Interface to Image Reconstruction

Éric Thiébaut[1], John Young[2], Jonathan Léger[1], Michel Tallon[1], Gilles Duvert[3], Guillaume Mella[3], Laurent Bourges[3] & Martin Vannier[4]

[1] *CRAL,* [2] *University of Cambridge,* [3] *JMMC/IPAG,* [4] *Lagrange*

July 19, 2018

1 Introduction

Many image reconstruction algorithms have been developed to process optical interferometric data. Using these algorithms may require substantial expertise (for instance you may have to learn a specific programming language). One of the objectives of the OPTICON JRA4 is to provide a common graphical user interface (GUI) to drive such imaging algorithms in a *user friendly* way. For obvious reasons, we also want to avoid rewriting the image reconstruction algorithms as much as possible. A model of interaction between these two pieces of software under these constraints has to be developed. This document aims at specifying a common software interface for image reconstruction algorithms.

On the image reconstruction side, this involves specifying the inputs completely and precisely: OI-FITS file for the data, the initial image, and other settings such as the regularization and its hyper-parameters.[1] Given these inputs, the image reconstruction algorithm should start the reconstruction producing two kind of outputs:

Intermediate outputs: These consist of iteration information printed on the standard output (for instance) and captured by the GUI and also the current solution in the form of an image, the corresponding model of the measurements (*e.g.*, the complex visibilities). These outputs will be used by the GUI to display the current solution, to compare the actual data with the current model, *etc.*

Final outputs: These are essentially the final image and consistent information about the pixel size, the orientation, *etc.* but also about the input

[1] In a previous discussion, John Young suggested using a text file with a simple format to give the values of the input parameters. The format of this file must be simple and such that it can be easily edited by a human and easily read by a software program. Here a different proposition is made which exploits the flexibility of the FITS format.

data (*i.e.* to avoid a *traceability issue*), the reconstruction method and its parameters.

A few related points have to be considered:

1. The command line image reconstruction algorithms should (or not?) switch off their own display capabilities.
2. There must be a way to interrupt the image reconstruction algorithms so as to make sure they do release all their resources (no zombie process, undetached shared memory, etc.).
3. Restarting an image reconstruction (to continue the iterations or perhaps with a few settings changed) should simply be a matter of re-starting the algorithm with, as its input image, the last image produced.
4. In case of errors, we must adopt some kind of conventions to make sure that the GUI can properly account for the errors.
5. It would be helpful for the GUI to incorporate a facility whereby the user can pause the reconstruction, translate the image so that its centre-of-gravity is in the centre of the field, then continue the reconstruction.

To summarize, on entry:

- the data;
- image parameters (dimensions, pixel size, *etc.*);
- initial image;
- regularization and hyper-parameters;

On output (for the final and intermediate results), *in addition to the above list*:

- the current/final image;
- the model of the data;
- other relevant information;

To make the interfacing as simple as possible, it is proposed in this document to use a single entry file containing all input parameters and data and a single output file with all the results (either intermediate or final). The flexibility of the FITS file format is exploited to store all these information and data.

All image reconstruction software must be able to run from the command line and with two arguments: the names of the input and output files (possibly with some flag to indicate that the software must be run in a specific way). For instance:

```
command --batch input output
```

Table 1: FITS keywords used to specify the input parameters. The *Image Parameters* are stored in the HDU which contains the initial image. This HDU is specified by the value of the `INIT_IMG` keyword, which gives its `HDUNAME`. All other parameters are in a binary table HDU with `EXTNAME` of `'IMAGE-OI INPUT PARAM'`.

Data Selection		
Keyword	Type	Description
`TARGET`	string	Identifier of the target object to reconstruct
`WAVE_MIN`	real	Minimum wavelength to select (in meters)
`WAVE_MAX`	real	Maximum wavelength to select (in meters)
`USE_VIS`	string	Complex visibility data to consider if any†
`USE_VIS2`	logical	Use squared visibility data if any
`USE_T3`	string	Bispectrum data to consider if any†
		† value can be: `'NONE'`, `'ALL'`, `'AMP'` or `'PHI'`.
Algorithm Settings		
Keyword	Type	Description
`INIT_IMG`	string	Identifier of the initial image
`MAXITER`	integer	Maximum number of iterations to run
`RGL_NAME`	string	Name of the regularization method
`AUTO_WGT`	logical	Automatic regularization weight
`RGL_WGT`	real	Weight of the regularization
`RGL_PRIO`	string	Identifier of the HDU with the prior image
`FLUX`	real	Assumed total flux (1 is the default)
`FLUXERR`	real	Error bar for the total flux (0 means strict constraint)
Image Parameters		
Keyword	Type	Description
`HDUNAME`	string	Unique name for the image within the FITS file
`NAXIS1`	integer	First dimension of the image
`NAXIS2`	integer	Second dimension of the image
`CTYPE1`	string	`'RA---TAN'`
`CTYPE2`	string	`'DEC--TAN'`
`CDELTi`	real	Pixel increment along `i`-th dimension of the image (for `i` = 1 or 2)
`CUNITi`	string	Physical units for `CDELTi` and `CRVALi`; defaults to `'deg'` if omitted
`CRPIXi`	real	Index of reference pixel along `i`-th dimension (for `i` = 1 or 2); defaults to the geometric center of the field of view if omitted
`CRVALi`	real	Physical coordinate of reference pixel along `i`-th dimension (for `i` = 1 or 2) and relative to the center of the field of view; defaults to 0 if omitted

