ]]
then
  VERSION=$(TMP=$(brew info "${SCRIPT}" \
    | grep ".*\*$" \
    | grep -Eo "/${SCRIPT}/.* \("); \
    echo "${TMP:(${#SCRIPT}+2):(${#TMP}-(${#SCRIPT}+4))}")
else
  unset VERSION
fi
```

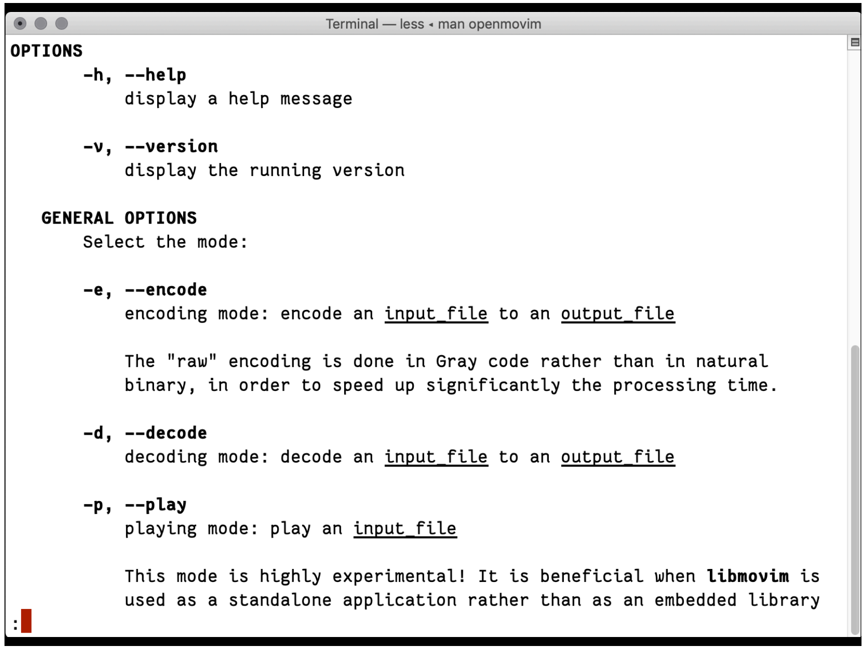
Moving Image

19

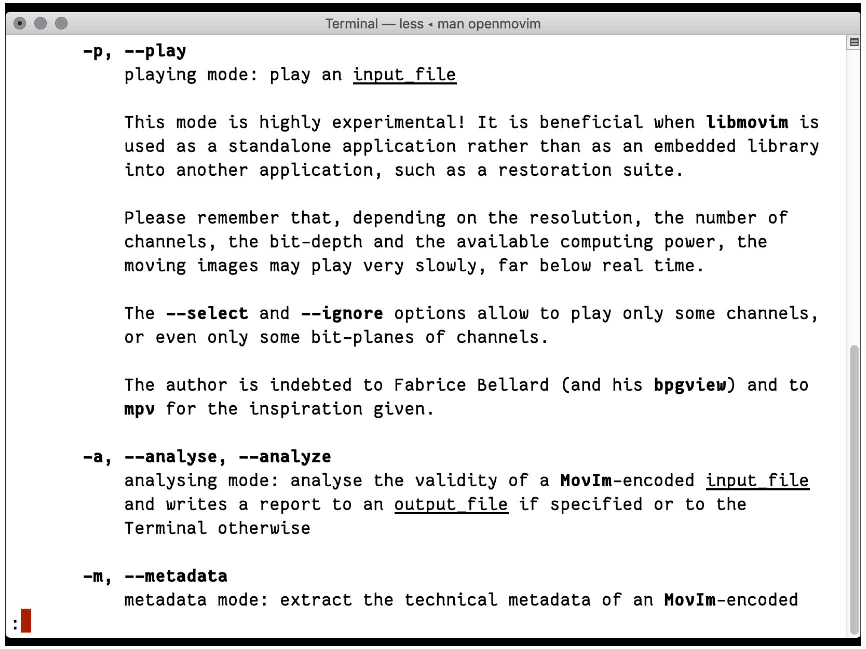


