One-Pass HST Photometry

with `hst1pass`

Jay Anderson
July 5, 2022

ABSTRACT

This document describes and announces the public release of a software routine, `hst1pass`, that has been optimized for PSF-fitting photometry on undersampled HST images. Previous versions of the code have been written for individual HST instruments and released as a part of various instrument-specific ISRs. But over the last few years, the code has been generalized to work for all of HST's main imagers (WFPC2, ACS/HRC, ACS/WFC, WFC3/UVIS and WFC3/IR). It also runs in aperture-photometry mode and, as such, can be run on `_drz` and `_drc` products or even non-HST images. The program itself is written in FORTRAN, but a simplified version in Python will be made available soon. In its typical usage, the user specifies some simple finding parameters, and the routine reads in (1) an HST image (`_flt` or `_flc`), (2) a PSF and, (3) a distortion solution. The routine then goes through the image pixel-by-pixel and returns a list of stars found and measured in the image. This star list can then be collated with similar lists, such as from a set of dithered exposures from the same program. In a future ISR the collation process will be described in more detail, but a simplified version of the collation software is provided here to facilitate preliminary analysis.

1. Introduction

Photometry and astrometry with the Hubble Space Telescope (HST) are both easier and harder than on traditional ground-based telescopes. HST's prime location above the Earth's atmosphere means that it does not suffer from seeing-related turbulence or from chromatic refraction off from the zenith. Its point-spread function (PSF) is exquisitely stable relative to PSFs from the ground. HST also does not suffer from atmospheric extinction or atmospheric background. The FWHM of HST's PSF is typically 100 mas, about 5 times tighter than what is considered

excellent non-adaptive seeing from the ground. As a result, stars in HST images find themselves on local backgrounds per pixel that are 25 times lower than in ground-based images, and therefore they can be detected to *much* fainter brightness levels.

HST photometry is not without its challenges, however. One challenge is that its PSF in most detector/filter combinations is undersampled, which means that it is so sharp relative to the pixel scale that images of stars are not fully resolved in a single exposure. It is still possible to measure exquisitely accurate fluxes and positions in the presence of undersampling, but we must take special care if we hope to obtain the highest possible accuracy. Another challenge is distortion. In order to maximize throughput, HST's cameras were designed with as few optical surfaces as possible. The telescope's designers prioritized sharp star images and a minimum of reflections at the expense of extreme distortion. HST's imagers typically have large linear distortions and sizable higher-order distortions. The linear distortions mean that its square detectors often project to a footprint on the sky that is more than 10% non-square: a rhombus, rectangle, or parallelogram. In addition to these linear distortions, the non-linear distortions can have amplitudes of several tens of pixels and have significant terms down to fourth or even fifth order. Finally, the detectors and the filters often have irregular distortion "fingerprint" patterns that cannot be described well via polynomials (See Kozhurina-Platais 2014).

HST's orbit keeps it well above the atmosphere, which means that both the PSFs and the distortion are quite stable. Thus, we can design calibration programs to measure the distortion and extract PSFs from fields that have a good density of bright stars, and we can then use these PSFs and distortion solutions in fields that do not contain such in-hand calibration sources.

Even so, while the PSFs and distortion solutions are extremely stable, they are not perfectly stable. As HST points in different directions relative to the Sun, it experiences different amounts of warming, and this results in small variations in its focal length (a process known as "breathing"). As a result, the PSF, the plate-scale and distortion solution change slightly (at the 1% level). These changes can be calibrated (see Kozhurina-Platais et al. 2009ab, and Kozhurina Platatais et al. 2015, Anderson & Bedin 2017, Anderson 2018, and Bellini 2018), but the variations have a very small effect on photometry and astrometry, particularly when measurements are made differentially.

To do photometry and astronomy on HST images, many groups use standard tools from the ground, such as DAOPhot (Stetson 1987) or DoPHOT (Schechter et al. 1993) to analyze the drizzled images, which are stacked versions of the un-resampled _flt or _flc images. More recently, Photutils (Bradley et al 2020), an astropy package, has been gaining popularity as well. The drizzling process (Fruchter & Hook 2002) is designed to rigorously preserve the zeroth moment of a star's distribution (its flux), and therefore the photometry on these images is quite good. It is much harder to preserve higher moments during the resampling process, therefore astrometry (the first moment) and FWHMs (the second moment) are harder to measure reliably in resampled (drizzled) images, at least for unresolved or marginally resolved objects.

There are indeed several advantages to analyzing the drizzled images. First, distortion has been removed, so it's easy to transform from one frame to another and do absolute calibration. Also, drizzled images are usually the combination of several dithered exposures, so one can make a single measurement that combines an entire visit of images, or even multiple visits imaging the same field. Finally, there are often low S/N sources that are too faint to be observed reliably in

individual exposures; multiple exposures are needed to bring up their S/N. The drizzle-combination of multiple deep exposures allows this.

However, there are also disadvantages that come with analyzing the drizzle products. As mentioned above, the drizzle process does preserve flux, but source position and shape are much harder to maintain through resampling. The individual pixels in drizzled images represent somewhat fuzzy constraints on the sky, since the pixel values in the drizzled images come from combining pixel values from multiple contributing images, each of which samples the astrophysical source differently. For example, the effective "center of light" of each drizzled pixel may not necessarily be at the center of that pixel. The chosen "dropsize" parameter can also affect the end result. Conversely, the pixels in the `_flt` / `_flc` images *are* direct constraints on the sky. They have well known errors from Poisson noise, readnoise, flat-field errors, etc. And, as such, they can be modeled exquisitely if we have a model for the PSF, distortion, and the scene.

Once we have a PSF model in hand, it is not hard to measure point sources in individual un-resampled _flt / _flc images. Similarly, once we have a model for the distortion, it is not hard to combine measurements made in a set of dithered exposures into a single comprehensive catalog. But even though the problems have been solved, it has not been easy for non-expert users to do such things. Implementing these solutions in a way that the typical scientist can use them is the goal of this software suite.

The `hst1pass` routine described in this document is designed to do the first task, namely to find and make accurate measurements of sources in individual images. Routines like `hst1pass` that do the basic PSF-fitting to pixels have been delivered to the public along with PSFs over the years (for example, in Anderson & King 2006), but they have not gained much traction in the community. This is likely due to the fact that measuring stars in individual exposures is only half the challenge. The other half involves combining these measurements into a useful framework for analysis. Particularly when high-precision astrometry is a goal, it is critical to go back and forth from distortion-corrected frames to raw frames to cross-identify stars and collate lists. A second program, `hst2collate`, is provided here to perform this second task. It will be described more fully in a subsequent ISR, but we provide a bare-bones version of this routine along with this ISR to enable preliminary analysis and artificial-star tests.

This document is largely a user's manual for `hst1pass`. **Section 2** describes how to download it and compile it. **Section 3** describes its basic inputs, such as finding parameters, PSFs, distortion solutions, and images to operate on. **Section 4** describes in detail how the routine finds and measures stars, including photometric and astrometric processing that gets done. **Section 5** describes the wide variety of possible outputs that can be produced. **Section 6** provides some tips for analysis. **Section 7** lists some use cases that are worked out in directories on the website. Finally, **Section 8** puts the routine into contexts and acknowledges that it is a work in progress and will be updated over time, likely with input from users. **Appendices C**, **D**, **G**, **H**, **P**, **S**, and **Z** go into more detail about CTE corrections, distortion corrections, the STDGDC files, the barebones version of `hst2collate` being provided, PSFs, Saturated stars, and possible future features that could be implemented if there is sufficient community interest.

Table of Contents

2. Downloading and Compiling

The `hst1pass` routine is being made available to the community on an as-is basis. The routine and its antecedents have been used for many years by many collaborators for science, but there are doubtless things it does imperfectly. The author will strive to update it and fix problems that are brought to his attention, but this will not be instantaneous. **Section 8** will cover how to find out about modifications/improvements.

2.1   Downloading

The routine can be downloaded from the link pointed to by either of the following websites:

https://www.stsci.edu/hst/instrumentation/wfc3/software-tools

https://www.stsci.edu/hst/instrumentation/acs/software-tools

There will also be a `FAQ.txt` file there that should be read before sending questions to help@stsci.edu. The directories on the linked website also provide PSFs, distortion solutions, a subset of the GAIA DR3 database, and other useful products.

2.2   Compiling

The routine is written in FORTRAN. Most of the code is F77-compatible, but it does make use of a very small number of F90 features used (such as dynamic memory allocation). A more limited version of the routine will soon be available in Python (see Bajaj 2022).

To compile the routine, on a terminal prompt with the code in the current directory, simply type:

```
gfortran hst1pass.F -o hst1pass.e <cr>
```

FORTRAN compilers often become more finicky over time, and sometimes code that compiles on one machine in one version of the compiler has problems with other machines and other versions of the compiler. Thankfully, by tightening up issues with variable types and such things, most of these issues can be resolved. As compilation problems are discovered, the code will be updated to ensure compliance with the lastest compiler requirements.

2.3   Running the routine

When the `hst1pass` routine is run with no arguments, it provides a list of arguments, an example of how to use them, and a very high-level description of them. We will describe these input parameters in detail in the next section. Note that the arguments with an equal sign must have no spaces around the equal sign, and arguments that include spaces (such as `"PSF=APPHOT 2.99 6 9"`) must be enclosed fully in quotes, to ensure that the command-line interpreter will process them as a single argument.

