

# A Data Exchange Standard for Optical (Visible/IR) Interferometry

Thomas A. Pauls<sup>a</sup>, John S. Young<sup>b</sup>, William D. Cotton<sup>c</sup>, John D. Monnier<sup>d</sup>

<sup>a</sup>Naval Research Laboratory, Code 7210, 4555 Overlook Avenue SW, Washington, DC USA 20375-5351;

<sup>b</sup>Astrophysics Group, Cavendish Laboratory, Madingley Road, CB3 0HE, UK;

<sup>c</sup>NRAO, 520 Edgemont Road, Charlottesville, VA USA 22903;

<sup>b</sup>University of Michigan, Department of Astronomy, Ann Arbor, MI USA 48109

## ABSTRACT

This paper describes a standard for exchanging calibrated data from optical (visible/infrared) stellar interferometers. The standard is based on the Flexible Image Transport System (FITS). The formal definition of the standard is contained in a separate document, the Format Specification. The latest version of the Format Specification is available from the website <http://www.mrao.cam.ac.uk/~jsy1001/exchange/>. This document gives an overview of the format, and explains some of the decisions taken in designing it.

**Keywords:** data formats, data exchange, FITS, interferometry

## 1. INTRODUCTION

The idea of a common data format for visible-wavelength and infrared astronomical interferometry (hence referred to collectively as “optical interferometry”) arose through discussions at the June 2000 NSF-sponsored meeting in Socorro.<sup>1</sup> Since 2001, T.P. and J.S.Y. have been responsible, under the auspices of the International Astronomical Union (IAU), for coordinating discussion on this topic, and for producing and maintaining the specification document for a format based on the Flexible Image Transport System (FITS).<sup>2</sup>

Preliminary drafts of the specification for such a format were discussed by the community at the 198th Meeting of the American Astronomical Society in June 2001, and the August 2001 Meeting of the IAU Working Group on Optical/IR Interferometry. The discussion continued by email amongst various interested parties, and the document was revised.

A public pre-release of the Format Specification and accompanying example C code was made in March 2002, followed by a second pre-release of the document in April 2002. This draft was discussed at the August 2002 IAU Working Group Meeting. Comments were presented by participants from the European Southern Observatory and the Interferometry Science Center (now the Michelson Science Center). Since the IAU Working Group meeting there were two further pre-releases of the Format Specification, prior to the first “production” release.

The standard was frozen on 7 April 2003 (release 5 of the Format Specification). The revision numbers of all tables are currently equal to one. Future changes to the Format Specification will require increments of the revision numbers for the changed tables.

Interferometer projects supporting the standard include COAST, NPOI, IOTA, VLTI, PTI, and the Keck Interferometer.

### 1.1. Nomenclature

We suggest that the common data format be referred to as the “OI Exchange Format”, or the “Exchange Format” when the context is clear. A suitable very short moniker is “OIFITS” (or “OI-FITS”). The document that formally describes the format is the “Format Specification”.

---

Further author information: (Send correspondence to J.S.Y.)

J.S.Y.: E-mail: [jsy1001@cam.ac.uk](mailto:jsy1001@cam.ac.uk), Telephone: 44 1223 337365

T.P.: E-mail: [pauls@nrl.navy.mil](mailto:pauls@nrl.navy.mil)

## 2. OUTLINE DESCRIPTION

The definitive description of the OI Exchange Format is contained in the formal Format Specification, available from <http://www.mrao.cam.ac.uk/~jsy1001/exchange/> (the latest version and all previous public releases are available from the website).

The outline description here should be read in conjunction with the Format Specification, and is intended to describe, in broad terms, the structure of an OIFITS file and the types of data that can be stored within it.

### 2.1. Purpose and Scope

By defining and maintaining a common data format, we hope to encourage the development of common software for analysis of data from optical stellar interferometers, and to facilitate the exchange of data between different research groups. In this way, the limited Fourier coverage of current instruments can be ameliorated by combining data from several interferometers. An example of this is given in the recent paper by Monnier et al.<sup>3</sup>

A first application of the Exchange Format, a “beauty contest” to compare the performance of a number of image reconstruction software packages, is described by Lawson et al. in these proceedings.<sup>4</sup>

The format is intended to support the storage of calibrated, time-averaged data. Such data can be prepared from the raw fringe measurements without using information about the detailed structure of the target (i.e. without doing any astrophysical interpretation), yet once the data is in the format, it can be analysed without knowing the details of the instrument. Calibrated data from different interferometers can be treated in the same way (provided there are no residual systematic errors).

