Beyond RGB:
Multispectral Imaging and FFV1

Reto Kromer • AV Preservation by reto.ch

No Time to Wait!

Rough Consensus and Running Archives

British Film Institute, London, United Queendom

25–26 October 2018

Beyond RGB

Past

The so-called «raw» scans and FFV1

Present

Multispectral Imaging and FFV1

Future

Hyperspectral Imaging

Past

Beyond RGB

Past

The so-called «raw» scans and FFV1

Present

Multispectral Imaging and FFV1

Future

Hyperspectral Imaging

Mid$_{old}$ Experimentations

from:

- AVI / CineForm (VC-5) with Bayer
- MOV / CineForm (VC-5) with Bayer
- MOV / ProRes

to:

- Matroska / FFV1 after de-mosaicking
- Matroska / ProRes

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

Mid$_{new}$ Experimentations

from:

- AVI / CineForm (VC-5) with Bayer
- MOV / CineForm (VC-5) with Bayer
- MOV / ProRes

to:

- Matroska / FFV1 version 4
- Matroska / ProRes with 12-bit support

Present

Beyond RGB

Past

The so-called «raw» scans and FFV1

Present

Multispectral Imaging and FFV1

Future

Hyperspectral Imaging

New Experimentations

from:

- MXF / OpenEXR
- flavour of NUT / multiple RGB48
- flavour of NUT / RGB72 or $Y'C_BC_R$ 24-bit
- flavour of NUT / expanded OpenEXR

to:

- **Matroska / FFV1 version 4 using RGB48, RGB72, floats and additional metadata**

© 1993–2018 by Reto Kromer
This website is released under a CC BY 4.0 License.

Reports from the Field

Working with Multispectral Moving Images

As it is current practice since years in other fields of fine arts' conservation and restoration, there is an increasing need to work with multispectral content for moving images as well. We wish to present to our colleagues and friends some results of our research in this field, and to share with them our thoughts.

The *openMSMI* package, which we are currently testing in an alpha release, contains the following elements:

- *MSMI* is an experimental video codec for multispectral moving images, specifically designed for conservation and restoration purposes;
- *libmsmi* is a C library implementing the *MSMI* video codec;
- *openmsmi* is a Bash command-line interface to *libmsmi* allowing to encode, decode, play and analyse multispectral moving images.

WHAT, WHEN AND WHERE

How to build in-house a multispectral scanner for moving images? What are the challenges? How to use the multispectral imaging in conservation and restoration workflows? What are the benefits?

A short talk with demonstrations by Reto Kromer, followed by an informal Q&A session and light refreshments, on Friday the 28th of September 2018

© 1993–2018 by Reto Kromer
This website is released under a CC BY 4.0 License.

- *MSMI* is an experimental video codec for multispectral moving images, specifically designed for conservation and restoration purposes;
- *libmsmi* is a C library implementing the *MSMI* video codec;
- *openmsmi* is a Bash command-line interface to *libmsmi* allowing to encode, decode, play and analyse multispectral moving images.

WHAT, WHEN AND WHERE

How to build in-house a multispectral scanner for moving images? What are the challenges? How to use the multispectral imaging in conservation and restoration workflows? What are the benefits?

A short talk with demonstrations by Reto Kromer, followed by an informal Q&A session and light refreshments, on Friday the 28th of September 2018, starting at 11 o'clock sharp, at AV Preservation by reto.ch in Écublens (a suburb of Lausanne, Switzerland). Before that, everyone is free to realise multispectral digitisations with our setting which may be used for the following demonstrations, and tea is served.

As usual, the admission is free, but an inscription is required, because only eight places are available at our small facility. This will be a technical meeting and the attendees should be familiar with high-school maths, including functions, linear transformations and elementary matrix operations.