Here is what a current no-parameter run of the routine looks like:

```
./hst1pass.e


This routine takes several args (in any order, 3 required)

        (defaults are shown first for some quantities,
                 examples for others)
hst1pass

  REQUIRED ---
    HMIN=5                              (integer; min isolation)
    FMIN=1000                           (real; min centrl flx)
    "PSF=APPHOT 2.9 6 9"/               (psf spec)
         STDPSF.fits/STDPBF.fits/
    IMG1.fits                           (command-line list)
    OUT=XYMpqUVrdxym                    (columns to output;
                                         must have at least one;
                                         can have multiple files)
  OPTIONAL ---
   "IMG2.fits[I1:I2,J1:J2]"             (limited regions in img)
    FITSs=file.list                     (file list of inp images)
    GDC=NONE/STDGDC.fits                (distortn spec; def=NONE)
    REG=xy,XY,uv,rd                     (region file to output)
    DOSATD+/-                           (meas satd stars? def+)
    PMAX=9999999                        (max pixel value)
    QMAX= 0.5                           (PSF quality of fit; def)
    CMIN=-1.0                           (centrl-pix excess; def)
    CMAX=+0.1                           (centrl-pix excess; def)
    PERT=0                              (perturb PSF? default=0)
    FOCUS=-1(find),0(mid),rF            (find/specify focus?)
    KSEL=0/1/2                          (restrict chip? def=no)
    IMIN=500 IMAX=510                   (restrict col range?)
    JMIN=922 JMAX=932                   (restrict row range?)
    ART_UVW=FILE.UVW (xym,XYM,UVW)      (add art stars)
    SHOW_USE-/+                         (output image searched)
    SHOW_FND-/+                         (output finding image)
    SHOW_REF-/+                         (output ref-frame img)
    SHOW_SUB-/+/~                       (make subtracted  image)
    SHOW_ART-/+/~                       (art-star image)


OUT= options for columns to output ; can output multiple files
     for the same finding list of stars

   ASTROMETRY ---
     x,y = raw chip-dependent x y coord (no CTE corr)
     X,Y = raw chip-dependent x y coord (after CTE corr)
     u,v = distortion-corrected x, y coord
     U,V = distortion-corrected X, Y coord (in WCS frame)
     r,d = RA, Dec (in degrees, from UV)
     R,D = RA, Dec (GAIA DR3 corrected; future)
```

```
PHOTOMETRY ---
   m   = instrumental mag (no CTE corr)
   M   = instrumental mag (after CTE corr)
   w   = instrumental mag, M + pix area correction)
   W   = instrumental mag, w + exptime normalization
   z   = flux (no CTE corr; can be negative for forced)
   Z   = flux (CTE corr; can be negative for forced)

VARIOUS ---
   s   = sky value
   S   = sky value (with postflash/dark electrons included)

   h   = isolation index (used for HMIN finding assessment)
   f   = crude 2x2 flux (used for FMIN finding assessment)
   q   = quality of fit (0 = perfect)
   c   = chisq of fit
   p   = brightest pixel value
   P   = central PSF value
   e   = generic error estimate in mags/pixels
   i   = local-max column location
   j   = local-max row location
   k   = chip number
   I   = row (zero-padded with I0000, for easy grepping)
   J   = column (zero-padded with J0000, for easy grepping)
   K   = chip number (with K1, for easy grepping)
   n   = number of saturated pixels associated with star
   N   = star number N00007
   c   = chisq for star
   C   = cen-xs for star
   o   = fraction of flux in aperture from possible neighbors
   O   = super conservative value for O
   t   = time of observation (fractional years, 2017.348)

PBAP photometry --- PSF-Based Aperture Photometry
   1   = 1x1-pixel PBAP photometry
   2   = 2x2-pixel PBAP photometry
   3   = 3x3-pixel PBAP photometry
   4   = 4x4-pixel PBAP photometry
   5   = 5x5-pixel PBAP photometry
   6   = 3.0-pixel-radius PBAP photometry
   7   = 3.5-pixel-radius PBAP photometry
   8   = 4.0-pixel-radius PBAP photometry
   9   = 4.5-pixel-radius PBAP photometry
   0   = sky from PBAP photometry
```

3. Input Parameters

As its name suggests, the `hst1pass` routine is focused on one-pass photometry. It may eventually have bells and whistles that allow it to do somewhat more than that[1], but its focus is on point sources that can be treated as "isolated", and hence do not require neighbor subtraction. The HST PSF is undersampled in most of HST's cameras, so stars do not have to be particularly far apart in order to be considered "isolated" and measured well without taking their neighbors into account. Conversely, when stars are close enough together to require neighbor subtraction, then that is *much* harder to do in HST's undersampled regime.

The typical aperture the routine uses for HST's detectors is 5×5 pixels, which has an effective radius of 2.8 pixels. This may sound small, but this aperture typically contains more than 85% of the light that lands in a 10-pixel-radius aperture[2]. The fraction of light that falls within this aperture varies by just 2.5% as the star moves from the center of a pixel to the corner of a pixel (this variation is naturally accounted for in the PSF-model fitting). Since such a small aperture contains such a large fraction of the flux, it makes sense to focus our fitting on the very few pixels where the source intensity is largest relative to that of its neighbors.

3.1 Required Finding Parameters

The routine has four required parameters: **HMIN**, **FMIN**, a **PSF** specification, and at least one image to search. If any of these is missing the routine will not proceed.

HMIN and **FMIN**

The quantities **HMIN** and **FMIN** regulate the finding process, which involves going through the image pixel by pixel and identifying all the pixels that might contain a star of interest. For each pixel, the routine evaluates the quantities **h** and **f**. Note that in many ground-based finding routines, the scene is convolved with a "matched filter" kernel (similar to the PSF) to ensure that signals in the image that are similar to the PSF have the highest possible signal-to-noise for finding. Since HST's filters are moderately undersampled, convolving the image with anything tends to lower the S/N. For that reason, we simply search the original image pixel by pixel and assert that any pixel that is a local maximum (greater than its 8 surrounding neighbors) might contain a star.

The quantity **h** reflects how far outward (radially) from a pixel of interest we have to go to find a brighter pixel. If **h** is small, then a source is not very isolated, if it is large then it is more isolated. A pixel that is a local maximum will have an **h** of at least 2. A star that has no brighter neighbors within 9 pixels will have **h** = 9.

---

[1] See **Appendix Z**.

[2] The remaining 15% of the light is distributed in an extended halo that can be seen for bright stars out to hundreds of pixels.

The quantity **f** (aka **f₂ₓ₂**) is a quick-and-dirty measure of the flux, from the star's brightest 2×2 pixels and a super-local sky value extracted from an annulus just around that. No PSF model is used to determine **f₂ₓ₂**. It's simply a rough selection tool — a first cut for potential sources.

The **HMIN** and **FMIN** parameters allow users to select only bright and/or isolated objects. One could choose **HMIN** = 2 and **FMIN** = 0 to measure every single local-maximum pixel in the image as a star. This will typically produce one detection for every 9 pixels in the image, potentially amounting to 1.8 million possible "detections" in a 4096 × 4096 ACS or UVIS image. Alternatively, one could choose **HMIN** =9 and **FMIN** =10000 to get only the very brightest stars. A typical use case has **HMIN** = 5 and **FMIN** = 1000: it finds the relatively isolated stars with S/N of at least ~30.

PSF

A PSF can either be specified by pointing to a FITS file in STDPSF or STDPBF format, or by specifying an aperture-sky pair for aperture photometry. See **Section 3.3** for more details.

OUT

Users should also specify an output to be generated. **Section 5** describes the list of possible output quantities.

At least one FITS-format HST image to run on

At least one image in FITS format must be supplied for `hst1pass` to run. The image supplied can be in various HST formats (typically `_flt` or `_flc` HST images). These are the image types that the routine can use HST PSFs with. The routine also allows `_drz` and other image types for aperture photometry. Multiple images can be listed on the command line, explicitly or with wildcards. A list of images to run on can even come from a file with a `FITSs=file_list.txt` command-line argument.

A Simple Example Execution of `hst1pass`

The simplest possible run of the program looks like:

```
./hst1pass.e HMIN=5 FMIN=2500 PSF=STDPSF_WFC3UV_F606W.fits  \
             OUTPUT=xym ibc301qrq_flt.fits
```

The order of the parameters does not matter. Multiple specifications of **OUTPUT** are allowed as are multiple images.

The image listed above is a 40s exposure through filter F606W of the starfield at the center of the great globular cluster, Omega Centauri. We will discuss the **OUTPUT** specification in detail in **Section 5**, but suffice it to say that the above program call will generate a list of sources and will report for each source its (x, y) position in the image and its instrumental magnitude, m, which is simply $-2.5 \times \log_{10}(\text{flux})$.

In the rest of this section, we will describe additional parameters that can change the search criteria, change the measurement algorithm, provide additional outputs, etc.

3.2 Additional Finding Parameters

In addition to `HMIN` and `FMIN`, there are a few other parameters that regulate the finding.

QMAX

One of the quantities that is measured when `hst1pass` fits a PSF to stars is **q**, a "quality of fit" metric. This is simply the sum of the absolute value of the residual of the PSF-fit over the standard 5×5-pixel PSF-fitting aperture divided by the measured flux:

$$q \equiv \Sigma_{AP} \mid P_{ij} - z_* \psi_{ij} - s_* \mid / z_*,$$

where P_{ij} is the pixel value, z_* is the star's flux, ψ_{ij} is the fraction of the star's light that should land in that pixel based on the PSF model and the measured position of the star (x_*,y_*) relative to pixel [i,j]. Finally, s_* is the measured sky value.

Figures 1 and 2 show examples of the selections that the PSF-based **q** parameter allows.
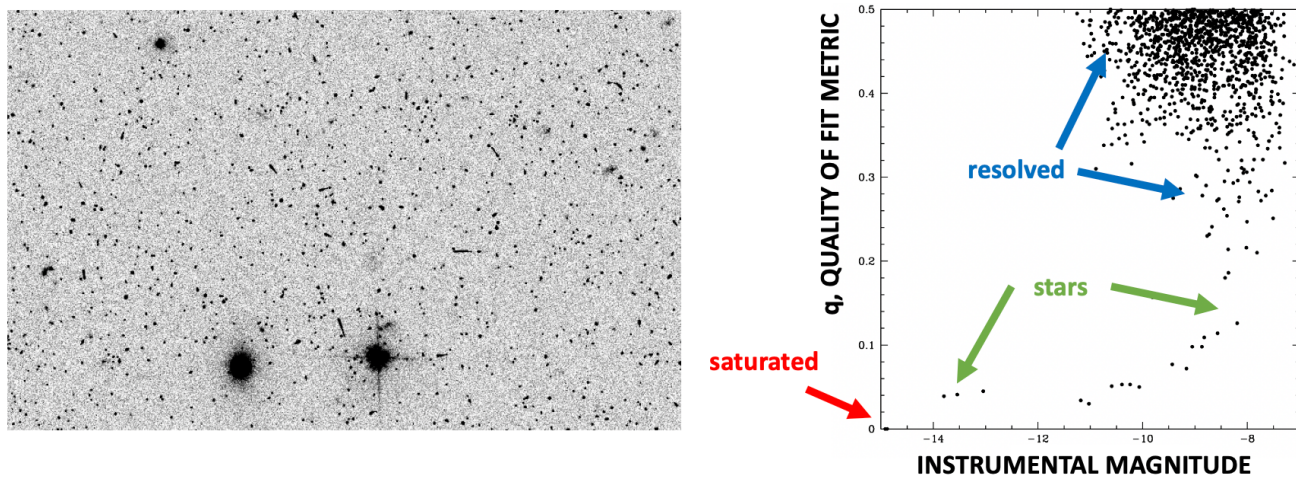
Figure 1: (Left) a single F606W ACS/WFC exposure of the Ultra Deep Field. (Right) the plot of q against instrumental magnitude for the sources identified. Note that below m ~ −10, stars have about the same value of q, but above this, the increasing S/N in the star affects the fit. The detections at the top with poor fits are either galaxies, cosmic rays (CRs) or unsubtracted hot pixels.

