On Representing Celestial Images Using FITS Tables

W. Pence, A. Rots, L. Angelini

NASA Goddard Space Flight Center
Greenbelt, MD 20771

Date: 19 Oct 1995

1 Introduction

There are three commonly used methods of representing an astronomical celestial image in a FITS format file:

- as a 2-dimensional array of pixels in the FITS primary array or in an IMAGE extension,
- as a 2-dimensional vector within a single element of a FITS binary table, or
- as a tabulated list of pixel coordinates (and other optional parameters) in a FITS ASCII or binary table.

The first method is the most commonly used, but the other 2 ways of representing an image are useful in certain applications. In high energy astrophysics, for example, the images are often very sparse, so it is more efficient to simple list the pixel location in the image of all the detected photons rather than store the full 2-D image.

In addition to storing the image data values themselves in the FITS file, one also needs to record an associated set of parameters which describe how the image data are projected onto the celestial sphere. The paper entitled "Representations of Celestial Coordinates in FITS" (here after referred to as RCCF) by E. W. Greisen and M. Calabretta dated October 5, 1995, defines a convention for a set of FITS keywords that describe the celestial coordinate system for the first type of image representation, namely, for a FITS primary array or IMAGE extension. The purpose of this present paper is to define a convention for the analogous set of keywords that should be used when the image is represented in one of the other 2 methods.

2 Types of image representations

This section describes in more detail the three basic ways in which a celestial image may be represented within a FITS file.

2.1 Primary array image or IMAGE extension

The most common way of representing an image in a FITS file is as a 2-dimensional array of pixels stored either in the primary array or in an IMAGE extension. Readers should refer to the RCCF paper which defines all the necessary keywords needed to describe the celestial coordinate system in this case.

2.2 2-dimensional vector in a binary table

One important feature of the FITS binary table (BINTABLE) format is that each element of the table (i.e., the intersection of a row and a column in the table) is not limited to a scalar value and may instead contain a vector of data values. Consequently, one can construct a binary table in which each row contains one or more columns containing images. In principle there is no limit to the number of images that can be stored in this way in a single table. Other columns in the table can be used to store other information associated with each image, such as the date of the exposure or the exposure duration. This format is particularly useful for compactly storing a set of logically related images in a single file, rather than storing each image in a separate FITS file. Note that the `TDIM`n keyword (where n is an integer column number) is used in this case to specify the number and size of the dimensions in the vector column. Usually, all the images in a given column of the table have the same dimensionality, but one can use the variable-length array feature in binary tables to store different-sized images in the same column.

2.3 Tabulated list of pixels

The third way to represent an image in a FITS file is as a list of pixel coordinates and other optional associated parameters in a binary or ASCII table extension. This representation is often used in high-energy astrophysics as a way of tagging the position and other properties of individual detected photons (similar to the QPOE format used in IRAF system). This is a compact way of representing sparse images since only the non-zero pixels need to be given in the list. At a minimum, this format requires that the FITS table have 2 columns which give the X and Y pixel coordinate of the corresponding event in the virtual 2-D image; any number of other columns may be included in the table to store other parameters describing the event, such as arrival time, or photon energy. One can of course convert this virtual image representation into a real image by computing the 2-dimensional histogram of the X and Y pixel columns.

3 The reserved FITS coordinate system keywords

A standard set of keywords is needed to describe the celestial coordinate system for each of the 3 FITS image representations discussed above. The RCCF paper has already defined the keywords to be used with images stored in a FITS primary array or image extension, and we repeat them here without any modification. Since it is not possible to use the exact same keywords in the other 2 cases (for one reason, the keyword names must all have an index number pointing to the column to which they apply), we have defined an analogous set of keywords to be used in these cases, as shown in the following table.