```
openMovIm(1)               AV Preservation by reto.ch               openMovIm(1)

NAME
       openmovim - Command-line interface to encode, decode, play and analyse
       moving images using 'libmovim'

SYNOPSIS
       openmovim (-e | -d | -p | -a | -m) -i input_file [-o output_file]

       openmovim (-c | -u) -i input_file [-o output_file]

       openmovim -h | -v

DESCRIPTION
       MovIm is a video codec for moving images specifically designed for both
       conservation and restoration purposes.

       libmovim is a C library implementing the MovIm video codec.

       openmovim is a Bash command-line interface to libmovim allowing to
       encode, decode, play and analyse moving images.

       The openMovIm package includes the libmovim library and its openmovim
       command-line interface.
:
```

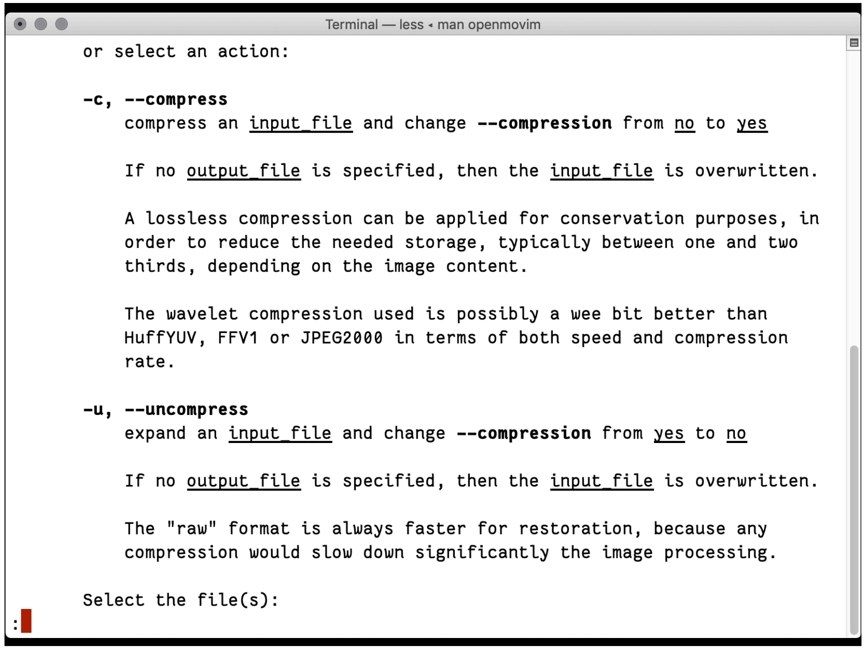
20



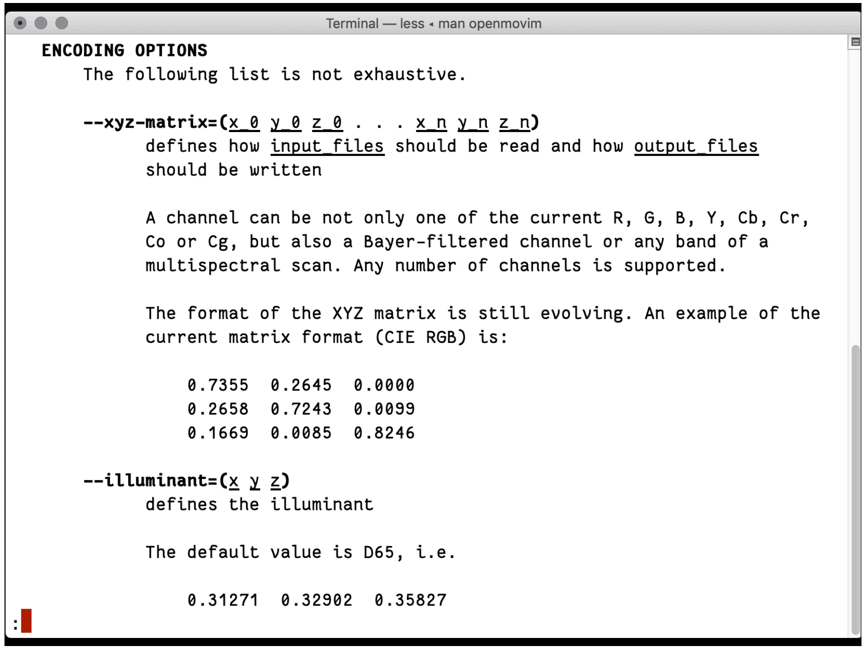
```
OPTIONS
       -h, --help
           display a help message