The design of the FITS “meta-format” on which OIFITS is based has led us to adopt a structure analogous to a relational database. Different categories of information are stored in distinct “tables” (implemented as FITS binary tables<sup>2</sup>) within a file. The tables themselves contain the information needed to cross-reference from one table to another.

As currently defined, the standard includes the information needed to perform “imaging” from a calibrated data-set. Here imaging refers loosely to any process for making inferences about the sky brightness distribution from the data.

As it stands, the format is not suitable for astrometric applications, but other FITS tables could perhaps be added for this purpose.

The format defines an ancillary table (OLARRAY) giving the locations of the array elements, assuming that they are fixed on the Earth’s surface. However, instance(s) of this table may be omitted from an OIFITS file (the definitive  $uv$  coordinates are in other tables), and so the format can already be used for aperture masking or space (imaging) interferometry applications. We anticipate that equivalents to OLARRAY will be included in future releases of the standard.

A consequence of trying to define a minimal set of information that usefully describes any data-set is that the standard, by itself, is unsuitable as an archive format. However, the standard explicitly allows extra tables (not defined by the standard) or additional columns within defined tables to be included in an OIFITS file. In this way, instrument-specific ancillary data may be stored if so desired.

We encourage those who make use of additional tables or columns to bring to the attention of the community any that might be generally applicable. Such definitions can then be incorporated in future releases of the standard (probably as optional components).

### 2.2. Content of an OIFITS file

The content required for imaging includes the interferometric observables (complex visibility, squared visibility and triple product), which correspond to Fourier components of the sky brightness distribution, and their coordinates in the Fourier plane ( $uv$  coordinates).

The minimal set of ancillary information defined by the standard includes: Universal Time of measurements, spectral bandpasses, target identification and coordinates.

The standard also includes (for Earth-fixed array stations, for the moment) the information needed to reconstruct, and interpolate between, the definitive  $uv$  coordinates stored with the data.

The format can accommodate (mostly by allowing multiple tables of the same type) multiple targets, epochs of observation, wavebands and interferometer arrays within the same data file. This facilitates the merging of heterogeneous data-sets into a single file (e.g. to combine data from different facilities). The downside of this choice is that parsing an OIFITS file may not be straightforward - a particular application will often necessitate picking out a subset of the data, which may involve making more than one pass through the file to follow cross-references between tables.

### 2.3. Tables Defined by the Format

The tables defined by the OIFITS standard are all instances of FITS binary tables. In general, FITS binary tables (strictly binary table header-data units, or HDUs) consist of an ASCII header containing a number of “keywords” and associated values, followed by a binary data part containing tabular data. Mandatory header keywords describe the structure of the binary part, and further keywords may be used for application-specific purposes.

The tabular part of a binary table consists of a number of labelled columns of homogeneous data. A particular table cell can contain either a single scalar value or a vector of values of a specified data type (types used in OIFITS are 16-bit integers, ASCII characters, 32- and 64-bit floating point values, 128-bit complex numbers, and logical (boolean) values).

OIFITS defines a number of types of tables, specifying the header keywords and table columns that must be present for each table type. The data types that must be used are specified. For most of the table types, multiple instances may be present in the same OIFITS file. Additional columns, beyond those defined by the standard, may be present in any of the defined tables. Further tables, whose structure is not defined by the standard, may also be present.

An OIFITS file must contain:

- One **OL\_TARGET** table containing the coordinates of the observed targets
- *Optionally*, one or more **OL\_ARRAY** tables giving the coordinates of the interferometer array elements
- One or more of the data tables:
  - **OL\_VIS**: Complex Visibility
  - **OL\_VIS2**: Squared Visibility
  - **OL\_T3**: Triple Product (i.e. bispectrum/closure phase)
- Each data table must refer to an **OL\_WAVELENGTH** table that is present in the file. This describes the wavebands at which the data were obtained

Future releases of the standard may specify equivalents to OL\_ARRAY, for e.g. aperture masking experiments, or space interferometry. In the meantime the OL\_ARRAY table(s) should always be omitted if the apertures are not fixed on the Earth’s surface for the duration of the observation (the OL\_ARRAY(s) *may* be omitted even if the data are from a ground-based interferometer).