Keyword Description	Primary Array	BINTABLE Vector	Table Pixel List
Axis Type	CTYPE*i*	*i*CTYP*n*	TCTYP*n*
Axis Units	CUNIT*i*	*i*CUNI*n*	TCUNI*n*
Reference Pixel	CRPIX*i*	*i*CRPX*n*	TCRPX*n*
Reference Value	CRVAL*i*	*i*CRVL*n*	TCRVL*n*
Pixel Increment	CDELT*i*	*i*CDLT*n*	TCDLT*n*
Axis Rotation	CROTA*i*	*i*CROT*n*	TCROT*n*
Rotation Matrix	PC*iiijjj*	*ij*PC*n*	TPC*nnmm*
Coord. Rotation	LONGPOLE	LONGPOLE or LONGP*n*	LONGPOLE or LONGP*n*
Coord. Rotation	LATPOLE	LATPOLE or LATP*n*	LATPOLE or LATP*n*
Proj. Param. 1	PROJP1	PROJP1 or P1PRO*n*	PROJP1 or P1PRO*n*
Proj. Param. 2	PROJP2	PROJP2 or P2PRO*n*	PROJP2 or P2PRO*n*
Coord. Epoch	EQUINOX	EQUINOX or EQUIN*n*	EQUINOX or EQUIN*n*
MJD of Obs.	MJD-OBS	MJD-OBS or MJDOB*n*	MJD-OBS or MJDOB*n*
Reference Frame	RADECSYS	RADECSYS or RADEC*n*	RADECSYS or RADEC*n*

Notes:

– *i, j* are integer axis numbers referring to an axis of the array;
– when used as a keyword prefix these are limited to 1 digit.
– *iii, jjj* are 3–digit integer axis numbers, including leading zeros (001–999).
– *n* is an integer table column number (1–999) without any leading zeros.
– *mm, nn* are 2–digit integer table column numbers (01–99).

The following paragraphs briefly describe each type of keyword.

1. Axis Type: a character string label which defines the type of axis. In the case of the celestial coordinate system projections discussed in RCCF, this keyword also specifies the type of map projection and implicitly defines the units of the physical coordinates (see also the Axis Units keyword below).

2. Axis Units: specifies the physical units of each axis. For celestial coordinates, angles must always be given in units of degrees (i.e., 'deg').

3. Reference Pixel: the pixel coordinate of the reference point. When dealing with arrays, the first pixel runs from coordinate number 0.5 to 1.5 on each axis and the coordinate (1.0, 1.0) refers to the center of the lower left pixel in a 2-D image.

4. Reference Value: the physical coordinate value at the position of the reference pixel. For celestial coordinates this is always given in units of degrees.

5. Pixel Increment: the increment per pixel of the physical coordinate value at the position of the reference pixel. For celestial coordinates this is always given in units of degrees.

6. Axis Rotation: the rotation angle between the pixel axis and the physical coordinate axis in degrees. This keyword is superseded by the Rotation Matrix keywords, but the `CROTA`i keyword should still be written to assist older FITS–reading programs. This keyword is only appropriate for 2–dimensional arrays and should only be given for the 2nd axis (i.e., `CROTA2`).

7. Rotation Matrix: the rotation (and skew) terms needed to convert from the pixel coordinate system to the physical coordinate system. After this transformation has been applied the coordinates are aligned with the physical axes but are still in pixel units. The Pixel Increment value is then used to scale the pixel values to the physical values. Note that in the table pixel list image format the `TPC`$nnmm$ keyword is limited to 2–digit column numbers, thus one is restricted to using columns 1 – 99 for storing the X and Y pixel coordinates.

8. Coordinate Rotation: the longitude and latitude in the native coordinate system of the standard system's north pole. These keywords are only applicable to certain celestial coordinate system projections (see RCCF). If different columns of the table contain images with different rotations, then the `LONGP`n and `LATP`n form of the keyword must be used to give the rotation for each column individually.

9. Projection Parameters: some projections require additional parameters. The interpretation of these keywords depends on the value of the Axis Type keyword. If different columns of the table contain images with different projection parameters, then the `P1PRO`n and `P2PRO`n form of the keywords must be used to give the parameter for each column individually.

10. Coordinate Epoch: epoch of the mean equator and equinox in years. It is recommended that the value of this keyword be either 1950.0 or 2000.0. If different columns of the table contain images with different epochs, then the `EQUIN`n form of the keyword must be used to give the epoch value for each column individually.

11. MJD of Observation: the modified Julian date of the observation in decimal days. If different columns of the table contain images with different dates, then the `MJDOB`n form of the keyword must be used to give the date for each column individually.

12. Reference Frame: the frame of reference of the equatorial coordinates (e.g., 'FK4' for EQUINOX 1950.0 and 'FK5' for EQUINOX 2000.0). If different columns of the table contain images with different reference frames, then the `RADEC`n form of the keyword must be used to give the frame for each column individually.