       -v, --version
           display the running version

   GENERAL OPTIONS
       Select the mode:

       -e, --encode
           encoding mode: encode an input_file to an output_file

           The "raw" encoding is done in Gray code rather than in natural
           binary, in order to speed up significantly the processing time.

       -d, --decode
           decoding mode: decode an input_file to an output_file

       -p, --play
           playing mode: play an input_file

           This mode is highly experimental! It is beneficial when libmovim is
           used as a standalone application rather than as an embedded library
:
```



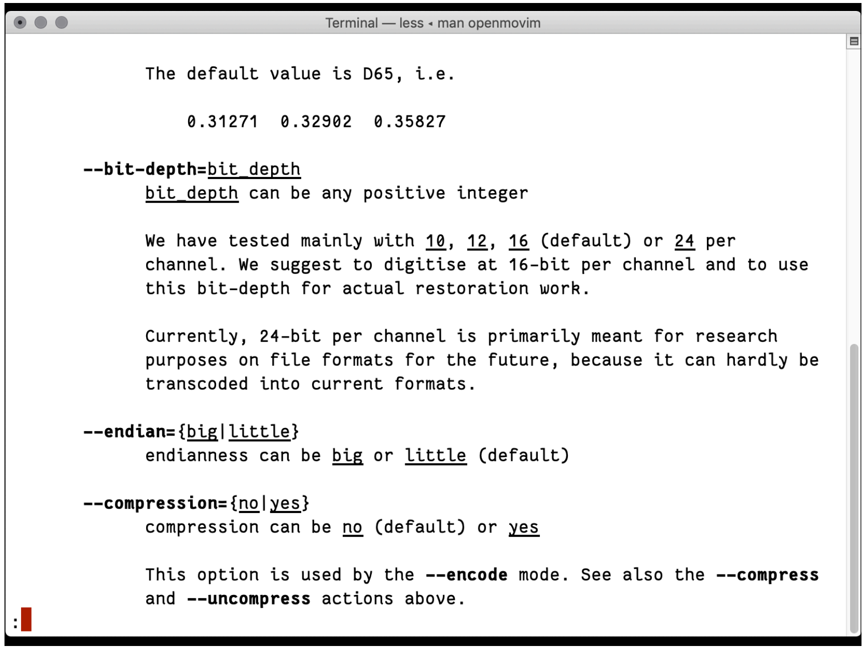
```
       -p, --play
           playing mode: play an input_file

           This mode is highly experimental! It is beneficial when libmovim is
           used as a standalone application rather than as an embedded library
           into another application, such as a restoration suite.

           Please remember that, depending on the resolution, the number of
           channels, the bit-depth and the available computing power, the
           moving images may play very slowly, far below real time.

           The --select and --ignore options allow to play only some channels,
           or even only some bit-planes of channels.

           The author is indebted to Fabrice Bellard (and his bpgview) and to
           mpv for the inspiration given.

       -a, --analyse, --analyze
           analysing mode: analyse the validity of a MovIm-encoded input_file
           and writes a report to an output_file if specified or to the
           Terminal otherwise