2 Input format

All existing algorithms take their input data in the OI-FITS format (Pauls et al., 2005). This format stores the optical interferometric data in FITS binary tables. More generally, a FITS file (Pence et al., 2010) consists of a number of so-called header data units (HDU), each HDU having an header part with scalar parameters specified by FITS keywords[2] and a data part. The header part obeys a specific format but it is textual and easy to read for an human. The data part usually consists of binary data stored in various forms. In this document we only consider FITS *images* (that is a multidimensional array of values of the same data type) and FITS *binary tables*. These are sufficient for our purposes.

By adding FITS HDU or introducing new FITS keywords, it is possible to provide any additional data (*e.g.* the initial image) and parameters required to run the image reconstruction process. Thus, it is proposed that the input file be a valid OI-FITS file (resulting from the merging of all the optical interferometric data to process) with the addition of HDU containing further binary data needed for the reconstruction (for instance, the initial image) and with scalar input settings provided as the values of FITS keywords. This is detailed in the following subsections.

Note that the FITS standard states that angular measurements expressed as floating-point values and specified with reserved keywords *should* be given in degrees.

The HDUs that are specific to the interface specification described in this document have `EXTNAME` values prefixed with `'IMAGE-OI'`. This is to distinguish them from OIFITS HDUs, which use the prefix `'OI_'`.

2.1 Scalar input parameters

In version 1 of OIFITS, the first (primary) HDU is almost empty. However, the draft version 2 of OIFITS introduces primary header keywords summarizing the data and giving its provenance. We propose, therefore, storing the scalar image reconstruction input parameters in a separate HDU. In anticipation of perhaps needing to store vector parameters in future, this HDU shall be a binary table HDU. This binary table shall have the `EXTNAME` keyword set to `'IMAGE-OI INPUT PARAM'` and shall contain all of the non-image parameters, with the exception of the pixel size and the dimensions of the reconstructed image which are specified by those of the initial image (see below), itself stored in a dedicated (image) HDU. Table 1 gives some examples of the input parameters.

2.2 Data selection

To keep things simple, any sophisticated selection, merging or editing of the data should be done by a separate tool. The image reconstruction software

[2]A FITS keyword consists of upper case latin letters, digits, hyphen or underscore characters and has at least one character and at most 8 characters.

applications shall assume that they receive clean input data. There are however a few parameters devoted to the selection of data. The `TARGET` keyword specifies the name of the target object to reconstruct. The value of this keyword should match one of the identifiers in the column `TARGET` in the `OI_TARGET` binary table of the OIFITS file. In order to restrict the types of interferometric data used for the reconstruction, keywords `USE_VIS`, `USE_VIS2` and `USE_T3` should be set with values specifying which complex visibility data, powerspectrum data and bispectrum data to use if any. More specifically, keywords `USE_VIS` and `USE_T3` take string values which are `'NONE'`, `'ALL'`, `'AMP'` or `'PHI'` to indicate whether all, none, only the amplitude or only the phase of such data are to be used. Keyword `USE_VIS2` has a boolean value indicating whether to use the powerspectrum data. Not all algorithms can use all types of data and the values of these keywords should be set (in the output file) so as to reflect what was really used.

Whatever these settings, the image reconstruction software must honor the `FLAG` column of the OI data. We recall that this field has a logical value which is true when the corresponding piece of data should be discarded and false when it should be considered.

Wavelength selection is discussed in the next section.

2.3 Wavelength range

The primary objective is to consider monochromatic image reconstruction. The interferometric data are generally available at many wavelengths. The result will, in fact, be a gray image of the target built from the data in the wavelength range (inclusive) specified by the FITS keywords `WAVE_MIN` and `WAVE_MAX`. The wavelength range is given in meters.

2.4 Initial image

All reconstruction algorithms are iterative and require an initial image to start with. The initial image is provided as a FITS image in one of the HDU of the input file. The primary HDU can be used to store the initial image (however *cf.* the discussion about which image to store in the primary HDU). The pixel size and the dimensions of the initial image determine that of the reconstructed image. The dimensions are given by the FITS keywords `NAXIS1` and `NAXIS2` while the pixel size is given by the FITS keywords `CDELT1` and `CDELT2` (both values must be the same).

