

Improving Position Accuracy in Archived HST Images

Nadezhda Dencheva, Warren Hack, Michael Droetboom, Andrew Fruchter, Perry Greenfield

Space Telescope Science Institute

A convention for storing distortion information in HST images was developed and implemented in two software packages - PyWCS and STWCS. These changes allow the development of a WCS based version of Multidrizzle and image alignment software. The distribution of WCS solutions is discussed.

Distortion Representation

Mathematically the transformation from detector to sky for an ACS/WFC image can be approximated by:

$$\begin{pmatrix} \alpha \\ \delta \end{pmatrix} = \begin{pmatrix} CRVAL1 \\ CRVAL2 \end{pmatrix} + \begin{pmatrix} CD11 & CD12 \\ CD21 & CD22 \end{pmatrix} \begin{pmatrix} u' + f(u', v') + LT_x(x', y') \\ v' + g(u', v') + LT_y(x', y') \end{pmatrix}$$

Figure 1 shows the coordinate transformation pipeline, where $(x', y') = \text{DET2IM}(x, y)$ is a small correction for a detector defect expressed as a periodic change in the pixel width. It is represented by an extension with EXTNAME = D2IMARR following WCS Paper IV Lookup Table description. The (x', y') coordinates are the input to all distortion corrections.

The polynomial distortion $f(u', v')$ and $g(u', v')$ is stored in the header using the Simple Imaging Polynomial (SIP) convention:

$$f(u', v') = \sum_{p+q=2}^{AORDER} A_{p,q} u'^p v'^q, g(u', v') = \sum_{p+q=2}^{BORDER} B_{p,q} u'^p v'^q$$

$$(u', v') = (x' - CRPIX1, y' - CRPIX2)$$

The non-polynomial distortion LT_x , LT_y , is stored in the science files using Paper IV Lookup Table convention. For each X and Y axis a separate extension with an EXTNAME value of WCSDVARR is attached to the science file.

The linear WCS, SIP coefficients and non-polynomial distortion extensions include corrections for the time dependent component of the distortion and the velocity aberration of the observation. The new file structure is shown below.

EXT#	FITSNAME	FILENAME	EXTVE	DIMENS	BITPIX
0	j9irwb1q_fit	j9irwb1q_fit.fits			16
1	IMAGE	SCI	1	4096x2048	-32
2	IMAGE	ERR	1	4096x2048	-32
3	IMAGE	DQ	1	4096x2048	16
4	IMAGE	SCI	2	4096x2048	-32
5	IMAGE	ERR	2	4096x2048	-32
6	IMAGE	DQ	2	4096x2048	16
7	IMAGE	D2IMARR	1	4096	-32
8	IMAGE	WCSDVARR	1	64x32	-32
9	IMAGE	WCSDVARR	2	64x32	-32
10	IMAGE	WCSDVARR	3	64x32	-32
11	IMAGE	WCSDVARR	4	64x32	-32

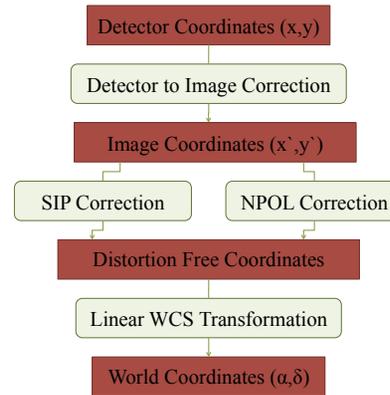


Figure 1: Coordinate transformation pipeline

Work in Progress – Distributing WCS Solutions

➤ An optional method allowing to keep track of all updates through a binary table with EXTNAME=WCS CORR is being developed.

➤ Aligning an image with a catalog typically only requires small corrections to the WCS, which can be stored as alternate WCS's in the headers. By definition these share the same distortion model.

➤ In cases when a WCS registration solution was generated for a specific astrometric solution a mechanism for passing all components of the WCS is needed. We are considering using FITS files, tentatively called 'headerlets', based on the distortion and WCS representation described above for this purpose. A headerlet file will look like this:

EXT#	FITSNAME	FILENAME	EXTVE	DIMENS	BITPIX
0	j9irw4b1q_hdr				16
1	IMAGE	SIPWCS	1		16
2	IMAGE	WCSDVARR	1	65x33	-32
3	IMAGE	WCSDVARR	2	65x33	-32
4	IMAGE	SIPWCS	2		16
5	IMAGE	WCSDVARR	3	65x33	-32
6	IMAGE	WCSDVARR	4	65x33	-32
7	IMAGE	D2IMARR	1	4096	-32

Software Implementation

➤ **PyWCS** – general WCS library

- a Python wrapper around Mark Calabretta's WCSLIB library. As such it provides access to WCSLIB's projections and image and spectra coordinate transformations.

- Implements Simple Imaging Polynomial (SIP) convention.

- Implements Paper IV Lookup Table distortion convention.

status: current release v. 1.9-4.4.4

➤ **STWCS** - HST specific WCS package

- Run in the HST pipeline to compute the WCS of an image based on the observation specific distortion.

- Defines an HSTWCS object allowing detector to sky transformations without image resampling.

- Given a list of HSTWCS objects computes a common frame on the sky and its WCS based on their distortion corrected footprints.

- Provides support for managing alternate WCS's in headers as described in WCS Paper 1.

status: current release v. 0.5

➤ **BetaDrizzle** – WCS based implementation of Multidrizzle

- Functionality split in two – coordinate transformation part based on PyWCS and STWCS and a resampling part based on the drizzle algorithm.

- Implemented in Python and C.

status: in testing, released at STScI

➤ **TweakReg** – tentative name for image registration software

- An extended version of the current tweakshifts script, implemented in C and Python.

- Updates the headers directly with the determined WCS correction necessary to align the the image with the reference image used for the fit without the need for image resampling (drizzling).

status: in development, basic implementation complete

Summary

The changes presented here aim at increasing the astrometric accuracy of HST images. Once implemented in the HST pipeline and archive, users will be able to retrieve calibrated images and align them to astrometric references without the need to download additional reference files. Distributing WCS solutions should dramatically simplify the process of generating photometrically uniform, distortion-free mosaics from HST data regardless of when the data was taken.