       -m, --metadata
           metadata mode: extract the technical metadata of an MovIm-encoded
:
```



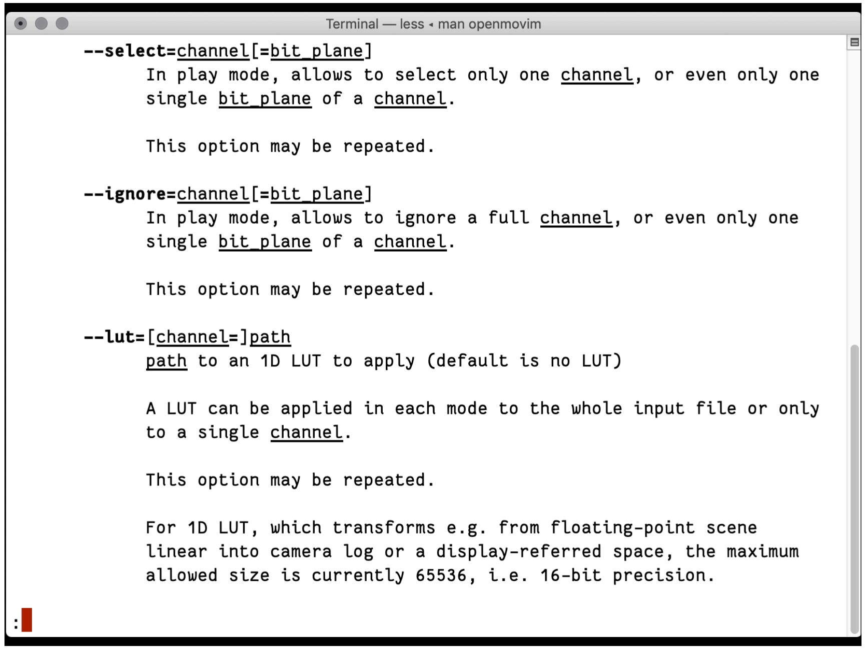
```
       or select an action:

       -c, --compress
           compress an input_file and change --compression from no to yes

           If no output_file is specified, then the input_file is overwritten.

           A lossless compression can be applied for conservation purposes, in
           order to reduce the needed storage, typically between one and two
           thirds, depending on the image content.

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

       -u, --uncompress
           expand an input_file and change --compression from yes to no

           If no output_file is specified, then the input_file is overwritten.

           The "raw" format is always faster for restoration, because any
           compression would slow down significantly the image processing.

       Select the file(s):
:
```



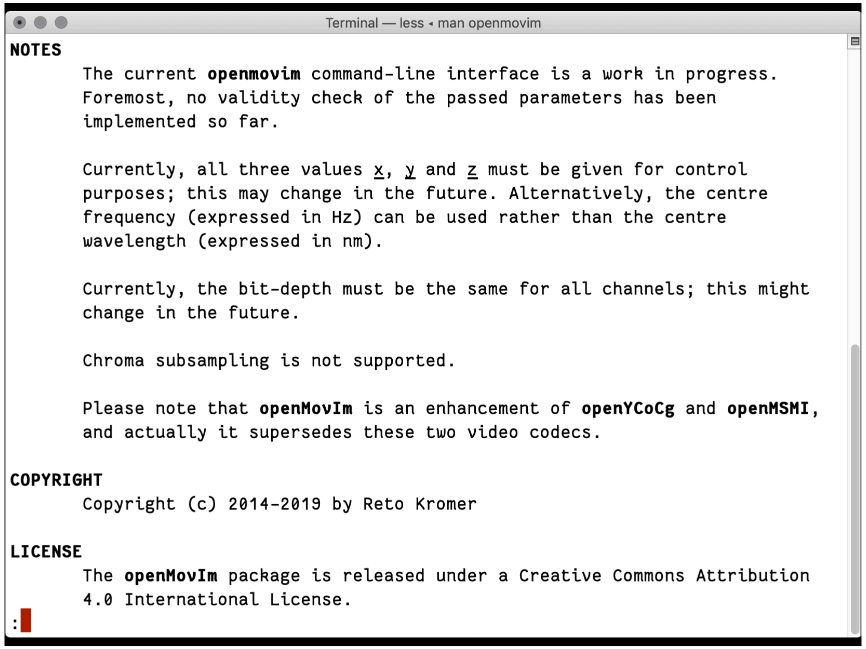
```
   ENCODING OPTIONS
       The following list is not exhaustive.

       --xyz-matrix=(x_0 y_0 z_0 . . . x_n y_n z_n)
              defines how input_files should be read and how output_files
              should be written

              A channel can be not only one of the current R, G, B, Y, Cb, Cr,
              Co or Cg, but also a Bayer-filtered channel or any band of a
              multispectral scan. Any number of channels is supported.

