MIPP and SMHI/DMI Common Processing Environment

Release v0.9.2

Lars Orum Rasmussen

March 30, 2015

Contents

This is a presentation of:

python-mipp an introduction

mipp is a Meteorological Ingest-Processing Package (http://github.com/loerum/mipp).

> It's a Python library and it's main task is to convert low level satellite data into a format understood by mpop (http://github.com/mraspaud/mpop). The primary purpose is to support Geostationary satellite data (level 1.5) but there is also support for the reading of some polar orbiting SAR data (see below).
>
> A more sophisticated interface to satellite data objects is supported by mpop.

Currently it handles data from all current Meteosat Second Generation (MSG) satellites, Meteosat 7, GOES 11-15, MTSAT's, and GOMS, all as retrieved via EUMETCast:

```
L-000-MTP___-MET7________-00_7_057E-PRO______-201002261600-__
L-000-MTP___-MET7________-00_7_057E-000001___-201002261600-C_
L-000-MTP___-MET7________-00_7_057E-000002___-201002261600-C_
L-000-MTP___-MET7________-00_7_057E-000003___-201002261600-C_
...
...
L-000-MSG2__-GOES11______-00_7_135W-PRO______-201002261600-__
L-000-MSG2__-GOES11______-00_7_135W-000001___-201002261600-C_
L-000-MSG2__-GOES11______-00_7_135W-000002___-201002261600-C_
L-000-MSG2__-GOES11______-00_7_135W-000003___-201002261600-C_
...
...
```

In addition mipp handles Synthetic Apperture Radar (SAR) data from Terrscan-X, Cosmo-Sky Med, and Radarsat 2.

mipp will:

- Decompress XRIT files (if Eumetsat's xRITDecompress is available). Software to uncompress HRIT/XRIT can be obtained from EUMETSAT (register and download the Public Wavelet Transform Decompression Library Software). Please be sure to set the environment variable XRIT_DECOMPRESS_PATH to point to the full path to the decompression software, e.g. /usr/bin/xRITDecompress. Also you can specify where the decompressed files should be stored after decompression, using the environment variable XRIT_DECOMPRESS_OUTDIR. If this variable is not set the decompressed files will be found in the same directory as the compressed ones.
- Decode/strip-off (according to [CGMS], [MTP], [SGS]) XRIT headers and collect meta-data.
- Catenate image data into a numpy-array.
  - if needed, convert 10 bit data to 16 bit
  - if a region is defined (by a slice or center, size), only read what is specified.

Note:

- MET7: not calibrated.
- GOES, METSAT: calibration constants to Kelvin or Radiance (not Reflectance).

1.1 Code Layout

`xrit.py`

It knows about the genric HRIT/XRIT format

- `headers = read_headers(file_handle)`

`MTP.py`

It knows about the specific format OpenMTP for MET7

- `mda = read_metadata(prologue, image_file)`

`SGS.py`

It knows about the specific format Support Ground Segments for GOES and MTSAT

- `mda = read_metadata(prologue, image_files)`

`sat.py`

It knows about satellites base on configurations files. It returns a slice-able object (see below).

- `image = load('met7', time_stamp, channel, mask=False, calibrated=True)`
- `image = load_files(prologue, image_files, **kwarg)`

`slicer.py`

It knows how to slice satellite images (return from `load(...)`). It returns meta-data and a numpy array.

- `mda, image_data = image[1300:1800,220:520]`
- `mda, image_data = image(center, size)`

Utilities

`cfg.py`

It knows how to read configuration files, describing satellites (see below).

`convert.py`

10 to 16 byte converter (uses a C extension)

`bin_reader.py`

It reads binary data (network byte order)

- `read_uint1(buf)`
- `read_uint2(buf)`
- `read_float4(buf)`
- ...

`mda.py`

A simple (anonymous) metadata reader and writer

`geosnav.py`

It will convert from/to pixel coordinates to/from geographical longitude, latitude coordinates.

1.2 Example definition of a satellite

```
# An item like:
#   name = value
# is read in python like:
#   try:
#       name = eval(value)
#   except:
#       name = str(value)
#
```

```
[satellite]
satname = 'meteosat'
number = '07'
instruments = ('mviri',)
projection = 'geos(57.0)'

[mviri-level2]
format = 'mipp'

[mviri-level1]
format = 'xrit/MTP'
dir = '/data/eumetcast/in'
filename = 'L-000-MTP___-MET7________-%(channel)s_057E-%(segment)s-%Y%m%d%H%M-__'

[mviri-1]
name = '00_7'
frequency = (0.5, 0.7, 0.9)
resolution = 2248.49
size = (5000, 5000)

[mviri-2]
name = '06_4'
frequency = (5.7, 6.4, 7.1)
resolution = 4496.98
size = (2500, 2500)

[mviri-3]
name = '11_5'
frequency = (10.5, 11.5, 12.5)
resolution = 4496.98
size = (2500, 2500)
```