### 2.4. Cross-referencing

We mention above that each data table (OL\_VIS, OL\_VIS2, or OL\_T3) must refer to an OL\_WAVELENGTH table that is present in the file. Each data table has an “INSNAME” keyword whose value must match that for the INSNAME keyword in one (and only one) of the OL\_WAVELENGTH tables.

Each data table *may* contain a keyword ARRNAME - if present there must be an OL\_ARRAY table (or its equivalent for non-Earth-fixed stations) with the same ARRNAME. The STA\_INDEX values in the body of the

data table then refer to the rows with the same STA\_INDEX in the array table — these rows give the coordinates of the array stations contributing to each data point.

The TARGET\_ID column of each data table is an index into the single, compulsory, OI\_TARGET table.

All of these cross-references must be maintained when merging tables, or when copying tables between OIFITS files.

## 2.5. Data Representation

The formal definitions of the data quantities are given in Sect. A. Some explanatory notes are given here.

### 2.5.1. Field of view issues

There is currently no provision in the format for indicating either the field of view over which interference fringes are detected, or the field over which the total flux used to normalise the visibility is collected.

The format does not specify a  $w$  coordinate to go with  $u$  and  $v$ . All currently-operating stellar interferometers can only image fields smaller than the Airy disc of one of their unit telescopes (typically 0.5 arcsec or less) — with modest baselines  $w$  is not necessary as a “flat-sky” approximation (which leads to the Fourier transform relationship between visibility and sky brightness) is valid. Users of the format can add a  $w$  column where this is needed, while still adhering to the standard.

### 2.5.2. Complex visibility

A number of different classes of data can be represented as complex visibilities (Eq. (1)), including several varieties of differential phase data. In all cases the standard should only be used to store averaged data. Thus, as with triple products, we must consider the shape of the noise ellipse in the complex plane.

It has been demonstrated (see <http://www.mrao.cam.ac.uk/~jsy1001/exchange/complex/complex.html>) that both circularly-symmetric noise, and noise ellipses elongated parallel to or perpendicular to the mean vector can occur in practice. Thus far there has been no evidence for noise ellipses elongated parallel to the real or imaginary axes, although examples of some classes of data have yet to be presented.

Hence an amplitude/phase representation of complex visibilities, mirroring that used for triple products, has been adopted in the current version of the standard.

### 2.5.3. Squared visibility

This data type should be used for visibilities whose phases are not directly meaningful, the phase errors typically being due to atmospheric turbulence. The format uses squared visibility (Eq. (2)) rather than the modulus of the complex visibility, as unbiased estimators (typically based on the fringe power spectrum) exist for the squared quantity.

### 2.5.4. Triple product

The data are assumed to be complex triple products (Eq. (3)) averaged over a large number of “exposures”. In such a case, the noise can be fully described in terms of a Gaussian noise ellipse in the complex plane. Photon, detector and background noise tend to lead to noise ellipses that are close to circular. On the other hand, fluctuating atmospheric phase errors across telescope apertures typically cause fluctuations in the amplitude of the triple product which are much larger than the fluctuations in the phase. Thus the atmospheric contribution to the noise ellipse is elongated along the direction of the mean triple product vector in the Argand diagram. Such noise needs to be characterised in terms of the variance  $\sigma_{\perp}^2$  perpendicular to the mean triple product vector  $T$  and the variance  $\sigma_{\parallel}^2$  parallel to  $T$ . We can parameterize the perpendicular variance in terms of a “phase error”  $\sigma_{\theta} = (180/\pi)(\sigma_{\perp}/|T|)$ . The phase error gives an approximate value for the rms error in the closure phase in degrees. We denote  $\sigma_{\parallel}$  as the “amplitude error”.