              The format of the XYZ matrix is still evolving. An example of the
              current matrix format (CIE RGB) is:

                  0.7355  0.2645  0.0000
                  0.2658  0.7243  0.0099
                  0.1669  0.0085  0.8246

       --illuminant=(x y z)
              defines the illuminant

              The default value is D65, i.e.

                  0.31271  0.32902  0.35827
:
```



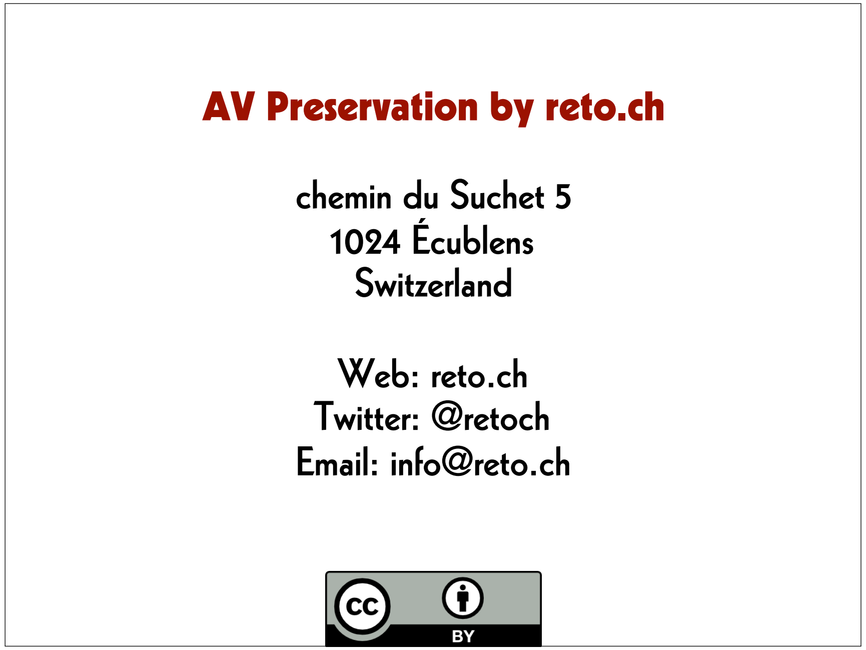
```
          The default value is D65, i.e.

              0.31271  0.32902  0.35827

     --bit-depth=bit_depth
          bit_depth can be any positive integer

          We have tested mainly with 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.

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

     --endian={big|little}
          endianness can be big or little (default)

     --compression={no|yes}
          compression can be no (default) or yes

          This option is used by the --encode mode. See also the --compress
          and --uncompress actions above.
:
```



```
       --select=channel[=bit_plane]
            In play mode, allows to select only one channel, or even only one
            single bit_plane of a channel.

            This option may be repeated.

       --ignore=channel[=bit_plane]
            In play mode, allows to ignore a full channel, or even only one
            single bit_plane of a channel.

            This option may be repeated.

       --lut=[channel=]path
            path to an 1D LUT to apply (default is no LUT)

            A LUT can be applied in each mode to the whole input file or only
            to a single channel.

            This option may be repeated.

            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.

:
```



```
NOTES
       The current openmovim command-line interface is a work in progress.
       Foremost, no validity check of the passed parameters has been
       implemented so far.

       Currently, all three values x, y and z must be given for control
       purposes; this may change in the future. Alternatively, the centre
       frequency (expressed in Hz) can be used rather than the centre
       wavelength (expressed in nm).

       Currently, the bit-depth must be the same for all channels; this might
       change in the future.

       Chroma subsampling is not supported.

       Please note that openMovIm is an enhancement of openYCoCg and openMSMI,
       and actually it supersedes these two video codecs.

COPYRIGHT
       Copyright (c) 2014-2019 by Reto Kromer

LICENSE
       The openMovIm package is released under a Creative Commons Attribution
       4.0 International License.
:
```



AV Preservation by reto.ch

chemin du Suchet 5
1024 Écublens
Switzerland

Web: reto.ch
Twitter: @retoch
Email: info@reto.ch