The quantity **q** is zero for stars that are perfectly fit by the PSF and is ~0.2 for poorly-fit stars. If the profile is too non-star-like (i.e., **q** > **QMAX**), then the object is ignored. The default value for **QMAX** is 0.5, but users can increase that if they want to include non-stellar objects in the catalog. Saturated stars have $q \equiv 0$, since there are no central pixels to measure. Likewise, stars measured with aperture photometry have $q \equiv 0$, since there is no PSF to fit them with. In general, since the PSF model is never perfect, it is most useful to examine **q** in the context of other stars in the image, plotting **q** against instrumental magnitude for all sources can help discriminate between point sources and resolved objects.
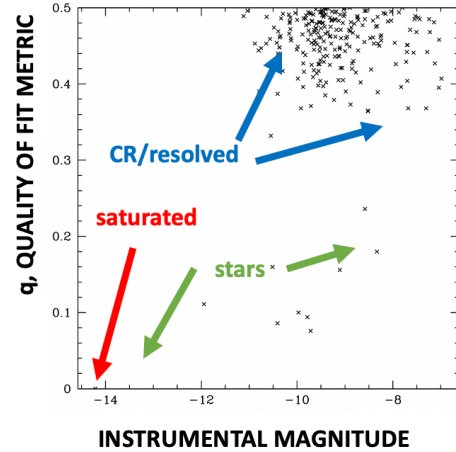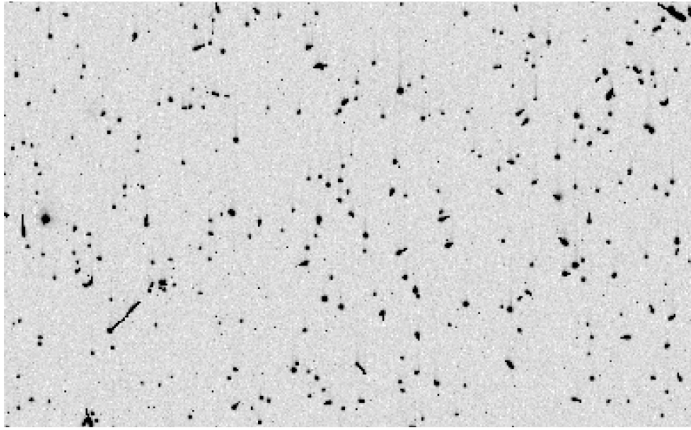
10

Figure 2: (Left) a single half-orbit exposure (ieou07kuq) through F438W with WFC3/UVIS. The vast majority of apparent sources are cosmic rays. (Right) The q parameter clearly identifies the 9 stars out of the more than 50,000 non-stellar sources. You can see one star with S/N ~100 at the left of this 400×200-pixel image. Identification of stars this way makes it easier to get accurate inputs for tweak-reg in the `AstroDrizzle` package to accomplish good alignment.

Note that **q** is often more useful to examine than the chi-squared metric **c**, since chi-squared is computed relative to the expected Poisson noise of the star, but **q** is computed relative to the noise in the PSF. The PSF is typically accurate to ~1% pixel-to-pixel and ~3% in terms of MAD residual over the fitting aperture. It's very hard to get a PSF more accurate than this, since the PSF varies due to breathing at about this level (see **Appendix P**), and to measure a PSF to better than 1% we need a large number of stars in each individual exposure with S/N > 100.

CMIN, CMAX

The quantity **C** (note capitalization) measures the fractional flux excess in an unsaturated star's central pixel (I,J):

$$C \equiv (P_{IJ} - z_* \psi_{IJ} - s_*)/z_*.$$

It can be positive or negative. If it is very positive, then it is likely a cosmic ray (CR), hot pixel, or some other artifact that is sharper than a star can possibly be. If it is very negative, then the source is much broader than a star should be. In either case, the object is ignored if it is outside of the **CMIN** / **CMAX** bounds. As with **q** above, if a star is saturated or there is no PSF, then **C** is set to zero. Also, as above for **q**, since the central-pixel intensity of the PSF can change with breathing, this metric is often most usefully when examined in a *relative* sense by plotting **C** against instrumental magnitude for sources in an exposure. This can help discriminate between objects that are and are not star-like. Note that in the output, a lowercase letter **c** corresponds to chi-squared, which is not very useful for selecting sources (see above the discussion about **q** and modeling with imperfect PSFs).

KSEL

If the detector is made up of multiple chips, then it is possible to select a particular chip to search by means of the **KSEL** command-line keyword. Only chips where **k** = **KSEL** will be searched and included in the output. The chip mapping for WFPC2 is obvious. For WFC3/UVIS and

ACS/WFC, the bottom chip is **k** = 2 and the top chip is **k** = 1. Note that the outputs for ACS/WFC and WFC3/UVIS report coordinates in a coordinate system that places the bottom chip (**k**=2) into pixels [0001:4096, 0001:2048] and the top chip (**k**=1) is placed above this in pixels [0001:4096, 2049:4096]. The top three rows in each chip of the WFC3/UVIS chips (the ones closest to the chip gap) behave strangely and have been clipped off in our analysis. The coordinate system for WFPC2 has PC1 and WF2 in the bottom 800 pixels and WF4 and WF3 in the upper 800 pixels. If you would like to visualize the as-searched image, then simply set **SHOW_USE+** on the command line.

IMIN, IMAX, JMIN, JMAX

In case a user would like to restrict a sub-region of the detector for searching, these four parameters can be specified. Pixels outside of that region will be ignored. One can also specify these by putting brackets after the input image name (remember to put it in quotes in order to avoid unfortunate parsing by the command line). For example, "iabcdefgq_flt.fits[0617,3142]" will search one pixel in the top UVIS chip, while "iabcdefgq_flt.fits[0617:0717,3142:3242]" will search a 100×100 region.

SATD

The **SATD** flag regulates whether the routine will find saturated stars. The default for the **SATD** flag is "on". If the **SATD** flag is set to off (with **SATD−** on the command line), then saturated stars are ignored in the finding procedure. Even if one does not want to use saturated stars in the analysis, it can be beneficial to have them in the output list, since that makes it easier to match output lists with catalogs or other lists. Saturated stars are generally measured to better than 5% in terms of both astrometry and photometry, so their measurements can be useful beyond catalog matching. Saturated-star recovery is discussed and demonstrated in **Appendix S**.

3.3 PSF-related parameters (PSF, FOCUS, PERT)

The `hst1pass` routine provides the option of using an empirical "effective" PSF, or doing aperture photometry. If no PSF model is available (which should be the case, for instance, when analyzing drizzled images), then one can simply specify aperture photometry with "**PSF=APPHOT 3.5 6 9**". This does aperture photometry using the whole pixels within a radius of 3.5 pixels from the star's central pixel using an interactively sigma-clipped sky taken from an annulus between $r = 6$ and $r = 9$ pixels.

If a PSF is specified, it must be an effective PSF in the "standard" STDPSF format, which is a $101 \times 101 \times N_{PSFs}$ FITS image with the header detailing exactly where the N_{PSFs} fiducial PSFs are located in the detector frame. Between fiducial locations, the effective PSF is interpolated linearly among the nearest models. The STDPSF format is described comprehensively in Anderson (2016). Library effective PSFs for the various instruments can be accessed at the `hst1pass` webpage pointed to in **Section 2**, or on the instrument pages[3]. The STDPSF models are typically arrayed in a 9×10 or 7×8 grid of fiducial models across the 4096×4096 CCD

---

[3] Note that the WFC3 and ACS teams have put together a colletion of "realized" PSFs (aka, star images) in a searchable PSF database. The effective PSFs here are ×4-supersampled models and should not be confused with the ×1-sampled star images.

detectors[4]. They were extracted from datasets early in the lifetime of the detector from images of Omega Centauri or 47 Tuc that had a near-optimal density of stars for the purpose of PSF extraction.

Alternatively, the PSF could be supplied in STDPBF format, which has an extra dimension to allow for changes in the PSF due to focus variations (see Anderson 2018 and Bellini et al 2018). If the PSF is provided in STDPBF format, the user can set the **FOCUS** parameter to specify that a particular focus level is to be used, or the user can specify **FOCUS= −1** and the program will attempt to use the moderate S/N stars in the image to sense the focus level and use that to measure stars. In practice, using a focus-diverse STDPBF and solving for the **FOCUS** does not always improve results.

One additional option in the context of PSFs is to allow the routine to "perturb" the library PSF by means of the **PERT** option. This can be done if there are several (at least 10) relatively bright, isolated stars in the field. If **PERT=1**, then `hst1pass` constructs a single perturbation PSF that applies across the entire image. *This often provides the best results*. If **PERT=2** or higher is specified, then it will generate an N×N array of perturbation PSFs spaced evenly across the detector. Of course, extracting so many independent PSF adjustments requires having plenty of bright, isolated stars that can tell us about the PSF. The perturbation PSF adjustment is added to the array of PSFs in the library model (there are N_{PSFs} of them). The perturbation typically adjusts the central pixels of the PSF by a few percent to account for differences between the PSF present in the image and the library PSF, either due to breathing or jitter. We find that when **PERT** is used, the image-to-image zeropoint variation decreases from about ±0.02 magnitude to about ±0.005 magnitude. If a perturbation PSF is determined, then the routine outputs a file named `LOG_PERTS.fits` that displays side-by-side the perturbations extracted from the various images. The units of the PSF perturbation are the same as those for the PSF itself: the fraction of the star's total flux in each pixel at a given location relative to the star's center. Solving for a PSF also generates an output a file for each individual exposure (i.e., `iabcdefgq_psf.fits`). This is a full STDPSF-format PSF and consists of the original library PSF plus the perturbation. It can be used for analysis on the image by a subsequent calling of `hst1pass` or any other routine that uses STDPSF-format PSFs.

[Appendix P](#) explores how using perturbations or solving for focus can affect results. The main benefit of going beyond a static "library" STDPSF model is that the photometric zeropoint varies less from exposure to exposure. Since calibration is generally done separately from `hst1pass`, and inter-image zeropoints are often solved for in the collation process, in most cases it is adequate to use a static "library" PSF. The author recommends starting with the unadjusted STDPSF models and only opting for more sophisticated models if the analysis warrants it.

3.4 Distortion Specification

For historical reasons, the `hst1pass` routine handles distortion in the pixel frame in its own particular way. The target distortion-corrected frame (u, v) is designed to be as close as possible to the original frame (x, y). When designing a distortion solution, one has four degrees of

---

[4] The ACS/WFC model has an array of 9×10 PSFs, while WFC/UVIS was found to be best represented by 7×8.

freedom, since undistorted frames can differ from one another by "conformal" transformations[5] and still be distortion free.

The convention adopted here places the pixel at the designated "center" of the frame, such that it has no distortion correction in either (x, y) position, in y pixel scale, or in y pixel orientation. If there are any linear skew terms in the distortion solution, the distortion-corrected frame will have a non-unity pixel scale along x and/or a non-zero x pixel orientation at this location. Such terms must result when a square detector is mapped to a rhombus or a parallelogram on the sky.

The distortion solutions for `hst1pass` make use of an image-based approach, which enables very fast forward and inverse transformations. The STDGDC files store distortion solutions in a 5-extension `fits` image that provides astrometric and photometric[6] corrections for distortion. **Appendix G** provides more details on the STDGDC format.