1.3 Usage

```
import xrit

image = xrit.sat.load('meteosat07', datetime(2010, 2, 1, 10, 0), '00_7', mask=True)
mda, image_data = image(center=(50., 10.), size=(600, 500))
print mda
fname = './' + mda.product_name + '.dat'
print >>sys.stderr, 'Writing', fname
fp = open(fname, "wb")
image_data.tofile(fp)
fp.close()
```

1.4 Examples of the usage of some lower level tools

Here an example how to get the observation times (embedded in the 'Image Segment Line Quality' record) of each scanline in a segment:

```
import mipp.xrit.MSG

segfile = "/local_disk/data/MSG/HRIT/H-000-MSG3__-MSG3________-WV_062___-000002___-201311211300-_
lineq = mipp.xrit.MSG.get_scanline_quality(segfile)
print lineq[0]
```

```
(465, datetime.datetime(2013, 11, 21, 13, 1, 48, 924000), 1, 1, 0)
```

1.5 A script, process_fsd

The script is intended for work on other geostationary data than the MSG (Meteosat) data, the so-called Foreign Satellite Data (FSD). That is e.g. GOES, MTSAT and COMS.

```
process_fsd --check-satellite <prologue-file>
    check if we handle this satellite

process_fsd --check [-l] <prologue-file>
    check if number of image segments are as planned
    -l, list corresponding image segment files

process_fsd --decompress [-o<output-dir>] <file> ... <file>
    decompress files to output-dir (default is working directory)
    -l, list decompressed files

process_fsd --metadata <prologue-file> <image-segment> ... <image-segment>
    print meta-data

process_fsd [-o<output-dir>] <prologue-file> <image-segment> ... <image-segment>
    it will binary dump image-data and ascii dump of meta-data)
```

CHAPTER 2

Calibration comments

2.1 MSG series

The calibration of the meteosat second generation satellites is done according to the Eumetsat documents [refl], [bt].

2.1.1 Reflectances

The visible and near infrared channels are calibrated according to the following formula:

r = R / I

where

- r is the bidirectional reflectance factor
- R is the measured radiance
- I is the solar irradiance

R is derived from the xRIT data, and I is given in [refl].

In [refl] the additional following corrections are applied:

- sun-earth distance correction
- cosine of the solar zenith angle.

CHAPTER 3

The `mipp` API

3.1 MIPP

exception `mipp.`**`CalibrationError`**

exception `mipp.`**`ConfigReaderError`**

exception `mipp.`**`DecodeError`**

exception `mipp.`**`MippError`**

exception `mipp.`**`NavigationError`**

exception `mipp.`**`NoFiles`**

exception `mipp.`**`ReaderError`**

exception `mipp.`**`UnknownSatellite`**

`mipp.`**`strptime`**`()`
string, format -> new datetime parsed from a string (like time.strptime()).

3.1.1 Metadata

class `mipp.mda.`**`Metadata`**

`dont_eval` = **('satnumber',)**

`ignore_attributes` = ()

`read``(`*file_name*`)`
Read until empty line, 'EOH' or 'EOF'.

`save``(`*file_name*`)`

`token` = **':'**

`mipp.mda.`**`mslice`**`(`*mda*`)`

3.1.2 Configuration

`mipp.cfg.`**`read_config`**`(`*satname*, *instrument=''*`)`

3.1.3 Logging

class mipp.log.**NullHandler**
Empty handler.

emit (*record*)
Record a message.

mipp.log.**debug_on** ()
Turn debugging logging on.

mipp.log.**get_logger** (*name*)
Return logger with null handle

mipp.log.**logging_off** ()
Turn logging off.

mipp.log.**logging_on** (*level=None*)
Turn logging on.

3.2 XRIT input layer

3.2.1 MSG

This module will read MSG level1.5 files, format documented in: 'MSG Level 1.5 Image Data Format Description', EUM/MSG/ICD/105, v5A, 22 August 2007

mipp.xrit.MSG.**read_metadata** (*prologue*, *image_files*, *epilogue*)
Selected items from the MSG prologue file.

3.2.2 GOMS

Read Electro L N1 HRIT files.

mipp.xrit.GOMS.**read_epiheader** (*fp*)

mipp.xrit.GOMS.**read_metadata** (*prologue*, *image_files*, *epilogue*)
Selected items from the Electro L N1 prolog file.

mipp.xrit.GOMS.**read_proheader** (*fp*)

3.2.3 MTP

This module will read satellit data files in OpenMTP format (eg. Meteosat-7 prolog file). Format described in: 'The Meteosat Archive; Format Guide No. 1; Basic Imagery: OpenMTP Format'; EUM FG 1; Rev 2.1; April 2000

mipp.xrit.MTP.**read_metadata** (*prologue*, *image_files*)
Selected items from the Meteosat-7 prolog file.

