*The 2010 STScI Calibration Workshop*
*Space Telescope Science Institute, 2010*
*Susana Deustua and Cristina Oliveira, eds.*

# BetaDrizzle: A Redesign of the MultiDrizzle Package

A. S. Fruchter, W. Hack, N. Dencheva, M. Droettboom, P. Greenfield

*Space Telescope Science Institute, Baltimore, MD 21218*

**Abstract.** We describe substantial changes to MultiDrizzle intended to allow users to easily and accurately align and combine HST images taken at multiple epochs, and even with different instruments. In the first part of this program, the correction of the time-dependent distortion of ACS was introduced into MultiDrizzle to provide distortion corrections good to a few hundredths of a pixel level. Now, we are undertaking a more profound change in the underlying method that MultiDrizzle (and HST) use to represent astrometry and geometric distortion. As part of this change we have temporarily renamed MultiDrizzle as BetaDrizzle to avoid confusion between the programs. BetaDrizzle uses new simple extensions of the fits format that allow us to fully represent the ACS and WFC3 distortions in the header of the image, meaning that a calibrated image needs no other files to describe its astrometry. Users will also be able to easily pass astrometric header information to each other or back to the archive for use by others. The presence of the full astrometric solution in the header means that users can fit the astrometry of one image to another image, or to an external catalog, without drizzling the image. Precise coordinate information can be extracted directly from the .flt image itself.

## 1. Introduction

When we started this program we had one guiding principle: Astrometry should be like any other calibration; once an image is calibrated you should not need to carry around separate calibration files. All of the necessary information should be in the image and its headers. This lead us to several more specific goals.

- The best available geometric distortion information should be incorporated into HST image headers.
- A user should be able to easily align an image and update its WCS to match another image or catalog
- Users should be able to easily exchange astrometric solutions with each other and/or the archive

To accomplish these goals we have extended the FITS standard and have created several new pieces of software to extend the capabilities and accuracy of MultiDrizzle (Koekemoer et al. 2002). The FITS extensions allow users to incorporate (compressed) versions of the astrometric information in their headers and to keep several different WCS solutions in an image at the same time. The upgraded MultiDrizzle software set, which we call BetaDrizzle for now to avoid user confusion, provides a means for users to control this WCS information, to update it as needed, and to transfer astrometric solutions to each other or the archive.

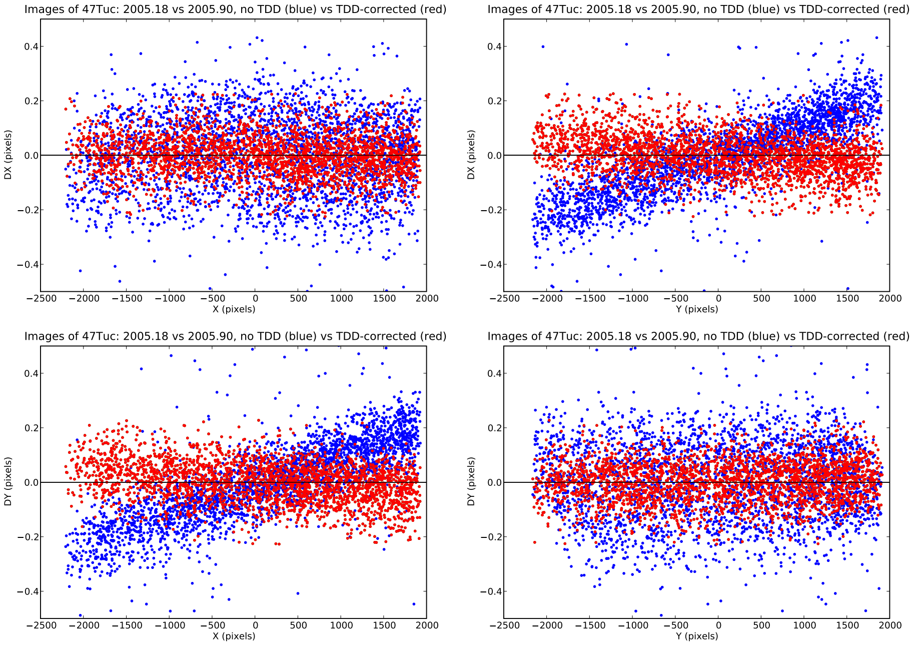


Figure 1: A comparison of distortion error with and without time-dependent distortion (TDD) correction. The stellar positions from two ACS/WFC images of 47 Tuc are compared. Both images were taken in 2005, but at a relative rotation of about 90 degrees. The blue points represent offsets between the same stars in the two different images after stellar positions have been fit for a possible rotation, shift and scale change, but without the application of the time-dependent distortion correction. The red points show the result of the same fit after time-dependent distortion correction is applied. While small systematic errors remain (up to perhaps 0.05 pixels), the peak-to-peak distortion error has been reduced nearly an order of magnitude. The TDD correction is added to the image by modifying the World Coordinate System maxtrix to reflect the TDD.