The distortion solutions provided along with the `hst1pass` release have very assorted pedigrees. They have been constructed at various times for WFPC2, ACS/HRC, ACS/WFC, WFC3/UVIS, and WFC3/IR using various observations of star clusters. They often account for pixel irregularities[7] and the "fingerprint residual" specific to various filters. The ACS/WFC solution is also known to vary over time (see, Anderson 2007, Ubeda L. & Kozhurina-Platais 2013, Kozhurina-Platais et al. 2015, and Hoffman & Kozhurina-Platais 2020), in both its linear and non-linear terms, but those variations are not yet addressed in the solutions we have available[8]. Also, while there are several solutions to choose from, we do not have solutions for every detector/filter combination. We generally suggest that users choose the closest filter in wavelength, even though some of the corrections (such as the filter-specific fingerprints) are not really wavelength related.

The ACS solutions provided along with the `hst1pass` release come directly from the official IDCTAB solution (constructed in Kozhurina-Platais 2014). In **Appendix D** we will show that this is better than the solutions constructed by Anderson & King early in ACS's lifetime (Anderson & King 2006). The official solutions for the other detectors have not been as thoroughly validated as that for ACS, so we will provide here the astrometry-focused solutions constructed by Anderson & King (2004) for the HRC, Anderson & King (2003) for WFPC2, Bellini et al. (2011) for WFC3/UVIS, and an unreferenced solution for WFC3/IR constructed by the author. Archival proposal 15632 (PI-Casetti-Dinescu) aims to improve the astrometry for WFPC2, and we hope to fold in improved solutions for this detector in short order. Some results have already been written up (See Casetti-Dinescu et al. 2021), and new STDGDC files have been generated, it will be do documented in the `hst1pass` update notes.

Finally, it is worth noting that a distortion solution is not required for all usages of `hst1pass`. For this reason, the default distortion solution is "**NONE**". Note that if users do not specify a

---

[5] A "conformal" transformation involves 4 parameters: a 2-D offset, a rotation, and a scale change.
[6] In the presence of non-linear distortions, pixels subtend different areas and the flat-field convention adopted in the HST pipeline preserves surface brightness rather than counts. So a PAM (pixel-area map) correction is necessary.
[7] Pixel irregularities consist of WFPC2's 34-row skip, WFC/ACS's 68-column pattern, and WFC3/UVIS's zonal correction.
[8] We note that linear variations in the distortion solutions can be subsumed into the transformations when general 6-parameter transformations are used to map from distortion-corrected frame to distortion-corrected frame. Formally, distortion-corrected frames should require only "conformal" 4-parameter transformations to relate them, but with the extra two parameters we find we can implicitly deal with many incidental issues, so much so that the author rarely has to consider how stable the telescope's skew terms are.

distortion solution but select an output option that requires a distortion solution (such as `u`, `v`, `U`, `V`, `r`, `d`, etc.), then the routine will do no distortion correction.

3.5 Reference Frame Specified by WCS

A reference frame can be specified with the **`WCS=`** keyword on the command line. The WCS is a mapping from RA/Dec into a particular pixel frame, specified by a reference pixel coordinate (`CRPIX1`, `CRPIX2`) and a reference sky coordinate (`CRVAL1`, `CRVAL2`) corresponding to that pixel location, along with the linear terms of the transformation (`CD1_1`, etc.).

The WCS can be specified explicitly with a keyword such as: `"WCS=(5000,5000,80.5,−69.5,50.0)"`, which maps $(\alpha, \delta) = (80.5, -69.5)$ to pixel [5000,5000] with north up (+y), east to the left (−x), and a pixel-scale of 50.0 mas/pixel. When specifying the WCS explicitly like this, it is always North up. Be sure to include this in quotes on the command line, so that the parser will not try to interpret the commas or parentheses.

Alternatively, the WCS can be specified *implicitly* from the header of an image, for instance **`WCS=HDR{iabcdefg0_drz.fits}`**. This can be useful if the user desires to analyze a set of images that have all been associated into a single drizzle product. This WCS specification will "push" the reference-frame coordinates into the appropriate drizzled frame. If no WCS is specified, the WCS will have the target coordinate (from the `flt`-image header) at coordinate $(0, 0)$ with the pixel-scale appropriate for that detector (and of course North up).

Finally, the WCS can be specified implicitly with a `_mat` file (an output from the `hst2collate` routine, see **Appendix H**). In this case, it will formally decouple the RA and Dec determined from the image header from the reference frame. The reference frame will instead be specified by the transformation implied by a list of point associations between star-positions in the reference frame and star positions previously measured in this particular `_flt` frame. The star positions for the point associations are given in a file with a name such as: `iabcdefgq_mat.UVuvWw`. The first two columns of the file correspond to the position for the cross-identified star in the reference frame (`UV`) and the second two to the position for the same star in the distortion-corrected frame (`uv`) of the particular exposure. The next column (`W`) is the magnitude of the source in the reference frame, and the final column (`w`) is the magnitude of the source in the local distortion-corrected frame. One of the reasons for using this WCS specification is to allow insertion of artificial stars at consistent locations with accurate PSFs into individual exposures.

3.6 Auxiliary Image Outputs (`SHOW_SUB+, etc`)

The `hst1pass` routine is capable of outputting some diagnostic images to aid in understanding what it has done. The default is to not output any additional images, but to output only the star list and possible changes to the PSF used. Various outputs can be enabled by toggling flags on the command line.

With a "+" toggle, the output image has the input image rootname and the three letters after the underscore replace `flt` / `flc`; for example, `imgestemq_sub.fits`. Note that these output images are provided as simple zero-extension `fits` images in the code's internal representation

of the detector. For example, WFC3/UVIS has two 4096×2051 detectors, WFC1 (the top chip) is in extension [4] of an `_flt` or `_flc` image, and WFC2 (the bottom chip) is in extension [1]. For simplicity, this is represented as a single 4096×4096 image in the meta-frame, with the bottom chip in [1:4096,1:2048] and the top chip in [1:4096,2049:4096]. Note that for WFC3/UVIS, we have stripped off the three pixels adjacent to the chip-gap, since analysis shows them to have grossly different row heights relative to the rest of the detector. ACS/WFC has a similar shape to WFC3/UVIS, except that it does not have any problem rows. For WFPC2, we put the four 800×800 chips into a 1600×1600 with the chips all oriented with the vertex in the lower left.

If the user desires to have these auxiliary images placed into the shell of a multi-extension pipeline product image, then instead of "+", one should use "~". This is only valid for real images and does not yet work for some platforms (such as WFPC2 or HRC). The reason to allow images to be inserted into existing `flt`/`flc` "shell" images is to allow them to more easily be handled by procedures (such as `astrodrizzle`) that expect a certain form for input images. Note that the error and DQ array information is not modified: only the pixel array is modified. The routine will minimally update the header in the first blank `HISTORY` line with a keyword `HST1PASS = ART/SUB/USE/etc`.

SHOW_USE+: The `_use` output image simply reports the image that was read in from the file and is searched through for stars. This image is not necessarily identical to the input image. For example, saturated-star processing may have been performed, which puts a peak at the center of contiguous saturated distributions. The name of the output image will be: `iabcdefgq_use.fits`.

SHOW_ART+: If artificial stars are added to the image (see **Section 3.7** below), an output image with just the artificial stars will be `iabcdefgq_add.fits` and the modified image will be `iabcdefgq_art.fits`. If a "shell" image is specified (`SHOW_ART~`), then it will output `iabcdefgd_flt.fits`, instead, as the image with only the artificial stars and `iabcdefga_flt.fits` as the normal image with the stars added. (Note the replacement of the 'q' in the 9th position[9] with 'd' or 'a'; this allows us to keep `_flt`, which may be needed by drizzle or other routines for downstream analysis. See **USE CASE #3** in **Section 7** for an example of how to do this.

SHOW_FND+: This integer output image reports whether a particular pixel was determined to contain a star, and if not, at which stage it was determined not to contain a star (see **Section 4** for the meaning of each number). This particular output image cannot be placed in a shell, since it is an integer image. The name of the output image will be: `iabcdefgq_fnd.fits`.

SHOW_MOD+: This image will reproduce the found stars in an image with zero sky. If this is to be placed in a shell, the image name looks like: "`iabcdefgm_flt.fits`", otherwise its name will be: `iabcdefgq_mod.fits`.

SHOW_SUB+: Normally `hst1pass` does not produce images with the identified stars subtracted, but it can upon request. It subtracts only unsaturated stars. If this image is to be

---

[9] A small number of images in the archive have a letter other than "q" in the 9th position, related to issues involved in downloading the data from the spacecraft.

placed in a shell, the image name looks like: "`iabcdefgs_flt.fits`", otherwise it is will be named: `iabcdefgq_sub.fits`.

SHOW_REF+: This is an image that has been distortion corrected and blotted into the reference frame. It has a WCS header consistent with what the pipeline put into the original image. This particular image also cannot be placed into a shell. The blotting performed here is extremely crude and should not be compared to a refined blot done by `astrodrizzle`; it is done simply for sanity-checking purposes. The name of the output image will be: `iabcdefgq_ref.fits`.

SHOW_MSK+: When this flag is set, `hst1pass` takes the saturated stars and a conservative upper-limit model of the PSF and its diffraction spikes and generates a mask that provides in each pixel the maximum possible size for PSF artifacts; this way, PSF artifacts can be avoided in finding. If this is to be placed in a shell, the image name ends with "`x_flt.fits`", otherwise it will be: `iabcdefgq_msk.fits`.

3.7 Artificial-Star Tests

A list of artificial stars (ASs) to be inserted into images before finding and photometry can be specified either in the image frame or the reference frame. Stars are added according to the input PSF model and with Poisson noise but with no other sources of noise[10]. If stars are added to WFC3/UVIS or ACS/WFC `flt` images, then the flux is diminished and the y-position shifted away from the readout amplifier, consistent with the CTE tables.

If the stars to be inserted are specified by their properties in the reference frame, then the command-line argument is **ART_UVM=file.UVW**, where `file.UVW` is a file with quantities **U**, **V**, and **W** provided in the first three columns. The routine will presume the existence of an `iabcdefgq_mat.UVuvWw` file for each exposure analyzed and will use it to transform the input **UV** position into the **uv** frame, then the inverse distortion correction is used to map this into the **XY** frame, and the inverse table-based CTE correction is applied to get the **xy** frame location where they are placed. Photometrically, a zeropoint is used to convert the input **W** magnitude into the **w** system, then the inverse pixel-area correction is applied to convert it to an **M** instrumental magnitude, and finally the inverse CTE correction is applied to yield **m**, the instrumental magnitude with which the star is inserted.

If the stars to be inserted are specified by their local-frame values, then the command-line argument is **ART_XYM=file.XYM**. Only the CTE is uncorrected to arrive at the insertion location. Of course, if the image under analysis is in `flc` format, then no inverse CTE correction is performed, since a correction has already been done. When ASs are inserted into an image, the routine produces a file `iabcdefgq_art.info` providing information about how and where they were inserted. See **USE CASE #3** for an example of how to add artificial stars into images.

