THE STDGDC FORMAT: A DETECTOR-BASED DISTORTION SOLUTION

The distortion solutions described in the Instrument Handbook and provided on the Reference File page are designed to map the detector pixels into the V2-V3 telescope plane for the purposes of absolute astrometric calibration. To facilitate transformations among `flt` images, it is common to use more locally focused distortion solutions. These solutions map the distorted frame of the detector into the closest possible distortion-free frame. Typically, the correction is zero at the center of the detector and its scale and orientation match that of the *y* axis of that pixel. For all the other pixels in the detector, the correction simply tells us where the center of that pixel is located relative to the central pixel.

Like the official solution, these solutions are constructed from polynomials plus various look-up tables to account for detector artifacts or filter anomalies. In transformations from one FLT frame to another, it is necessary to use the forward distortion solution to convert a raw coordinate into the adopted reference frame (often a linear transformation is necessary as well). Similarly, we need the inverse solution to convert a reference-frame coordinate into an individual-raw-frame position. We clearly need easy access to both the forward and inverse distortion solutions. Unfortunately, it can be complicated to find an inverse solution when the distortion solution has many layers (polynomial, plus look-up table, plus periodic errors, etc). For this reason, we adopted a format for the distotrion solution that makes both directions as easy and as fast as possible.

We will put our distortion solutions into a "standardized" format (called STDGDC format) to provide a similar interface or all detectors. The solution is provided in a multi-extension fits file. Each extension is a fits image.

The first extension is the size of the detector (1014×1014 pixels for WFC3/IR, 4096×4096 for WFC3/UVIS). The value of each pixel in the first-extension image tells us the *x* coordinate of that pixel's mapping into the reference frame. Similarly, the second-extension image tells us the distortion-corrected *y* position. In these images, the distortion-correction mapping is available only for the centers of the pixels. Bilinear interpolation can be used to get the distortion correction at any non-integer location in the pixel grid. There is no need for higher-order interpolation of the pixel grid.

The third extension is also the size of the original detector. It gives the pixel-area correction that converts a magnitude measured in the `flt` image into a flux-preserving system. The flat fielding of the `flt` images has been done to preserve surface brightness, and in the presence of non-linear distortion this means it does not preserve flux. This correction should be added to `flt`-measured magnitudes.

The fourth and fifth extensions provide the inverse distortion solution. These images are 1148×1015 pixels in size for the WFC3/IR correction, corresponding to the entire extent of the distortion-corrected frame. Each pixel in this image maps a particular location in the distortion-corrected frame back into the raw detector frame. Since we chose to constrain our distortion solution to have no correction for the central pixel of the detector (i.e., pixel [507,507] maps to [507,507]), it often happens that the correction maps to negative (xc, yc) coordinates at the edges of the detector. We cannot of course have negative pixel locations in an image, so we use the `LTV1` and `LTV2` keywords to allow an offset. If we have a distortion-corrected location of (xc, yc), then the corresponding raw detector coordinate can be found by bi-linearly interpolating the 4th and 5th extension images at location: x = xc + (`LTV1-1`) and y = yc + (`LTV2-1`). Both the forward and the reverse distortion solutions are needed to relate positions measured in different `flt` images to one another.

It is worth noting that these image-based solutions are extremely easy to use and can be evaluated very rapidly. They are much faster than a polynomial and look-up-table correction, and much much faster than inverting a polynomial and several other layers of solution.