## 2. A (Relatively) Simple Image Convention

One of the greatest difficulties faced by users of Drizzle (Fruchter and Hook 1999) and Multidrizzle is aligning large sets of images either of the same field taken at different epochs and thus different orientations and guide stars, or of adjacent fields to create a mosaic. The standard way of handling this has been through pixel shifts. However, for larger fields – particularly mosaics – it is preferable to think in sky coordinates. Therefore the first part of our program has been to redesign the HST image headers so that all distortion information is included in the headers. To do this we are using an extended version of the Simple Image Polynomial convention (Shupe et al. 2005) which has already been used by Spitzer:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \texttt{CD1\_1} & \texttt{CD1\_2} \\ \texttt{CD2\_1} & \texttt{CD2\_2} \end{pmatrix} \begin{pmatrix} u + f(u,v) + LT_x(u,v) \\ v + g(u,v) + LT_y(u,v) \end{pmatrix} \quad (1)$$

Here $(u, v)$ represent the input pixel positions, and $(x, y)$ are output sky coordinates. $f$ and $g$ are polynomials with no term smaller than second order – thus the linear terms of the distortion are entirely contained in the CD matrix. $LT_x$ and $LT_y$ are look-up tables which follow the convention of Paper IV proposed as a FITS standard (Calabretta et al. 2004). The distortion of ACS, for instance, can be described to a few tenths of a pixel with only a quadratic polynomial. However, to get distortion errors below 0.1 pixel with ACS one needs two sets of corrections beyond a low order polynomial. One needs higher order corrections, which can be added by a lookup table (Anderson 2002), and one needs to adjust the skew of the CD matrix with time (Anderson 2007). The distortion correction look up tables have been in use with the ACS for some time – but as full size images rather than an interpolated table. Here we return them to the much smaller look-up tables, which enables us to put the information in the header. Note that in the SIP convention all of the linear terms of distortion are included in the World Coordinate System (WCS) matrix. Our software now also removes the time dependent distortion of the ACS by modifying the skew of the WCS when the SIP coefficients are created for the first time by the STScI pipeline (see Figure 1).

In addition to these primary corrections, very small corrections to (u,v) will be allowed prior to the SIP implementation above. For instance, both the ACS and WFPC2 have very small periodic changes in pixel column width ($\ll$ 0.1 pixel). Detector distortions such as these are best thought of as separate from and subject to the optical distortions, which are represented fully in the extended SIP format.

For the ACS this means the user will be trading more than 128 MB of external reference files (the DGEO – or differential geometry – files) for about an extra 50kB of header data. WFC3 at present needs no calibration beyond the SIP polynomial; however, we expect that higher order corrections will be added in the future. In the case of WFPC2 data, which is now statically archived, the user will still need to retrieve the DGEO file, which contains a correction for a small lithographic errors in the detector. BetaDrizzle will incorporate this information into the header of the WFPC2 image the first time it is run. Figure 2 shows what a present image header looks like when it comes out of the archive.

## 3. Off With Their Headerlets

Incorporating the full astrometric solution in the header allows a new approach to aligning images. One does not need to Drizzle images before aligning them. One can find the pixel location of sources in the flt images and then accurately convert these pixel locations to sky coordinates. The image sky coordinates can then be compared either to a catalog of coordinates derived from an astrometric survey (such as 2MASS) or with a catalog from another image. Because we have already removed any distortion in the image, we now only need to solve for a shift, rotation and possible scale change to bring the images into

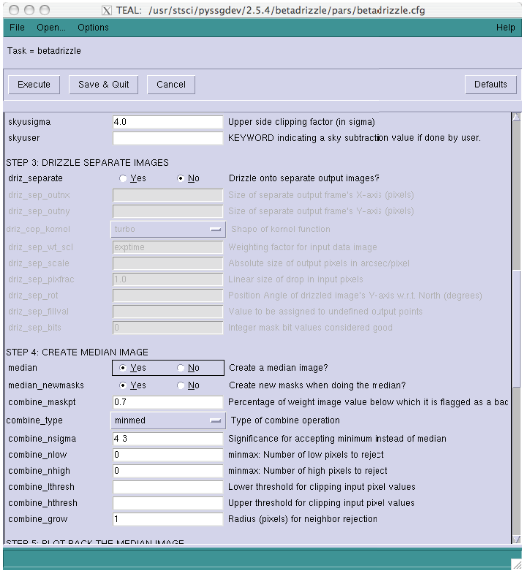
```
--> catfits j8hw27c4q_flt.fits
EXT#  FITSNAME      FILENAME             EXTVE DIMENS      BITPI OBJECT

0     j8hw27c4q_flt j8hw27c4q_flt.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     65x33       -32
9       IMAGE       WCSDVARR             2     65x33       -32
10      IMAGE       WCSDVARR             3     65x33       -32
11      IMAGE       WCSDVARR             4     65x33       -32
12      BINTABLE    WCSCORR                    18Fx10R
-->
```