---

[10] Actual stars have additional sources of noise, an inexhaustive list of which involve: a mismatch between the PSF model used to measure them and the true PSF and uncertainties in the pixel sensitivity and flat-field correction.

4. How `hst1pass` Finds Stars

The `hst1pass` program follows a very simple finding strategy. It reads in the image and the necessary auxiliary meta-information. If it is asked to, it will add in artificial stars. It then searches through the image pixel by pixel to identify stars. It subjects each pixel to a series of tests to determine whether that pixel might house a star[11].   The following 14 tests are applied:

1) **In outer border:** if a pixel is within 5 pixels of the edge of the detector or the user-specified region, then it is passed over as a non-star.

2) **Local maximum**: if a pixel is less than its eight surrounding (side- and corner-adjacent) pixels, then it is passed over as a non-star.

3) **`h < HMIN`**: measure the isolation index **`h`** (see **Section 3.1**) by going out radially from the pixel to determine how far out we have to go to find a greater pixel. If a pixel is not as isolated as the **`HMIN`** requirement says it needs to be, then pass it over.

4) **`f < FMIN`**: measure the local core flux **`f`** by finding the 2×2-pixel-aperture flux minus the local sky from the eight pixels surrounding this aperture. If this flux is less than the **`FMIN`** parameter, then pass this pixel over.

5) **`f < MASK(i,j)`**: If there is a mask generated to prevent features around the wings of saturated stars from being identified as stars (with the **`SHOW_MSK+`** option), then the local core flux **`f`** must be above the mask threshold, otherwise pass this pixel over.

6) **Outside of other pre-defined boundaries**: if a pixel is, say, within 5 pixels of the gap between chips in WFC3/UVIS or ACS/WFC, then pass the pixel over.

7) **`pix(i,j) < LOFLAG`**[12]: if the pixel is below the low-bad flag, then pass it over.

8) **`pix(i,j) > HIFLAG and DOSAT-`**: if the pixel is saturated, and the find-saturated-star flag is set to "**`-`**", then pass over the pixel over.

9) **`pix(i,j) > HIFLAG but not at saturation center`**: if the pixel is saturated and we are finding saturated stars, but the star is not at the center of the saturated distribution, then skip it.

10) **`Saturated, but too close to brighter stars:`** if the pixel is saturated, but there's a brighter saturated star within 4 pixels, then skip it, as it will be very hard to measure it well.

---

[11] Part of the philosophy here is not to find and measure every possible star, but just the stars that can be measured well. The name of the routine contains "one passs" for a reason; it is a one-pass finding routine. For programs that require multiple passes to find and measure stars in crowded environments, it will be necessary to make use of more sophisticated routines. Even so, the output of a one-pass routine is often needed to begin the multiple-pass process. This is particularly the case for undersampled imagers such as those on HST, since finding blended stars is hard to do without dithering, and accurate mappings are needed to simultaneously process dithered exposures. Conversely, it's also not the `hst1pass` philosophy to ensure that every source measured is indeed a star (though the `qfit` tests allow for this quite well). The author recommends finding more than is necessary then weeding out the non-sources in the collation step. This can be done when analyzing multiple `flt` images, but is harder to do when analyzing `drz` products.

[12] **`LOFLAG`** and **`HIFLAG`** are set based on the detector read in, but this can be over-ridden by the user on the command line.

11) **Unsaturated, but too close to saturated stars:** if the pixel is unsaturated, but it is within 5 pixels of a saturated star, then pass it over. It will be very hard to measure this star well

12) `q` > `QMAX`: if `q`, the quality of fit of the PSF to the star's inner 5×5 pixels (see §3.2) is too poor, then skip over this pixel. The default value for `QMAX` is 0.5.

13) `c < CMIN or c > CMAX`: If the PSF fit shows that the star is too sharp or too broad to be a star (see §3.2), then skip over this pixel. The default values for `CMIN` or `CMAX` are +0.1 and −1.

14) **Peaky and saturated**: If the pixel is saturated, but its surroundings are too centrally concentrated to be a real star, then it is likely a rare saturated cosmic ray. Reject it!

15) **Unsaturated star!** If it survives the fourteen tests and it is not saturated, then the pixel contains an unsaturated star.

16) **Saturated star!** If it survives the fourteen tests and is saturated, then the pixel contains a saturated star.

4.1 Photometry and Astrometry of Unsaturated Stars

Once the `hst1pass` routine has determined that a star may be in a pixel, then it proceeds to PSF fitting if it has access to a PSF. If there is no PSF supplied, then it uses a local centroid position and aperture photometry.

Since all the detectors accommodated by `hst1pass` are at least moderately undersampled, we adopt a 5×5-pixel aperture as the default. The goal is to find the triplet (x, y, z), where (x, y) is the raw location for the star in the pixel grid, and z is the flux.

We must first find the sky value, s. To do this, the routine starts by computing a rough centroid position for the star. It then iteratively finds the flux in the 5×5 aperture, and the sky from a robust 2.5-sigma-clipped average of the pixels in the annulus between r_1 and r_1+2 as follows. We start with $r_1 = 4$ and list all the pixels between r=4 and r=6. We use iterative sigma clipping to estimate the sky from the annulus. This allows us to estimate the total flux of the star from the sum over the inner 5×5 pixels and the PSF, which tells us what fraction of light should fall in that aperture given the star's center within the aperture. This flux estimate allows us to correct the pixels in the sky annulus for the star's flux, which improves our estimate of the sky and thus the star's flux. This is iterated a couple times to convergence on a consistent flux and sky. If the RMS in the sky is more than 1% of the star's flux, then we increase the r_1 by one and repeat until r_1=9 or the RMS condition is satisfied. Typically, faint stars are measured with sky between 4 and 6, and bright stars between 9 and 11.

With an estimate of the sky in hand, the routine turns to solving for a more accurate position. It solves for the position through a simple adaptive-mesh grid search. At each trial location (x_T, y_T), it determines a flux for the star by adding up the flux over sky in the aperture (which does not change during the procedure) and then adding up fraction of light that should land in each pixel according to the PSF and the location of the trial center. At each trial center, the star's flux

is simply[13]: $z_T = \Sigma_{AP} (P_{ij} - s_*) / (\Sigma_{AP} \psi_{ij})$, where the sum is over the 25-pixel aperture and the PSF value gives the fraction of light that should land in each pixel: $\psi_{ij} \equiv \psi(\Delta x, \Delta y) = \psi(i - x_T, j - y_T)$. With z_T, the flux at the trial location, we then evaluate an estimate for chi-square for this location from: $\chi^2 = \Sigma_{AP}(P_{ij}-s_*-z_T\psi_{ij})^2/\sigma_{ij}$. The σ_{ij} is the estimate of the error in the pixel value made up of a $\sqrt{P_{ij}}$ term from Poisson noise and a $0.01 \cdot P_{ij}$ term, which assumes a ~1% error in the PSF model. At each trial location, the routine determines χ^2 as described and adopts the (x_*, y_*, z_*) as the triplet that minimizes chi-squared. Once it has these optimized values, it determines an additional metric, the "quality of fit" which is simply q $\equiv \Sigma_{AP} \mid P_{ij}-s_*-z_T\psi_{ij} \mid /z_*$. This is often more useful than χ^2 at assessing departure from PSF shape, since it is less sensitive to the brightness of the source.

In addition to x_*, y_*, z_*, s_*, q_*, and χ^2_*, we also determine C$_*$, which reflects the normalized central-pixel residual for the star. This can help to determine whether a source is too sharp or too broad to be a star. If it is too sharp, it is likely a CR or hot pixel; if it is too broad it is likely a resolved object.   It is difficult to set hard and fast limits for q and C in terms of what is stellar and what is not. But it is easy to examine the run of magnitude against q and C for fields that have multiple stars in order to see what the trends of real stars look like. Such an inspection often clearly differentiates which sources are stars and which are not. See **Section 3.2** for an example.

Measurement of saturated stars is more complicated than for unsaturated stars, and the errors are often larger. **Appendix S** describes how the routine measures fluxes and positions for saturated stars and estimates their errors. **Appendix C** describes the table-based CTE correction that can be constructed for `_flt` images.

4.2   Coordinate and Photometric Systems

There are a few coordinate systems involved in the measurements `hst1pass` makes. It is beneficial to summarize them here. We present them in the order that they are computed.

1) `x, y`, and `m`: the raw chip-measured positions and fluxes

The first system is the raw coordinates measured for objects in the `flt/flc` images. extracted directly by fitting the PSF to the pixels in the images.

2) `X, Y`, and `M`: the CTE-corrected positions and fluxes

The next evolution is a possible table-based CTE correction (see previous section), which corrects the position (as of 2022, just the y-position) for the apparent shift away from the readout

---

[13] Note that this formula assumes that the noise in the aperture can be approximated as being entirely Poisson noise from the source. If the background noise is significant, then a more complicated formula is necessary, but any star that has a PSF shape generally dominates the flux in the small 5×5-pixel aperture so this simplification is usually justified for even moderately faint stars.

register that results when the electrons closer to the readout are preferentially trapped and those farther from the readout are preferentially shielded from traps.

3) `u`, `v`, and `w`: the distortion corrected positions

If there is a distortion solution provided in STDGDC format, the **XYM** measurements are corrected for distortion. Recall that the distortion solution used here is somewhat different from that in the pipeline IDCTABs. In some cases, they have different heritages, but in all cases the corrections used here have been rotated and scaled to be as close as possible to the original **xy** image frame. The photometric correction simply applies the pixel-area correction (PAC) such that the **w** photometry reports the same flux for the same source anywhere on the detector.

4) `U`, `V`, and `W`: the reference-frame positions

The next step in coordinate conversion is to convert the distortion-corrected measurements into a reference frame. When no world-coordinate system is provided via a command-line **WCS** parameter (see **Section 3.6**), the default reference frame places the target from the image header (as specified by the RA_TARG and DEC_TARG keywords) at coordinate (**U**,**V**) = $(0, 0)$, oriented with North up, East to the left, and with nominal plate-scale for the detector (for ACS 50 mas/pixel, for WFC3/UVIS 40 mas/pixel, and for WFC3/IR 100 mas/pixel). For convenience, this WCS is reported in the output file, for convenience. The default units for the **W** photometry is electrons per second. For ACS and WFC3/UVIS observations, this involves dividing the solved-for flux by the exposure time. For WFC3/IR, **W** is identical to **w**, since the IR flt images are up-the-ramp slope fits.

A WCS can be provided either explicitly or implicitly. See **Section 3.6** for details.

5) `r`, `d`: the header-based reference-frame positions

The lower-case RA and Dec specification comes directly from a mapping of the distortion-corrected coordinates `uv` into the absolute system specified by the WCS in the `flt`-image header. Note at this point, there is no absolute photometry available.

6) `R`, `D`: the GAIA-improved reference-frame positions (coming soon)

The GAIA DR3 database contains over 1.6 billion sources covering the entire 4π steradians of the sky. In its lifetime thus far, HST has covered perhaps 1% of the sky. So, it makes sense to distill the GAIA database into a small subset that can be used to astrometrically calibrate HST.