In many cases, the observer may be interested primarily in the closure phase and not the triple product amplitude, and therefore may choose not to calibrate the amplitude (this is the case with COAST data). Such a case can be indicated in the above notation as an infinite amplitude error and a finite phase error. The data format specifies that such a case should be indicated by a NULL value for the amplitude (the amplitude error value is then ignored).

## 2.6. Shortcomings of the Format

We would like to highlight several features that are missing from the format. A number of these were mentioned as desirable when early drafts of the Format Specification were being discussed within the community. However, the authors took the decision to simplify the format as far as possible, consistent with its intended applications. This has hopefully encouraged take-up of OIFITS, and minimised the number of problems encountered in using the format.

The following simplifications and shortcomings remain in Release 5 of the Format Specification (the first “production release”):

- No equivalents to the OLARRAY table for aperture masking and space interferometry
- No instrument-specific information (configuration etc.)
- No astrometric information (internal metrology etc.)
- The spectral bandpass of the instrument is described by a single value giving the half-power bandwidth. Model-fitting software should probably assume a top-hat bandpass
- Complex visibility and Squared visibility cannot be specified in physical units (i.e. in terms of correlated flux)

The following items were in early drafts of the Format Specification, but were removed as they were deemed over-complicated and/or unnecessary at this stage:

- Polarization state to which the data correspond
- Description of *uv* plane averaging
- Calibrator codes (to specify which targets in a file are “calibrators”)

## 3. SOFTWARE

### 3.1. C Example Code

Example code written in ANSI C is available from the OIFITS website (<http://www.mrao.cam.ac.uk/~jsy1001/exchange/>). We expect this code will be updated whenever the Format Specification is revised.

The example code demonstrates how to use calls to the CFITSIO library<sup>5</sup> to read and write the constituent tables of the format. Data is transferred to or from a set of purpose-designed C structures. It is possible to use the supplied routines without modification, by accessing instances of these structures as appropriate. More often though, the routines will be useful as starting points for users to develop their own code for reading and writing the format.

### 3.2. IDL Routines

An IDL library designed to read and write optical interferometry data conforming to the OIFITS standard has been written by J.D.M., and is available from [http://www.astro.lsa.umich.edu/~monnier/oi\\_data/](http://www.astro.lsa.umich.edu/~monnier/oi_data/).

All features of the OIFITS format are supported in v1.2, including the ability to read and write multiple OLARRAY, OL\_WAVELENGTH, OL\_VIS, OL\_VIS2, and OL\_T3 binary tables. Merging data from different epochs and arrays is very straightforward, using the supplied `merge_oidata.pro`.

Each of the six allowed tables, OLARRAY, OL\_TARGET, OL\_WAVELENGTH, OL\_VIS, OL\_VIS2, and OL\_T3, is converted from FITS into an IDL structure. While values are stored directly in the structure when possible, pointers must be used in order to accommodate the full standard which allows multiple OL\_WAVELENGTH tables.

### 3.2.1. Example

In order to read in the tables from the `testdata.fits` file which is distributed by J.S.Y. (<http://www.mrao.cam.ac.uk/~jsy1001/exchange/>) one merely executes the following IDL command:

```
IDL> READ_OIDATA, 'testdata.fits', oiarray,oitarget,oiwavelength,$
      oivis, oivis2,oit3, /inventory
      This file Satisfies the requirements of the OI_DATA format
      Inventory:
      OI_ARRAY:          1
      OI_TARGET:        1
      OI_WAVELENGTH:    1
      OI_VIS:           1
      OI_VIS2:          1
      OI_T3:            1
      Unknown Tables:   0
IDL> print,oivis2(0).time,*oivis2(0).vis2data,*oivis2(0).vis2err
      82810.000      0.67700000      0.064000000
IDL> print,string(*oivis(0).flag)
      F
```

Writing the data back is just as easy:

```
IDL> WRITE_OIDATA, 'testdata1.fits', oiarray,oitarget,oiwavelength,$
      oivis, oivis2,oit3
```

### 3.3. Python Library Module

A library coded in the Python language (see <http://www.python.org>) for reading and writing the Exchange Format is available from the OIFITS website. This makes use of the PyFITS library<sup>6</sup> from Space Telescope Science Institute (see [http://www.stsci.edu/resources/software\\_hardware/pyfits](http://www.stsci.edu/resources/software_hardware/pyfits)).