4 Multiple images and the 'Greenbank Convention'

In the general case of the binary table vector representation of images, the images contained in each row of the table will not necessarily have the same coordinate transformation keywords values. If, for example, the value of the reference pixel is different for each image/row then a single 1CRVLn keyword in the header is not sufficient to record all the individual values. Instead, this keyword should be replaced by a column with the same name (i.e., TTYPEn = '1CRVLn') which is used to store the reference value appropriate for each row of the table. This convention for expanding a keyword into a table column (or conversely, collapsing a column of identical values into a single header keyword) was originally developed and promoted at the Greenbank Radio Observatory, and hence this is commonly referred to as part of the Greenbank Convention for keywords.

5 Example FITS Headers

The following examples show how the same image may be represented using different FITS formats. In each case the images are 300 x 200 pixels in size and are rotated 30 degrees with respect to the RA and DEC celestial coordinate system.

5.1 Primary Array

This is a sample FITS header for an image stored in a FITS primary array.

```
SIMPLE  =                    T / file does conform to FITS standard
BITPIX  =                   32 / number of bits per data pixel
NAXIS   =                    2 / number of data axes
NAXIS1  =                  300 / length of data axis   1
NAXIS2  =                  200 / length of data axis   2

CTYPE1  = 'RA---TAN'           / TAN projection used
CRPIX1  =                  150 / reference pixel
CRVAL1  =                45.83 / RA at the reference pixel
CDELT1  =          -.00277777 / increment per pixel (degrees)
CUNIT1  = 'deg     '           / physical units of axis 1

CTYPE2  = 'DEC--TAN'           / TAN projection used
CRPIX2  =                  100 / reference pixel
CRVAL2  =                63.57 / DEC at the reference pixel
CDELT2  =           .00277777 / increment per pixel (degrees)
CUNIT2  = 'deg     '           / physical units of axis 2
CROTA2  =                 30.0 / image rotation (degrees)

PC001001=          .866025403 / Coord. Descrp. Matrix:  cos(CROTA2)
PC002002=                  -.5 / Coord. Descrp. Matrix: -sin(CROTA2)
```

```
PC001002=                   .5 / Coord. Descrp. Matrix:  sin(CROTA2)
PC002001=           .866025403 / Coord. Descrp. Matrix:  cos(CROTA2)

EQUINOX =               2000.0 / coordinate epoch
MJD-OBS =        44258.7845612 / MJD of the observation
END
```

5.2 BINTABLE vector with fixed transformation parameters

In this example the image is stored as a vector in column 2 of the table where each row of the table contains a different 300 x 200 pixel image of the same object (taken at different times). All the coordinate system parameters are the same for each image and are given as header keywords, except that the `MJD-OBS` column gives the date corresponding to the image in that row.

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =               240008 / width of table in bytes
NAXIS2  =                    4 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)

TFIELDS =                    2 / number of fields in each row
TTYPE1  = 'MJD-OBS '           / date of observation
TFORM1  = '1D      '           / data format of the field: double
TTYPE2  = 'Image   '           / 2-D image vector
TFORM2  = '60000J  '           / data format of the field: I*4 vector
TDIM2   = '(300,200)'          / dimension sizes of the vector

COMMENT  The following keywords define the coordinate system of the image
COMMENT  contained in Column 2 of the table

1CTYP2  = 'RA---TAN'           / TAN projection used in axis 1
1CRPX2  =                  150 / reference pixel in axis 1
1CRVL2  =                45.83 / RA at the reference pixel
1CDLT2  =           -.00277777 / increment per pixel
1CUNI2  = 'deg     '           / physical units of axis 1

2CTYP2  = 'DEC--TAN'           / TAN projection used
2CRPX2  =                  100 / reference pixel
2CRVL2  =                63.57 / DEC at the reference pixel
2CDLT2  =            .00277777 / increment per pixel
2CUNI2  = 'deg     '           / physical units of axis 2
```

```
2CROT2  =                 30.0 / image rotation (degrees)

11PC2   =           .866025403 / Coord. Descrp. Matrix:  cos(CROTA2)
22PC2   =                  -.5 / Coord. Descrp. Matrix: -sin(CROTA2)
12PC2   =                   .5 / Coord. Descrp. Matrix:  sin(CROTA2)
21PC2   =           .866025403 / Coord. Descrp. Matrix:  cos(CROTA2)