We have done this and have produced a file, gaia4hst.DB that contains about 21 million entries from the GAIA catalog. The file contains all fields observed with HST with WFPC2, ACS/WFC, ACS/HRC, WFC3/UVIS, and WFC3/IR. It also contains information from fields that are anticipated for observation in the current cycle (Cy 29, from the most recent paec file[14]).

---

[14] https://archive.stsci.edu/hst/paec.html, the planned and executed (exposures) catalog.

In the future, it will be possible for the user to supply a list of reference stars to use in the form of a text file with RA, Dec, V magnitude, and associated errors.

If a location of the file `gaia4hst.DB` is specified on the command line with a GAIADB= entry, then the routine will cross-match the **rd** coordinate list with the GAIA catalog, evaluated at the time of observation. It will then determine a 6-parameter linear mapping from **rd** to **RD**, taking projection into account, naturally, and will report those coordinates in the output file. It will also list the GAIA stars that overlap the field along with their identified HST counterparts in comment lines of the output file. *Note that when there are very few GAIA stars, the routine will do the best it can to provide a good calibration, but one should carefully look to make sure that the calibration makes sense.*

If the GAIA database is used to improve the absolute astrometry, then `hst1pass` will go back and modify the output **UV** coordinates to correspond to the **RD** estimates pushed into the output WCS frame (assuming it is not specified by **WCS=MAT**). One reason one might *not* want to do this is if the intent is to convert positions in an flt image into the drz frame of the stack of the visit, either by just referencing the header or by using `WCS=MAT`. Often `drz`-image headers are off by ~0.5 arcsecond from true absolute pointing. For some projects, it may be more important to push the coordinates into the specific `drz` frame than it is to have them absolutely calibrated.

It is possible to improve the absolute calibration of the `drz` frame first. To do so, one should run the routine with the **GAIADB** location set and with **GAIA2HDR+**, which will modify the header of a drz image to improve the astrometric solution. It can be convenient to run this at the beginning, before doing any artificial star tests.

Note that when information from the GAIA database is used in a paper, a reference needs to be made to the relevant GAIA consortium papers. See the GAIA website[15] for more details.

---

[15] **https://www.cosmos.esa.int/web/gaia/home**

5. Output Quantities

When the hst1pass routine is run without any command-line arguments, an info page is provided that contains the list of parameters that it can be run with, along with a list of data that can be output. For convenience, this output is provided in **Section 2.3**.

Here we will elaborate on the many quantities that can be provided as columns in the output file(s). A common set of output parameters is: **OUT=xympqXYMUVWZrd**. This includes all the quantities that hst2collate will expect. Below, we have divided the output parameters into four categories: finding, PSF-fitting, astrometry and photometry.

5.1 Finding Parameters

i: the local-max column number where the star was centered.

j: the local-max row number where the star was centered.

k: the chip the source was found in

I: same as i above, but it has a "I" affixed to it, and is zero-filled, for easy grepping.

J: same as j above, but it has a "J" affixed to it, and is zero-filled, for easy grepping.

K: same as k above, but it has a "K" affixed to it, for easy grepping.

h: the isolation index for the star (the distance to a brighter pixel, used to pass the HMIN criterion)

f: the value of the star's 2x2-pixel flux used in the finding

p: the value of the star's central pixel (without any sky subtraction)

n: the number of the star in the list; if the finding was done by a directed list or forced photometry, then this is the same as the number in the input list.

N: the number of the star in the list, zero padded and with "N" affixed for easy grepping.

5.2 PSF-Fitting Parameters

s: the sky value in the image (as the image was read-in)

S: the sky value after addition for any post-flash or dark-current that was subtracted by the pipeline. This is what is used to evaluate the CTE correction.

q: the "quality of fit" metric for the PSF fit. A value of zero signifies either a perfect fit, or that no fit was done. See **Section 3.2**.

c: a lower-case **c** represents the chi-squared value for the PSF fit. The numerical value of chi-square isn't the best way to examine star fits, since it is hard to have a perfect understanding of all sources of error. For this reason, bright stars tend to have large chi-square values and fainter stars asymptote to 1.0.

C: an upper-case-C represents the central-pixel residual. For values of **C** zero, the PSF was a good fit to the source's central pixel; if it is too positive, then the detection is likely to be a CR (cosmic ray) and not a star, and if it is too negative, then perhaps it is resolved. The residual is best examined in the context of other stars in the image.

P: if there is a PSF, this reports the fraction of a star's light that should land in a pixel that a star is centered on. It is an indication of the concentration of the PSF model.

e: generic error based on instrumental magnitude and instrument; in terms of raw coordinates/magnitudes.

5.3 Astrometry Parameters

x, **y**: raw chip-dependent *x* or *y* coordinate in the abutted frame. For WFC3/UVIS and ACS/WFC this means the bottom chip is from [1:4096,1:2048] and the top chop from [1:4096,2049:4096]. For WFPC2, it means the four chips fill the [1:1600,1:1600] abutted frame with each vertex in the lower left.

X, **Y**: for flt images, the upper-case versions of **x** and **y** report the CTE-corrected positions. As of mid-2022, there is no CTE correction for the x coordinate, but one may eventually become available.

u, **v**: the distortion-corrected position, from the STDGDC distortion solution applied to the **X**,**Y** coordinates. Again, the adopted STDGDC frame is constructed to be as close as possible to the original frame.

U, **V**: the upper-case versions of **u** and **v** report positions in the WCS-specified frame (see **Section 3.5**)

r, **d**: the RA and Dec, in degrees, from (**U**, **V**) and the WCS information in the primary image header.

R, **D**: the capitalized versions have RA and Dec as corrected by GAIA, if available.

5.4 Photometry

**m**: the raw instrumental magnitude measured using the standard 5×5-pixel aperture for PSF fitting or that from the aperture-photometry parameters.

M: the table-based CTE-corrected version of **m**. If no supplemental CTE correction must be performed (such as for a _drz or a _flc image), then **M** is identical to **m**.

w: lower-case w is M corrected for pixel area (if such a correction is possible/needed).

W: the capitalized version of **W** has been zero-pointed to agree with the reference frame, when the WCS is specified with a MAT file. If there is no such specification, it corresponds to flux per second rather than flux per exposure.

o: The flux in the aperture from other stars.

O: The flux in the aperture from other stars; conservatively treating every bump as a source.

It is possible to measure PSF-based fluxes for stars with apertures other than the standard 5×5-pixel aperture. Output-list parameters with numbers **1** through **9** give PSF-based aperture photometry (PBAP) with a diameter in pixels given by the number itself and a sky between 5.5 and 8.5 pixels (iteratively determined as described above). **Figure 3** below shows the set-up.

PSF-based aperture photometry involves measuring the flux in the aperture and dividing it by the fraction of the PSF that the model says should have landed within that aperture to arrive at the star's scaled flux. Note that if a star is a point source, then it should have the same instrumental magnitude for all apertures. If it contains contributions from something wider than a point source, then it will have more flux in the larger apertures than can be accounted for by a point source.

1: a super-local 1×1-pixel PBAP flux.

2: the very-local 2×2-pixel PBAP flux.

3: the somewhat local 3×3-pixel PBAP flux; a good choice extremely undersampled detectors, such as WFC3/IR.

4: the PBAP flux with an aperture diameter of 4 pixels.

5: the PBAP flux with an aperture diameter of 5 pixels.

6: the PBAP flux with an aperture diameter of 6 pixels.

7: the PBAP flux with an aperture diameter of 7 pixels.

8: the PBAP flux with an aperture diameter of 8 pixels.

9: the PBAP flux with an aperture diameter of 9 pixels.
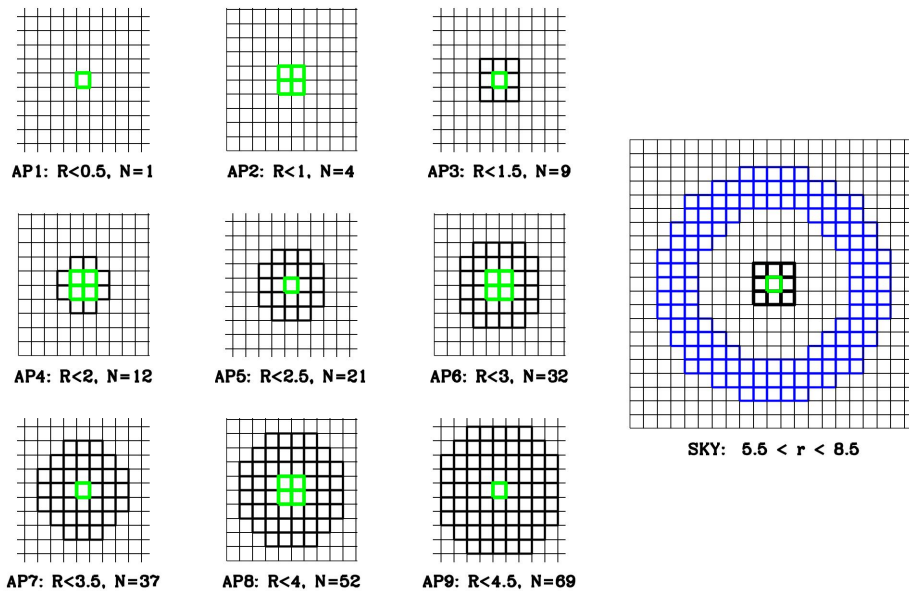
Figure 3: The apertures used for output options 1 through 9. The green box in the middle shows the pixels that the box is centered on. For odd apertures, it is centered on the single brightest pixel; for even ones, it's the brightest 2×2 box. (Right) the default sky aperture, just beyond AP9.

25

The default sky for this variable-aperture photometry is shown in the panel on the right in **Figure 3**. Since some of the star's PSF falls within this aperture, the routine iteratively solves for the 3×3 flux and star-subtracted sky. Note that it is possible to produce multiple output files from the same run, in the event that the number of output parameters in each file starts to become unwieldy. Future changes to make the number-based photometry more flexible, potentially with more local sky options, may be implemented. See **Appendix Z** for possible future bells and whistles.

5.5 Region Files

In addition to the **OUT=xym** type output, it is also possible to output a ds9-type region file. This allows immediate inspection of the stars that were found. One can specify: **REG=xy**, **REG=xy**, **REG=UV**, or **REG=rd** to get it to output the star list in various systems. In the region files, unsaturated stars are shown with a green circle and saturated stars are shown in red with a heavy circle. The size of the circle is proportional to the instrumental magnitude.

5.6 Additional Background Information in the Output Files

Each output file contains some basic information about the image and the finding procedure in lines preceded by a "#", such as useful keywords from the header (the filter, exposure time, date and time of exposure, dark current, post-flash, etc.). There is also information about the PSF (including perturbations), the saturated-star pre-processing, the finding statistics, the luminosity function of found stars, and possible GAIA matches. Some is just for general informational purposes, but some is used by subsequent processing steps (such as hst2collate).