3.2.4 SGS

This module will read satellit data files in SGS (Support Ground Segments) format (eg. GOES, MTSAT). Format described in: 'MSG Ground Segment LRIT/HRIT Mission Specific Implementation'; EUM/MSG/SPE/057; Issue 6; 21 June 2006

mipp.xrit.SGS.**read_metadata** (*prologue*, *image_files*)
Selected items from the GOES image data files (not much information in prologue).

3.2.5 _xrit

This module will read LRIT/HRIT headers. Format described in: "LRIT/HRIT Global Specification"; CGMS 03; Issue 2.6; 12 August 1999 "MSG Ground Segment LRIT/HRIT Mission Specific Implementation"; EUM/MSG/SPE/057; Issue 6; 21 June 2006

`mipp.xrit._xrit.`**`read_prologue`**(*file_name*)

`mipp.xrit._xrit.`**`read_epilogue`**(*file_name*)

`mipp.xrit._xrit.`**`read_imagedata`**(*file_name*)

`mipp.xrit._xrit.`**`read_gts_message`**(*file_name*)

`mipp.xrit._xrit.`**`read_mpef`**(*file_name*)

`mipp.xrit._xrit.`**`read_mpef_clm`**(*file_name*)

`mipp.xrit._xrit.`**`decompress`**(*infile*, *outdir='.'*)

> Will decompress an XRIT data file and return the path to the decompressed file. It expect to find Eumetsat's xRITDecompress through the environment variable XRIT_DECOMPRESS_PATH

`mipp.xrit._xrit.`**`list`**(*file_name*, *dump_data=False*)

3.2.6 bin_reader

`mipp.xrit.bin_reader.`**`read_cds_expanded_time`**(*buf*)

`mipp.xrit.bin_reader.`**`read_cds_time`**(*buf*)

`mipp.xrit.bin_reader.`**`read_cuc_time`**(*buf*, *coarce*, *fine*)

`mipp.xrit.bin_reader.`**`read_float4`**(*buf*)

`mipp.xrit.bin_reader.`**`read_float8`**(*buf*)

`mipp.xrit.bin_reader.`**`read_int2`**(*buf*)

`mipp.xrit.bin_reader.`**`read_int4`**(*buf*)

`mipp.xrit.bin_reader.`**`read_uint1`**(*buf*)

`mipp.xrit.bin_reader.`**`read_uint2`**(*buf*)

`mipp.xrit.bin_reader.`**`read_uint4`**(*buf*)

`mipp.xrit.bin_reader.`**`read_uint8`**(*buf*)

3.2.7 loader

class `mipp.xrit.loader.`**`ImageLoader`**(*mda*, *image_files*, *mask=False*, *calibrate=False*)

> **`raw_slicing`**(*item*)
>
> > Raw slicing, no rotation of image.

3.2.8 convert

`mipp.xrit.convert.`**`dec10216`**(*in_buffer*)

`mipp.xrit.convert.`**`hrpt_dec10216`**(*in_buffer*)

3.2.9 sat

mipp.xrit.sat.**load_meteosat07**(*time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load_meteosat09**(*time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load_goes11**(*time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load_goes12**(*time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load_goes13**(*time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load_mtsat1r**(*time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load_mtsat2**(*time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load_electrol**(*time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load**(*satname*, *time_stamp*, *channel*, ***kwarg*)

mipp.xrit.sat.**load_files**(*prologue*, *image_files*, *epilogue=None*, ***kwarg*)

3.2.10 Metadata

class mipp.xrit.mda.**Metadata**

ignore_attributes = ('line_offset', 'first_pixel', 'coff', 'loff', 'image_data', 'boundaries')

token = ':'

3.3 XSAR input layer

3.3.1 Cosmo Sky-med

3.3.2 Radarsat-2

3.3.3 Terra-SAR X

3.3.4 sat

mipp.xsar.sat.**load**(*satname*, *time_stamp*, *channel*, ***kwarg*)

3.3.5 Metadata

class mipp.xsar.mda.**Metadata**

ignore_attributes = ('data', 'calibrate', 'tiepoints')

token = ':'

CHAPTER 4

Indices and tables

- *genindex*
- *modindex*
- *search*

Bibliography

[CGMS] LRIT/HRIT Global Specification; CGMS 03; Issue 2.6; 12 August 1999 “MSG Ground Segment LRIT/HRIT Mission Specific Implementation” EUM/MSG/SPE/057; Issue 6; 21 June 2006

[MTP] “The Meteosat Archive; Format Guide No. 1; Basic Imagery: OpenMTP Format”; EUM FG 1; Rev 2.1; April 2000

[SGS] “MSG Ground Segment LRIT/HRIT Mission Specific Implementation”; EUM/MSG/SPE/057; Issue 6; 21 June 2006

[refl] “Conversion from radiances to reflectances for SEVIRI warm channels” EUM/MET/TEN/12/0332

[bt] “The Conversion from Effective Radiances to Equivalent Brightness Temperatures” EUM/MET/TEN/11/0569