Figure 2: A listing of an ACS image showing the new extended FITS format we describe in this paper. Extensions 0 through 6 should already be familiar to ACS users. Extension 7 is the file for the detector distortions, and extensions 8 through 11 are the replacement for the present DGEO file. Extension 12 is a binary file which keeps track of the astrometric solutions and their distortion files. This binary table allows us to handle multiple astrometric solutions (and their potentially different correction tables) in a single header. The image shown here has only one astrometric solution – the original solution provided by the archive.

alignment. The solution derived can then be directly applied to the image header. Removing the requirement of drizzling the image not only speeds up the process of aligning images it can improve alignment accuracy, as drizzling a single image can introduce small astrometric errors when the original images are undersampled. A task presently called ”TweakReg” has been written which can in a single call find the sources in an HST image, match these to sources in a catalog (obtained either from another image or from the literature) and update the image header of the image so that the sky coordinates of the objects in the image match those in the catalog.

Our approach also enables another important leap. Once one has obtained an astrometric solution for an image, the header now contains all the information that is needed to describe that solution. If one were to package the astrometric information into a small FITS file, one could imagine passing that solution directly to another user. Our software will allow just this. We call the small astrometric FITS files *headerlets*. Users will be able to exchange these headerlets among themselves or back and forth with the archive. Thus, for instance, if a Multi-Cycle Treasury program derives good astrometric solutions for all of its images, it will be able supply headerlets for all of those solutions to the archive and other users will be able to download them.

There is, however, no reason why an image should be limited to only one astrometric solution in its header – a single image could be fit to multiple (and slightly discrepant) astrometric catalogs. We will allow multiple astrometric solutions in the header and we will provide a simple means for the user to choose which of these solutions to make the default. If all of these solutions use the same SIP coefficients (as will generally be the case), then the presently available DS9 software can already correctly switch between the multiple solutions.



TEAL: /usr/stsci/pyssgdev/2.5.4/betadrizzle/pars/betadrizzle.cfg

Task = betadrizzle

|  |  |  |
|---|---|---|
| skyusigma | 4.0 | Upper side clipping factor (in sigma) |
| skyuser |  | KEYWORD indicating a sky subtraction value if done by user. |
| STEP 3: DRIZZLE SEPARATE IMAGES |  |  |
| driz_separate | ○ Yes ● No | Drizzle onto separate output images? |
| driz_sep_outnx |  | Size of separate output frame's X-axis (pixels) |
| driz_sep_outny |  | Size of separate output frame's Y-axis (pixels) |
| driz_cop_kornol | turbo | Shapo of kornol function |
| driz_sep_wt_scl | exptime | Weighting factor for input data image |
| driz_sep_scale |  | Absolute size of output pixels in arcsec/pixel |
| driz_sep_pixfrac | 1.0 | Linear size of drop in input pixels |
| driz_sep_rot |  | Position Angle of drizzled image's Y-axis w.r.t. North (degrees) |
| driz_sep_fillval |  | Value to be assigned to undefined output points |
| driz_sep_bits | 0 | Integer mask bit values considered good |
| STEP 4: CREATE MEDIAN IMAGE |  |  |
| median | ● Yes ○ No | Create a median image? |
| median_newmasks | ● Yes ○ No | Create new masks when doing the median? |
| combine_maskpt | 0.7 | Percentage of weight image value below which it is flagged as a bac |
| combine_type | minmed | Type of combine operation |
| combine_nsigma | 4 3 | Significance for accepting minimum instead of median |
| combine_nlow | 0 | minmax: Number of low pixels to reject |
| combine_nhigh | 0 | minmax: Number of high pixels to reject |
| combine_lthresh |  | Lower threshold for clipping input pixel values |
| combine_hthresh |  | Upper threshold for clipping input pixel values |
| combine_grow | 1 | Radius (pixels) for neighbor rejection |