6. Some Tips for Analysis

The `hst1pass` routine was designed primarily for differential astrometry and photometry. Of course, it is possible to calibrate the photometry both photometrically and astrometrically into an absolute system, but a lot of science can be done with differential measurements.

6.1 A Word about Differential Measurements

It is worth remembering that the differential measurements with HST will always be more accurate than absolute measurements. As a consequence, it is often better to combine measurements differentially and only then to do the final calibration. If each exposure catalog is calibrated individually then the results combined, the residuals will show not only the measurement errors, but also the calibration errors, which could hide the true data quality.

For differential photometry, one can reference a set of dithered images relative to the first exposure to construct CMDs to evaluate scientific results long before they are calibrated. It is the author's preference to work in "instrumental" units as long as possible, so as to enable an examination of any systematic trends in the natural units of those trends.

For differential astrometry, one can transform the measurements in various exposures into a common frame, e.g. using for this the distortion-corrected frame based on the first long exposure. This allows observations to be kept in pixel units aligned with the axes of the detector for as long as possible, which can be useful for examining unexpected trends (such as CTE errors or distortion errors).

6.2 A Word about Photometric Calibration

The photometry produced by `hst1pass` is deliberately left uncalibrated for several reasons. One is that the closer we keep the photometry to the detector units (electrons, etc.), the easier it is to evaluate errors and determine whether results are consistent with expectations. Of course, `hst1pass` naturally carries around several different versions of the photometry (`m`, `M`, `w`, and `W`), so it shouldn't be so difficult to carry around something calibrated. So, there must be another reason. There are a few.

First, users should do their own photometric calibration so that they are aware of its errors and limitations. A "photons to ApJ" one-stop shop is very difficult to pull off and usually requires large teams to continuously validate. Furthermore, the absolute photometric calibrations improve and evolve continually, so it would require regular maintenance. For all these reasons, formal photometric calibration must be done after the `hst1pass` analysis. See the instrument pages for instructions on how to calibrate HST data. This can generally be done with either the `flt`/`flc` images or with `drz` images.

6.3 A Word about Astrometric Calibration

The GAIA database makes it much easier than ever before to do astrometric calibration. There will soon be option `GAIADB` on the `hst1pass` command line to do absolute calibration based on the GAIA DR3 catalog, but the quality of that calibration depends on the quality of the match to image stars, which is hard to guarantee in an automated routine. Note, too, that the best-

measured stars in the GAIA catalog are often saturated in HST images. Similarly, the best-measured stars by typical HST images often have large measurement errors in GAIA, or are too faint to be included in the catalog. Therefore, it is often a challenge to do an absolute calibration with the best precision of either catalog.

6.4 A Word about Astrometric Transformations

It is worth considering that—as with photometry—astrometric measurements are much more accurate in a differential sense than they are in an absolute sense. And relative measurements of stars that are close to each other are more accurate than measurements of stars on different sides of a detector, or even worse, on different sides of a mosaicked field. For this reason, when making a critically careful differential measurement, it is important to consider carefully which stars to use in the transformation. Often, basing the transformations on a local set of stars of similar brightness to the target will result in the best measurement. In this say, many CTE-related issues can drop out.

In **Appendix D**, we evaluate the quality of the distortion solutions. Based on average-RMS errors, they are generally accurate and stable (in a non-linear sense) to better than 0.02 pixel, but users should always look at residuals and make sure that there are no unanticipated sources of error.

We strongly recommend using full 6-parameter linear transformations when converting positions from one distortion-corrected frame to another (see **Appendix T**). The basis of these transformations is a set of associated coordinates for N stars, with one set in the starting frame (x_1, y_1) and another set in the destination frame (x_2, y_2). Choosing which stars to use for transformations, and determining whether individual stars should be rejected to optimize the transformations is very much an artform. It is beyond the scope of this document to delve into these details, but suffice it to say it is worth putting considerable attention into the stars you use as a reference. On a final note, the author finds that weighting stars used in the transformation can often cause more trouble than it's worth; it is best to find a set of stars that you can presume to be equally well measured in each frame[16], then evaluate which stars have consistent positions in the two frames such that they can be used to inform us about the transformations[17]. One can then reject—one by one or by some sigma-clipping algorithm—stars that are inconsistent, thereby iteratively homing in on the optimal transformations.

---

[16] Often the brightest 3-4 magnitudes of unsaturated stars.
[17] The position residual is simply x2-x2e, where x2e comes from the transformed position of (x1,x2) into Frame#2.

7. Some Use Cases

The USE_CASE directory on the `hst1pass` server site (see **Section 2**) has several sub-directories that provide specific examples. We do not supply all the images used here; only the output files. Users can download them separately from MAST, if they desire to repeat the use cases for themselves.

- USE CASE #1: Using a perturbation PSF, examining subtracted images
- USE CASE #2: Finding the stars in a UDF image
- USE CASE #3: Measuring CMD photometry of an ultra-faint dwarf, with AS tests
- USE CASE #4: Examining a time-series with `hst2collate`
- USE CASE #5: Generating a mask for saturated stars

8. Final Notes

8.1 Disclaimer

While the `hst1pass` routine has undergone significant testing over its many years of use and development, many of its capabilities have been added recently and as such have yet to be thoroughly tested. Thus, please don't trust the code blindly, but examine its output, refer to this manual and the `FAQ.txt` file, perform checks and cross-checks, and investigate whether the results make sense. If, after trying all the suggestions in the FAQ file, results still do no not make sense, then feel free to contact help@stsci.edu.

Also, if there are things you like or don't like about the routine, please bring these to our attention as well. If you think of ways in which it could be improved, then share that as well. If you read over the list of possible future bells and whistles in **Appendix Z** and find that one of those tools would be especially useful to your science, please let us know that, too, along with, perhaps a specific use-case. All of this will help to prioritize any future development.

8.2 Code Updates

Potential future improvements are discussed in Appendix Z. The author will update the `NEWS.txt` file with information on updates and fixes in-the-works. The website will also contain a living version of this document, with the most up-to-date information.

Acknowledgements

While all of this code has been written by the author (except the USNO routines to include parallax in the GAIA database and the quicksort routine, which was found on the internet and has been credited), it has not been written in a vacuum. It has been developed over the years in consultation with a great many collaborators who have found bugs, suggested features, and provided encouragement and myriad tests. If you are one of the many I "trained" to use this suite of tools, thank you for being patient, asking questions, and helping me know what others need to know before they can use it fruitfully.

In the past when I have trained people to use the software, it has been for a specific scientific purpose. Here, in this manual, I have provided more general instruction and, surely, I have left some important things out. So, I would also like to acknowledge my need for advice in improvement going forward. PSF-fitting on HST images is a powerful tool. It has taken this long to get a general tool out because there are many things that have to be done simultaneously to get any useful result out. This is my first attempt to put everything together and it will take time and input to make it a more streamlined package.

I am particularly grateful to Andrea Bellini and Mattia Libralato for their thorough reading of this document and some limited testing of the latest version of the routine. It has enabled me to avoid some certain issues before release.

9. References

Anderson, J. & King, I. R. 1999 PASP 111 1095: *Astrometric and Photometric Corrections for the 34th Row Error in HST's WFPC2 Camera*

Anderson, J. & King, I. R. 2000 PASP 112 1360: *Toward High-Precision Astrometry with WFPC2. I. Deriving an Accurate Point-Spread Function*

Anderson, J. & King, I. R. ACS/ISR 2004-15: *Multi-filter PSFs and Distortion Corrections for the HRC*

Anderson, J. & King, I. R. 2003 PASP 115 113: *An Improved Distortion Solution for the Hubble Space Telescope's WFPC2*

Anderson, J & King, I. R. ACS/ISR 2006-01: *PSFs, Photometry, and Astrometry for the ACS/WFC*

Anderson, J. ACS/ISR 2007-08: *Variation of the Distortion Solution of the WFC*

Anderson, J. & King, I. R. 2008 AJ 135 2114: *Deep Advanced Camera for Surveys Imaging in the Globular Cluster NGC 6397: Reduction Methods*

Anderson, J. & Bedin, L. R. 2010 PASP 122 1035: *An Empirical Pixel-Based Correction for Imperfect CTE. I. HST's Advanced Camera for Surveys*

Anderson, J. WFC3/ISR 2014-02: *The Impact of x-CTE in the WFC3/UVIS detector on Astrometry*

Anderson, J. WFC3/ISR 2014-24: *Local Bundles: Bringing the Pixels to the People*

Anderson, J. WFC3/ISR 2016-12: *Empirical Models for WFC3/IR PSF*

Anderson, J. WFC3/ISR 2018-14: *Focus-Diverse PSFs for Five Commonly Used WFC3/UVIS Filters*

Anderson, J. & Bedin, L. R. 2017 MNRAS 470 948: *The M4 Core Project with HST – V. Characterizing the PSFs of WFC3/UVIS by focus*

Anderson, J. & Sabbi, E. WFC3/ISR 2018-07: *Impact of Increased Jitter on WFC3/UVIS PSFs*

Anderson, J. WFC3/ISR 2021-13: *Table Based CTE Corrections for flt-format WFC3/UVIS*

Bajaj, V. Python notebooks, in prep.

Bellini, A., Anderson, J. & Bedin, L. R. 2011 PASP 123 622: *Astrometry and Photometry with HST WFC3. II. Improved Geometric Distortion Corrections for 10 Filters of the UVIS Channel*

Bellini, A. et al. ACS/ISR 2018-08: *Focus-diverse, Empirical PSF Models for the ACS/WFC*

Bradley, L. et al. 2020 (https://doi.org/10.5281/zendo.4044744): *astropy/photutils: 1.0.0*

Casetti-Dinescu, D. I. et al. 2021 PASP 133 064505 *A Comprehensive Astrometric Calibration of HST's WFPC2. 1. Distortion Mapping*

Chiaberge, M. ACS/ISR 2012-05: *A New Accurate CTE Photometric Correction Formula for ACS/WFC*

Di Nino, D. et al. ACS/ISR 2008-03: *HST Focus Variations with Temperature*

Fruchter, A. S. & Hook, R. N. 2002 PASP 114 144: *Drizzle: A Method for the Linear Reconstruction of Undersampled Images*

Gaia Collaboration (Prusti, T., et al.) 2016 A&A 595, A1: *The Gaia Mission*

Gaia Collaboration (Brown, A. G. A., et al.) 2021, A&A 649, A1: *GAIA Early Data Release 3: Summary of the Contents and Survey Properties*

Gilliland, R. L. ACS/ISR 2004-01: *ACS CCD Gains, Full-Well Depths, and Linearity up to and Beyond Saturation*

Gilliland, R. L., et al. WFC3/ISR 2010-10: *WFC3 UVIS Full-Well Depths, and Lineary Near and Beyond Saturation*

Hoffman, S. L. & Anderson, J. ACS/ISR 2017-08: *A Study of PSF Models for ACS/WFC*

Hoffman, S. L. & Kozhurina-Platais, V. ACS/ISR 2020-09: *Validation of New ACS/WFC Geometric Distortion Reference Files Derived using Gaia DR2*