The python library module is structured as a set of python classes, instances of which provide object-oriented access to data from an OIFITS file (which may be created in memory before the corresponding file exists on disc). The classes are as follows:

**OIFITS** Instance stores data for entire OIFITS file

The data attributes of the OIFITS instance are instances of other classes, or lists/dictionaries of these. Each of these “table classes” corresponds to one of the FITS binary tables defined in the standard:

**Array** OI\_ARRAY table

**Target** OI\_TARGET table

**Wavelength** OI\_WAVELENGTH table

**Vis** OI\_VIS table

**Vis2** OI\_VIS2 table

**T3** OI\_T3 table

The table classes typically have data attributes that are instances of lower-level classes, corresponding to rows of the FITS table. These “row classes” generally don’t have any method attributes.

The entire FITS file corresponding to an OIFITS instance may be read from/written to disk using the instance’s `FromHDUList()` and `ToHDUList()` methods. These convert from/to a `pyfits.HDUList` instance (which represents data from a generic FITS file), as defined in the third-party `pyfits` module. `Pyfits` provides many functions that can operate on `HDUList` instances (see the `pyfits` manual).

A script to write an OIFITS file would be structured as follows:

```
import oi_fits
oi = oi_fits.OI_FITS()
# insert code to assign to data attributes
oi.ToHDUList().writeto('output.fits')
```

A script to read from an OIFITS file would be structured like this:

```
import oi_fits, pyfits
hlist = pyfits.open('input.fits')
oi = oi_fits.OI_FITS()
oi.FromHDUList(hlist)
hlist.close()
# insert code to do something with oi's data attributes
```

Instances of table classes have `FromTable()` and `ToTable()` methods, which convert between the table class instance and a `pyfits.BinTableHDU` instance. These methods may be used when reading/writing individual tables, rather than the whole file.

A python script for copying selected tables from one OIFITS file to another, `oicopy`, is available from the OIFITS website.

### 3.4. Software Applications that read OIFITS

In this section, we list some freely-available (or soon-to-be so) software applications that support the OIFITS format.

#### 3.4.1. Model-fitting

Full descriptions of the packages are available on their own websites:

- `mfit` - <http://www.mrao.cam.ac.uk/~jsy1001/mfit/>
- `OYSTER` - <http://www.sc.eso.org/~chummel/oyster/oyster.html>

### 3.5. Image reconstruction

None of the imaging packages entered into the Beauty Contest<sup>4</sup> that support OIFITS directly are currently available to third parties. Interested parties may wish to contact the authors of the packages directly.

## 4. FUTURE ENHANCEMENTS TO THE FORMAT

We hope that the format will continue to develop, in order to better serve the needs of the community. Discussion of future enhancements should be addressed to the OLBIN email list (see <http://listes.obs.ujf-grenoble.fr/wws/info/olbin> for information on how to subscribe and post to the list). The earlier oi-data mailing list is defunct.

Rather than further development being driven by the authors (whose ideas of what would be most useful may not be widely shared!), we would like to see it being driven by the community. As mentioned above, we suggest that people publicise any extensions to the standard that they use in their own applications. The maintainers of the standard can then decide whether the new features are more generally applicable. If so, they can be incorporated in future releases of the Exchange Format, probably as optional features.

Any enhancements to the format that are not backwards-compatible will be accompanied by increments in the OLREVN version numbers of the altered tables. We will aim to avoid backwards-incompatible changes wherever possible, but we encourage users to allow for the possibility of future changes when designing software to use the format.

## ACKNOWLEDGMENTS

Development of the format was performed under the auspices of the IAU Working Group on Optical/IR Interferometry (see <http://olbin.jpl.nasa.gov/iau/>), and has the strong support of its members. We would like to thank Peter R. Lawson (Chair of the Working Group) for his encouragement and support.

The authors would like to thank David Buscher, Christian Hummel, and David Mozurkewich for providing material for the Format Specification and the OIFITS website. We would like to take this opportunity to thank all those in the community who have contributed to the definition of the format and to its take-up by many interferometer projects.

## APPENDIX A. DEFINITIONS

In the context of the OI Exchange Format, the quantities referred to in the main body of the paper are defined as follows. These definitions are taken from the Format Specification, which remains the definitive reference for OIFITS.