It is not intended that the image reconstruction algorithm be able to deal with any possible world coordinate system (WCS) nor with any possible coordinate units. We therefore restrict this standard to WCS which can be fully specified by the `CRPIX`*i*, `CRVAL`*i*, `CDELT`*i*, `CTYPE`*i*, and `CUNIT`*i* keywords (where *i* is the axis number). Other WCS keywords like `CROTAn`, `PC`*i*_*j*, `DC`*i*_*j*, should not be specified as their default values (according to the FITS standard) are suitable for us. The same WCS conventions hold for any output image produced by

Table 2: FITS keywords used to specify the Wold Coordinate System(WCS). All these keywords refer to a given axis denoted as *i*.

Keyword	Description	Default
`CRPIX`*i*	Index of the reference pixel (start at 1 and can be fractional)	Geometric center
`CRVAL`*i*	Physical coordinate of ref. pixel	0.0
`CDELT`*i*	Physical coordinate increment	
`CTYPE`*i*	Coordinate name	`'RA---TAN'` for 1st axis, `'DEC--TAN'` for 2nd axis
`CUNIT`*i*	Units of the physical coordinates	degree

the reconstruction algorithm. For any external software to correctly display the images, the parameters of the WCS must be completely and correctly specified.

The conventions are that the first image axis corresponds to right ascension (RA) and second image axis corresponds to declination (DEC) both relative to the center of the field of view (FOV) specified by the keywords `CRPIX`*i* (in fractional pixel units). If keywords `CRPIX`*i* are omitted, they default to the geometric center of the FOV. The pixel size (which is specified by the absolute values of `CDELT1` and `CDELT2`) must be the same in both directions. Following standard conventions for display of a celestial image, to have the relative right ascension (RA) oriented toward East to correspond to the left (first columns) of the image, `CDELT1` should be strictly negative, while, to have the relative declination (DEC) oriented toward North to correspond to the top (last rows) of the image, `CDELT2` should be strictly positive. By default, the physical coordinate units are in degrees; otherwise `CUNIT`*i* may be `'deg'` for degrees or `'arcsec'` for arcseconds. Table 2 summarizes these rules.

The OI-FITS norm specifies that `RAEP0` and `DECEP0` in `OI_TARGET` are the coordinates of the phase center, which we assume to be the absolute (not relative) world coordinates of the reference pixel in the reconstructed image. With these conventions, the relative world coordinates of the FOV center (specified by keywords `CRVAL`*i*) should be $(0, 0)$.

`fitsverify` gives a warning if `CTYPE`*i* are omitted. Should these given as `'RA'` and `'DEC'`, meaning linear axes, or as `'RA---SIN'` and `'DEC--SIN'`, or as `'RA---TAN'` and `'DEC--TAN'` which specify the appropriate non-linear transformations between the tangent plane and the celestial sphere? Eric: In the document, I have assumed that `'RA---TAN'` and `'DEC--TAN'` are the correct ones; can we check whether this is true?

3 Output format

It is proposed that the output format be as similar as possible as the input format. The output file must provide the reconstructed image but also some information for analyzing the result. In particular, as each imaging algorithm may implement its own method for estimate the complex visibilities given the image of the object, it is necessary that the model of every fitted data point be computed by the algorithm itself rather than by another tool.

3.1 Input parameters

The output file shall contain a copy of the input parameters. These shall be stored in the same way as in the input file, except that the initial image is not stored in the primary HDU (this location is reserved for the final/current image as explained below).

The image reconstruction software shall write out values for all of the input parameters that it accepts, not just those that were provided in the input file. Thus the output file will contain default or automatically-chosen values for the remaining parameters.

Need to define conventions for EXTNAME and HDUNAME of all image HDUs

3.2 Final/current image

To compare the initial and the final images, they must be stored in different HDU, the FITS keyword `HDUNAME` is used to distinguish them (the EXTNAME keyword should not be used in the primary HDU). As explained previously, the dimensions, pixel size and orientation of the output image(s) are the same as the initial image.

As most image viewers[3] are only capable of displaying the image stored in the primary HDU, we suggest storing the initial image in the primary HDU of the input file, but storing the final or current image in the primary HDU of the output file. The idea is to have the most relevant image stored in the primary HDU. For the image reconstruction algorithm and for any software designed to display or analyze the results, the different images are distinguished by their names (given by their `HDUNAME` keyword). OK but the last image should have a specific name, perhaps set `LAST_IMG` keyword in the *output* parameters HDU with name of the final image and let all reconstructed images be called:

```
HDUNAME = 'IMAGE-OI OUTPUTn'
```

with *n* the iteration number.