2018–09–25

```
[08:29:50]reto@Castor:~/Desktop$ openmsmi -h
Syntax:
  openmsmi (-e | -d | -p | -a | -m) -i <input_file> [-o <output_file>]
  openmsmi -h | -v
Parameters:
  -e  encode mode
  -d  decode mode
  -p  play mode
  -a  analyse mode
  -m  metadata mode
  -i  path to input file
  -o  path to output file
  -h  this help
  -v  installed version
See also:
  man openmsmi
  https://avpres.net/openMSMI/
About:
  Abstract: CLI to encode, decode, play and analyse multispectral moving images
  Version:  2018-10-13
  Path:     /usr/local/bin
[08:30:17]reto@Castor:~/Desktop$
```

NAME

openmsmi - Command-line interface to encode, decode, play and analyse multispectral moving images

SYNOPSIS

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

openmsmi -h | **-v**

DESCRIPTION

MSMI is a video codec for handling multispectral moving images, specifically designed for conservation and restoration purposes.

libmsmi is a C library implementing the **MSMI** video codec.

openmsmi is a Bash command-line interface to **libmsmi** allowing to encode, decode, play and analyse multispectral moving images.

The **openMSMI** package includes the **libmsmi** library and its **openmsmi** command-line interface.

OPTIONS

:

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 highly experimental! Please remember that, depending on the number of spectral bands and the available computing power, the moving images may play very slowly, far below real time.

The **--select** option and the **--ignore** option allow to play only some

:

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

This mode is highly experimental! Please remember that, depending on the number of spectral bands and the available computing power, the moving images may play very slowly, far below real time.

The **--select** option and the **--ignore** option allow to play only some spectral bands, or even only some bit_planes of spectral bands.

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

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

-m, --metadata
metadata mode: extract the technical metadata of an MSMI encoded input_file without analysing its validity (including the wavelength vector), 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. A **--match-vector** can be used to force the interpretation of the provided data.

    In decoding, analysing or metadata mode, the container formats NUT (.nut) and Matroska (.mkv) have been tested as wrappers for the MSMI video codec.

-o output_file, **--output=**output_file
    In encoding mode, the container formats NUT (.nut) and Matroska (.mkv) have been tested as wrappers for the MSMI video codec. Please note that such a file will work only with **libmsmi** and currently it will not play at all with VLC.

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

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

ENCODING OPTIONS

:

ENCODING OPTIONS

The following list is not exhaustive.

--wavelength-vector=(wavelength_0 . . . wavelength_n**)**

Each spectral band is defined by its medium wavelength. The wavelength is expressed in nm and decimals are not allowed.

For readability, we suggest to order the spectral bands by increasing wavelength.

In theory, any number of spectral bands is possible. Currently, we suggest to work with a multiple of 3.

--match-vector=(band_0 . . . band_n**)**

defines how input_files other than MSMI files should be read and how output_files other than MSMI files should be written

The format of the match vector is still evolving.

--bit-depth={16|24}

bit-depth can be 16 (default) or 24 per spectral band

Currently, the 24-bit per band flavour is primarily meant for research purposes on file formats for the future, because it can

:

--bit-depth={16|24}
bit-depth can be 16 (default) or 24 per spectral band

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

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

A lossless compression can be applied. However, please remember that 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 compression rate is typically between one and two thirds, depending on the image content.

OTHER OPTIONS
The following list is not exhaustive.

--report-fmt={json|plain|xml}
report format can be json, plain text (default) or xml

:

OTHER OPTIONS

The following list is not exhaustive.

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

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

--select=band[=bit_plane]

In play mode, allows to select only one spectral band, or even only one single bit_plane of a spectral band.

This option may be repeated.

--ignore=band[=bit_plane]

In play mode, allows to ignore a full spectral band, or even only one single bit_plane of a spectral band.

This option may be repeated.

NOTES

At the moment, if the number of spectral bands is a multiple of 3, then the processing of the file is faster. (The majority of our tests are done with 15 bands.)

The application programming interface of **libmsmi** is not stable yet,

:

This option may be repeated.

NOTES

At the moment, if the number of spectral bands is a multiple of 3, then the processing of the file is faster. (The majority of our tests are done with 15 bands.)

The application programming interface of **libmsmi** is not stable yet, therefore its current **openmsmi** command-line interface should be considered a work in progress.

COPYRIGHT

Copyright (c) 2017-2018 by Reto Kromer

LICENSE

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

DISCLAIMER

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

Future

Beyond RGB

Past

The so-called «raw» scans and FFV1

Present

Multispectral Imaging and FFV1

Future

Hyperspectral Imaging

AV Preservation by reto.ch

chemin du Suchet 5
1024 Écublens
Switzerland

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