### Baseline Vector

The baseline vector between two interferometric stations A and B whose position vectors are  $\mathbf{x}_A$  and  $\mathbf{x}_B$  is defined as  $\vec{AB} = \mathbf{x}_B - \mathbf{x}_A$ .

### UV Coordinates

$u$  is the East component and  $v$  is the North component of the projection of the baseline vector onto the plane normal to the direction of the phase center, which is assumed to be the pointing center.

### Complex Visibility

The basic observable of an interferometer is the complex visibility, which is related to the sky brightness distribution  $I(x, y)$  by a Fourier Transform:

$$V(u, v) = \iint dx dy I(x, y) \exp(-2\pi i(ux + vy)). \quad (1)$$

$x$  and  $y$  are displacements (in radians) in the plane of the sky (which is assumed to be flat). The origin of this coordinate system is the phase center.  $x$  is in the direction of increasing right ascension (i.e. the  $x$  axis points East), and  $y$  is in the direction of increasing declination (the  $y$  axis points North). With  $x$  and  $y$  defined, the above equation defines the sign convention for complex visibilities that should be used in the data format. The visibility is normalized by the total flux from the target, which is assumed to remain constant over the time spanned by the measurements in the file. Neither the field of view over which the total flux is collected, or the field of view over which fringes are detected (i.e. the limits of the above integral), can be inferred from the data file.

## Squared Visibility

The squared visibility is simply the modulus squared of the complex visibility:

$$S(u, v) = |V(u, v)|^2. \quad (2)$$

## Triple Product

The triple product, strictly the bispectrum, is the product of the complex visibilities on the baselines forming a closed loop joining three stations A, B and C. In the following expression,  $(u_1, v_1)$  is the projection of AB and  $(u_2, v_2)$  is the projection of BC (and hence  $(u_1 + u_2, v_1 + v_2)$  is the projection of AC):

$$T(u_1, v_1, u_2, v_2) = V(u_1, v_1) V(u_2, v_2) V^*(u_1 + u_2, v_1 + v_2). \quad (3)$$

The argument of the triple product is the closure phase, and its modulus is the triple product amplitude (not to be confused with the “closure amplitude”).

## REFERENCES

1. H. McAlister and T. Cornwell, eds., *Report on the Workshop on Imaging with Ground-based optical interferometers*, National Science Foundation, 2000. <http://olbin.jpl.nasa.gov/papers/Report1.0.PDF>.
2. R. J. Hanisch, A. Farris, E. W. Greisen, W. D. Pence, B. M. Schlesinger, P. J. Teuben, R. W. Thompson, and A. Warnock, “Definition of the Flexible Image Transport System (FITS),” *Astronomy & Astrophysics* **376**, pp. 359–380, 2001.
3. J. Monnier, R. Millan-Gabet, P. Tuthill, W. Traub, N. Carleton, V. C. du Foresto, W. Danchi, M. Lacasse, S. Morel, G. Perrin, I. Porro, F. Schloerb, and C. Townes., “High-resolution imaging of dust shells using Keck aperture masking and the IOTA interferometer,” *Astrophys. J.* **605**, pp. 436–461, 2004.
4. P. R. Lawson, W. D. Cotton, C. A. Hummel, J. D. Monnier, M. Zhao, J. S. Young, H. Thorsteinnsson, S. C. Meimon, L. Mugnier, G. L. Besnerais, E. Thiébaud, and P. G. Tuthill, “An interferometry imaging beauty contest,” in *New Frontiers in Stellar Interferometry*, W. Traub, J. D. Monnier, and M. Schöller, eds., *Proc. SPIE* **5491**, SPIE Press, Bellingham, WA, 2005. Paper 98, these proceedings.
5. W. Pence, “CFITSIO, v2.0: A New Full-Featured Data Interface,” in *ASP Conf. Ser. 172: Astronomical Data Analysis Software and Systems VIII*, p. 487, 1999.
6. P. E. Barrett and W. T. Bridgman, “PyFITS, a Python FITS Module,” in *ASP Conf. Ser. 216: Astronomical Data Analysis Software and Systems IX*, p. 67, 2000.