In order to continue the iterations of a previous reconstruction run, the image reconstruction algorithm may be started with the final image instead of the initial one. To that end, there must be some means to specify the starting image for the reconstruction. This is the purpose of the `INIT_IMG` keyword (see Table 1) in the input parameters HDU which indicates the `HDUNAME` of the initial image. This `HDUNAME` must be unique within the file.

[3] not all softwares deal with FITS files

3.2.1 Output parameters

Any scalar output parameters from the image reconstruction shall be stored in a binary table HDU with `EXTNAME` of `'IMAGE-OI OUTPUT PARAM'`. Examples of scalar outputs include χ^2, any regularization parameters estimated from the data, and any model parameters that are not image pixels (for example stellar disk parameters in SPARCO). Standard output parameters are listed in Table 3.

Table 3: FITS keywords used to specify the output parameters. These keywords must be stored in a binary table HDU with `EXTNAME` of `'IMAGE-OI OUTPUT PARAM'`.

Algorithm Results		
Keyword	Type	Description
`LAST_IMG`	string	Identifier of the final image
`NITER`	integer	Total iterations done in the current program run
`CHISQ`	real	Reduced chi-squared
`FLUX`	real	Total image flux

3.3 Model of the data

The OI-FITS format (Pauls et al., 2005) specifies that optical interferometric data be stored in binary tables as columns with specific names. As there is no restriction that the tables only contain the columns specified by the standard, we propose to store the model of the data in the same tables by adding new columns. The additional columns have the prefix `'NS_MODEL_'` to distinguish them from the columns defined by the OIFITS standard. The names of the new columns are listed in Table 4. In this way it is very easy to compare the actual data and the corresponding model values as computed from the reconstructed image and the instrumental model assumed by the reconstruction algorithm. Another advantage of this convention is that the same format can be exploited to store the values given by model fitting software.

4 Command line tools

It would be useful to provide command line tools for creating and editing input files (e.g. to change a parameter), in addition to the graphical user interface. Such tools would be useful for scripting tests, for example.

Some example command lines are shown below. The first creates a new input file named **`myinput.fits`** and sets the initial image from the primary HDU of **`myimage.fits`**. The second command changes parameter values in an existing file.

```
oi-image create myinput.fits CDELT1=0.25 MAXITER=200 INIT_IMG=myimage.fits
oi-image edit myinput.fits RGL_ALPH=1e-3 USE_T3A=NONE
```

Table 4: Colums inserted in OI-FITS binary tables to store the values given by the model. *NWAVE* is the number of wavelengths.

New columns in OI_VIS table		
Label	Format	Description
NS_MODEL_VISAMP	D(*NWAVE*)	Model of the visibility amplitude
NS_MODEL_VISAMPERR	D(*NWAVE*)	Model of the error in visibility amplitude (optional)
NS_MODEL_VISPHI	D(*NWAVE*)	Model of the visibility phase in degrees
NS_MODEL_VISPHIERR	D(*NWAVE*)	Model of the error in visibility phase in degrees (optional)
New column in OI_VIS2 table		
Label	Format	Description
NS_MODEL_VIS2	D(*NWAVE*)	Model of the squared visibility
NS_MODEL_VIS2ERR	D(*NWAVE*)	Model of the error in squared visibility (optional)
New columns in OI_T3 table		
Label	Format	Description
NS_MODEL_T3AMP	D(*NWAVE*)	Model of the triple-product amplitude
NS_MODEL_T3AMPERR	D(*NWAVE*)	Model of the error in triple-product amplitude (optional)
NS_MODEL_T3PHI	D(*NWAVE*)	Model of the triple-product phase in degrees
NS_MODEL_T3PHIERR	D(*NWAVE*)	Model of the error in triple-product phase in degrees (optional)

A Glossary

DEC Declination;

FITS Flexible Image Transport System;

FOV Field of view;

GUI Graphical User Interface;

HDU Header Data Unit;

OI-FITS Optical Interferometric exchange data format;

RA Right Ascension;

WCS World Coordinate System;

References

T. A. Pauls, J. S. Young, W. D. Cotton, and J. D. Monnier. A data exchange standard for optical (visible/IR) interferometry. *PASP*, 117:1255–1262, November 2005. doi: 10.1086/444523.

W. D. Pence, L. Chiappetti, C. G. Page, R. A. Shaw, and E. Stobie. Definition of the Flexible Image Transport System (FITS), version 3.0. *A&A*, 524: A42, Nov 2010. ISSN 1432-0746. doi: 10.1051/0004-6361/201015362. URL http://fits.gsfc.nasa.gov/fits_standard.html.