TEAL: /usr/stsci/pyssgdev/2.5.4/betadrizzle/pars/betadrizzle.cfg

Task = betadrizzle

|  |  |  |
|---|---|---|
| skyusigma | 4.0 | Upper side clipping factor (in sigma) |
| skyuser |  | KEYWORD indicating a sky subtraction value if done by user. |
| STEP 3: DRIZZLE SEPARATE IMAGES |  |  |
| driz_separate | ○ Yes ● No | Drizzle onto separate output images? |
| driz_sep_outnx |  | Size of separate output frame's X-axis (pixels) |
| driz_sep_outny |  | Size of separate output frame's Y-axis (pixels) |
| driz_cop_kornol | turbo | Shapo of kornol function |
| driz_sep_wt_scl | exptime | Weighting factor for input data image |
| driz_sep_scale |  | Absolute size of output pixels in arcsec/pixel |
| driz_sep_pixfrac | 1.0 | Linear size of drop in input pixels |
| driz_sep_rot |  | Position Angle of drizzled image's Y-axis w.r.t. North (degrees) |
| driz_sep_fillval |  | Value to be assigned to undefined output points |
| driz_sep_bits | 0 | Integer mask bit values considered good |
| STEP 4: CREATE MEDIAN IMAGE |  |  |
| median | ○ Yes ● No | Create a median image? |
| median_newmasks | ● Yes ○ No | Create new masks when doing the median? |
| combine_maskpt | 0.7 | Percentage of weight image value below which it is flagged as a bac |
| combine_type | minmed | Type of combine operation |
| combine_nsigma | 4 3 | Significance for accepting minimum instead of median |
| combine_nlow | 0 | minmax: Number of low pixels to reject |
| combine_nhigh | 0 | minmax: Number of high pixels to reject |
| combine_lthresh |  | Lower threshold for clipping input pixel values |
| combine_hthresh |  | Upper threshold for clipping input pixel values |
| combine_grow | 1 | Radius (pixels) for neighbor rejection |

Figure 3: One of the advantages of the new Teal GUI. In this case the user has decided not to do separate Drizzles of the images. In the first image she has turned off the separate drizzles (and the parameters associated with that choice are grayed out). In the second image she has turned off the median step. While this sort of parameter handling is now common in, for instance, the Mac OS X operating system, it is something IRAF still cannot do.

## 4. A New Engine and Chassis

BetaDrizzle has been built upon a philosophy of modularity. BetaDrizzle has two primary Python scripts that call modular components.The modular components are themselves written in Python or C, if that language would be more efficient (Drizzle itself has been converted from Fortran to C). The first primary script handles the coordinate transformations using the full distortion model, while the second is responsible for resampling (drizzling) the image. The new modular BetaDrizzle will enable the user to more easily customize the program to the specific needs of their data, while making the standard use of the software even simpler.

BetaDrizzle also employs a new graphical user interface, called Teal, to handle parameter input. Teal behaves in many ways like the ”epar” mechanism of IRAF, but has new features that will allow for a better and simpler user experience. For instance, as shown in Figure 3, Teal can grey out parameters depending upon choices made by the user. Teal also makes it easy for the user, or the Institute, to create ”configuration” files – that is sets of parameters – that can be named and saved, and written to handle specific instruments and situations. This will enable the Institute to provide better tailored parameter files and for the user to adapt those files for his or her use.

## 5. Conclusions

In this short paper we have provided a basic introduction to program to substantially advance the ability of users to combine HST images. The header philosophy and programs we have developed will allow users to determine offsets between images directly from their flt files without the need for drizzling, and to update the image headers to reflect the new astrometry. The full astrometric solution, including all distortion corrections, will be in the image header, and users will be able to pass astrometric solutions back and forth to each other or the archive through the use of headerlets. The BetaDrizzle program itself is modular and will use easily modifiable configuration files making it far simpler for users to adapt it to their particular needs. And BetaDrizzle uses a new GUI, Teal, that is a substantial advance beyond the present IRAF epar interface. We believe that all of these changes should make accurately combining HST images far simpler and less error prone.

## References

Anderson, J. 2002, in the Proceedings of the 2002 HST Calibration Workshop, S. Arribas, A. Koekemoer, and B. Whitmore, eds.

Anderson, J. 2007, Instrument Science Report on ACS, No. 2007-08,

Fruchter, A.S. & Hook, R.N. 2002 PASP, 114, 144

Koekemoer, A.M., Fruchter, A.S., Hook, R.N. & Hack, W. 2002, in the Proceedings of the 2002 HST Calibration Workshop, S. Arribas, A. Koekemoer, and B. Whitmore, eds.

Shupe, D.L., Moshir, M., Li, J., Makovoz, D., Narron, R. & Hook, R.N. 2005, in Astronomical Data Analysis Software and Systems, XIV, ASP Conference Series.

Calabretta, M.R., Valdes, F.G., Greisen, E.W. & Allen, S.L. 2004, `http://www.atnf.csiro.au/people/mcalabre/WCS/dcs_20040422.pdf`

# Part 3. Contributed Posters