EQUINOX =               2000.0 / coordinate epoch
END
```

5.3 BINTABLE vector with variable transformation parameters

This table, like the previous example, has 4 rows, each of which contains a 300 x 200 pixel image. In this case, however, the position of the image on the sky varies from image to image and so the Reference Values must be supplied in columns in the table rather than as header keywords. The pixel size and image rotation are still assumed to be the same for each image.

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =               240024 / width of table in bytes
NAXIS2  =                    4 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)

TFIELDS =                    4 / number of fields in each row
TTYPE1  = 'Name    '           / name of observed object
TFORM1  = '16A     '           / data format of the field: ASCII Character
TTYPE2  = '1CRVL4  '           / reference pixel of axis 1 in column 5
TFORM2  = '1E      '           / data format of the field: R*4
TTYPE3  = '2CRVL4  '           / reference pixel of axis 2 in column 5
TFORM3  = '1E      '           / data format of the field: R*4
TTYPE4  = 'Image   '           / 2-D image vector
TFORM4  = '60000J  '           / data format of the field: I*4 vector
TDIM4   = '(300,200)'          / dimension sizes of the vector

COMMENT  The following keywords define the coordinate system of the image
COMMENT  contained in Column 5 of the table

1CTYP4  = 'RA---TAN'           / TAN projection used
1CRPX4  =                  150 / reference pixel
1CDLT4  =           -.00277777 / increment per pixel
1CUNI4  = 'deg     '           / physical units of axis 1
```

```
2CTYP4  = 'DEC--TAN'           / TAN projection used
2CRPX4  =                  100 / reference pixel
2CDLT4  =            .00277777 / increment per pixel
2CUNI4  = 'deg     '           / physical units of axis 2
2CROT4  =                 30.0 / image rotation (degrees)

11PC4   =           .866025403 / Coord. Descrp. Matrix:  cos(CROTA2)
22PC4   =                  -.5 / Coord. Descrp. Matrix: -sin(CROTA2)
12PC4   =                   .5 / Coord. Descrp. Matrix:  sin(CROTA2)
21PC4   =           .866025403 / Coord. Descrp. Matrix:  cos(CROTA2)

EQUINOX =               2000.0 / coordinate epoch
MJD-OBS =        44258.7845612 / MJD of the observation
END
```

5.4 Pixel list

In this example the image is given as a list of the energy and pixel coordinates of a set of detected photons. A 2-D image array could be generated from this list by making a 2-D histogram of the number of photons that were recorded in each pixel. The TLMINn and TLMAXn keywords shown below are not required, but are useful for recording the legal range of the values in each column and are used when constructing histograms of the values.

```
XTENSION= 'BINTABLE'           / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                    8 / width of table in bytes
NAXIS2  =                10000 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)

TFIELDS =                    3 / number of fields in each row
TTYPE1  = 'ENERGY  '           / energy of the photon
TFORM1  = '1E      '           / data format of the field: R*8
TTYPE2  = 'X       '           / X pixel coordinate of the photon
TFORM2  = '1I      '           / data format of the field: ASCII Character
TTYPE3  = 'Y       '           / Y pixel coordinate of the photon
TFORM3  = '1I      '           / data format of the field: ASCII Character

TCTYP2  = 'RA---TAN'           / TAN projection used
TCRPX2  =                  150 / reference pixel
TCRVL2  =                45.83 / reference pixel
```

```
TCDLT2  =           -.00277777 / increment per pixel
TCUNI2  = 'deg     '           / physical units of axis 1
TLMIN2  =                    1 / lower limit of axis 1
TLMAX2  =                  300 / upper limit of axis 1

TCTYP3  = 'DEC--TAN'           / TAN projection used
TCRPX3  =                  100 / reference pixel
TCRVL3  =                63.57 / reference pixel
TCDLT3  =            .00277777 / increment per pixel
TCUNI3  = 'deg     '           / physical units of axis 2
TCROT3  =                 30.0 / image rotation (degrees)
TLMIN3  =                    1 / lower limit of axis 2
TLMAX3  =                  200 / upper limit of axis 2

TPC0202 =           .866025403 / Coord. Descrp. Matrix:  cos(TCROT2)
TPC0303 =                  -.5 / Coord. Descrp. Matrix: -sin(TCROT2)
TPC0203 =                   .5 / Coord. Descrp. Matrix:  sin(TCROT2)
TPC0302 =           .866025403 / Coord. Descrp. Matrix:  cos(TCROT2)

EQUINOX =               2000.0 / coordinate epoch
MJD-OBS =        44258.7845612 / MJD of the observation
END
```