Kuhn, B. & Anderson, J. WFC3/ISR 2021-06: *WFC3/UVIS: New FLC External CTE Monitoring 2009 – 2020*

Kozhurina-Platais, V. et al. ISR-2009-33: *WFC3 SMOV Proposal 11444- UVIS Geometric Distortion Calibration"*

Kozhurina-Platains ISR 2009-34: *WFC3 SMOV PROPOSAL 14445 IR Geometric Distortion Calibration*

Kozhurina-Platais ISR 2015-06:   *ACS/WFC Revised Geometric Distortion for* `DrizzlePac`

Kozhurina-Platais, V. WFC3/ISR 2014-12: *Astrometric Correction for WFC3/UVIS Filter-Dependent Component of Distortion*

Noeske, K, et al. WFC3/ISR 2012-09: *WFC3 UVIS Charge Transfer Efficiency October 2009 to October 2011*

Sabbi, E. & Bellini, A. WFC3/ISR 2013-11: *UVIS PSF Spatial & Temporal Variations*

Sahu, K. C. et al. ACS/ISR 2007-12: *ACS PSF Variations with Temperature*

Schechter, P. L., Mateo, M., & Saha, A. 1993 PASP 105 1342: *DoPHOT, A CCD Photometry Program: Description and Tests*

Stetson, P. B. 1987 PASP 99 191: *DAOPHOT: A Computer Program for Crowded-Field Stellar Photometry*

Ubeda, L & Kozhurina-Platais, V. ACS/ISR 2013-03: *ACS/WFC Geometric Distortion: a Time Dependency Study*

Appendix C: Automated Table-Based CTE Correction

The `hst1pass` program has the capability of correcting some photometric and astrometric measurements for CTE loss, particularly for `_flt` images from either WFC3/UVIS or ACS/WFC. This is done by means of a look-up table that quantifies the losses as a function of star brightness and local background level. A recent ISR by Anderson (2021) details how these tables have been constructed for WFC3/UVIS. The WFC/ACS astrometric and photometric table-based corrections come from tables constructed from a similar procedure (see Anderson in prep).

Note there already exists a formula-based photometric correction for ACS images constructed and maintained by Chiaberge (2012), and these corrections have been shown to work quite well. However, these corrections were designed for aperture photometry with larger apertures, and, more importantly, there is no correction provided for astrometry. For this reason, during the process of implementing the astrometric correction, a `hst1pass`-tailored photometric correction was constructed as well that corresponded more closely with the small 5×5-pixel aperture photometry done by `hst1pass`. Of course, `hst1pass` can also work on the pixel-by-pixel CTE-corrected pipeline products produced by the pipeline (the `_flc` images). Such images do not require the table-based correction so that step is suppressed.

C.1 The Need for the Total Background

By default, the `_flt` images from the standard calibration pipeline have post-flash and dark-current subtracted from them (`FLSHCORR = 'PERFORM'` or `'COMPLETE'`), so that the background will resemble as closely as possible the natural astronomical background. The `hst1pass` estimates the number of electrons that were subtracted during the post-flash and dark-current correction and determines the operative raw-electron sky level required to determine the table-based CTE correction.

C.2 Implementing the Table-Based CTE Corrections

The empirical CTE corrections to WFC3/UVIS `_flt` measurements based on PSF photometry with `hst1pass` were presented in Anderson 2021. The corrections have a photometric and an astrometric component. Post-measurement corrections such as these have been available for a long time for photometry (see Noeske et al. 2013 and Kuhn et al. 2021), but little has been available for astrometry; obsevers have been dependent on the calibration-pipeline pixel-based correction (`_flc`) for that.

The WFC3/UVIS corrections in Anderson (2021) are in the form of two-dimensional tables that report the observed CTE losses as a function of (1) true stellar flux (in instrumental magnitudes) and (2) background sky level. The brightness is tabulated every magnitude level and the background level at a few fiducial levels: 0 e$^-$, 8 e$^-$, 12 e$^-$, 16 e$^-$, 20 e$^-$, 25 e$^-$, 30 e$^-$, and 35 e$^-$. The spacing of the table entries is close enough that the corrections can be interpolated linearly in background level and instrumental magnitude. The astrometric corrections are similar; the astrometric tables provide the shift away from the readout direction, which is in the direction of the chip gap for WFC3/UVIS and ACS/WFC.

The tables provide the correction at a reference location on the detector ($j_{PAR} = 2048$) and at reference time (2021.1 for WFC3/UVIS and 2020.9 for ACS/WFC). The parameter j_{PAR} corresponds to the number of parallel shifts to the serial register, and parameter D is the date in fractional years. The table corrections should be scaled by:

$$f = \left(\frac{j_{PAR}}{2048}\right)\left(\frac{D - 2021.1}{2021.1 - 2009.4}\right)$$

for WFC3/UVIS, where 2009.4 corresponds to the installation date and 2021.1 to the model-pinning date. For ACS/WFC, these numbers would be 2002.4 and 2020.9, respectively. Our goal is to construct similar tables for ACS/HRC and WFPC2 in the future, though it is more complicated constructing CTE corrections for legacy instruments, as we must rely on data in the archive.

Note that the table entries are provided in terms of the *true* stellar instrumental magnitude, while `hst1pass` gives us the *apparent* instrumental magnitude, m_0. The reason for this is that when we constructed the table, we measured the flux decrement relative to known flux in calibration images. The routine iterates to determine the true flux.[18]

C.3 Outputting the CTE-Corrected Measurements

Users can choose to output the CTE-corrected photometry (when it is available) by selecting the capital-letter output columns: `X`, `Y`, and `M` represent the CTE-corrected measurements for the raw measurements `x`, `y`, and `m`. Note there is no correction available yet for X-CTE (see Anderson 2014), so `x` and `X` will return the same number, see **Appendix Z**. Finally, `s` reports the sky as measured in the pipeline-corrected `_flt` or `_flc` image, and `S` reports the sky that was used for the table-based CTE correction (i.e., with the post-flash and dark current added back in).

Note that the downstream coordinates (`u`, `u`, `U`, `V`, `r`, `d`, and `R`, `D`; see below) provide only the CTE-corrected measurements (if available). Users interested in inspecting what the CTE correction was for a general observation simply need to examine `M−m` for the photometric correction and `Y−y` for the astrometric correction.

---

[18] To do this, we first adopt the observed flux as the first guess of the true flux and interpolate the table at (m_0, S) and multiply by the scale factor f (above) to get Δm_0 and Δy_0. Subtracting Δm_0 from our observed magnitude m_0 provides us our first estimate of the true flux: $m_1 = m_0 - \Delta m_0$. We then find $f \times T_{\Delta m}[m_1, S]$ to obtain Δm_1 and Δy_1. We compute $m_2 = m_0 - \Delta m_1$, and again evaluate $f \times T_{\Delta m}[m_2, S]$ to determine Δm_2 and Δy_2. If $\Delta m_2 - \Delta m_1$ is less than 0.005, then we consider it converged. If not, we compute m_3, etc, until convergence is reached at the Nth iteration. The true flux estimate is $m_N = m_0 + \Delta m_N$ and the true position is $y_N = y_0 + T_{\Delta y}[m_N, S]$.
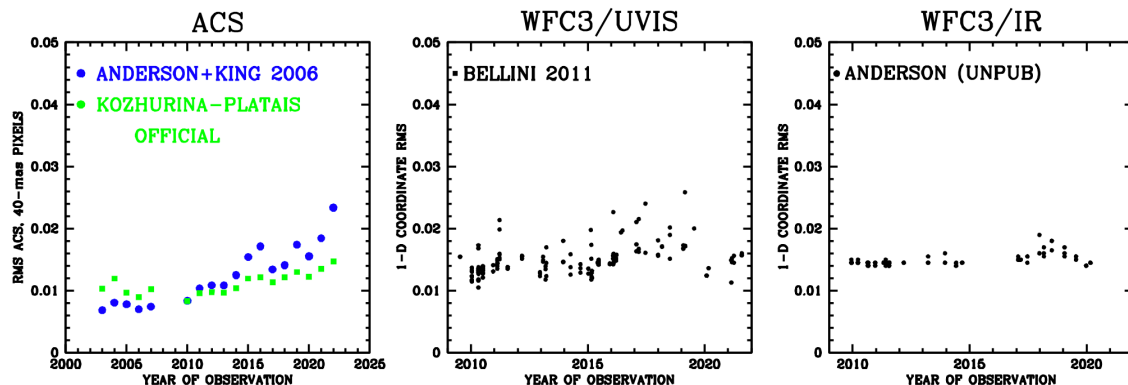
Appendix D: Distortion Solutions

The distortion solutions available for use by `hst1pass` are in the STDGDC format (see **Appendix G** below for a description) and are generally good to about 0.02 pixel. We recommend using full 6-parameter linear transformations when comparing positions in different distortion-corrected frames. In principle, distortion-corrected frames should differ only by a "conformal" 4-parameter linear transformation involving an offset (in x and y), a rotation, and a scale change. Allowing the off-axis linear "skew" terms as well allows for the fact that some detectors have linear terms that change systematically over time (ACS) and others vary with breathing.

For WFC3/UVIS we provide the distortion solution constructed by Bellini et al (2012). For WFC3/IR we provide an unpublished solution constructed by the author. For ACS/WFC, we provide the solution constructed by Kozhurina-Platais (2014) for the ACS pipeline, as that solution is better over all time than the solution constructed by Anderson & King (2004) soon after installation of ACS in 2002.

To evaluate the distortion solution, we need a moderately populated starfield that has been observed at multiple offsets and orientations over the lifetime of the detectors. All the major detectors have observed the center of Omega Cen, which serves this purpose well. One caveat with Omega Cen, though, is that the stars move almost 1 mas/yr, so if motions are not taken into account, one must compare positions in images taken within a year of each other to avoid having internal motions artificially increase the residuals.

In **Figure D**, for the three detectors, we compare every pair of images that were (1) taken within a year in time of each other and (2) offset by more than a third of a chip or with relative orientations greater than 45 degrees (so that the images probe different parts of the distortion solution). There were over 100 images taken of the center of Omega Cen for each detector, thus we could cross-compare many pairs for each detector. For each image, we compared its distortion corrected star positions against every good comparison image and constructed a 6-parameter linear transformation and examined the residuals. The 1-D single-exposure coordinate error is one half the 2-D residual, one factor of $1/\sqrt{2}$ comes from the conversion to 1-D and the other $1/\sqrt{2}$ because there is distortion in both systems. Of course, there is measurement error included as well, but these residuals should be roughly indicative of the distortion errors.

Figure D: RMS distortion residuals in detector pixels for the three main imagers. These RMSs represent averages over the detector; the errors tend to be larger at the detector edges.

Appendix G: The FITS-Format STDGDC File

The distortion solutions described in the Instrument Handbooks and provided on the Reference File page for each instrument are designed to map the detector pixels into the V2-V3 telescope plane for the purposes of absolute astrometric calibration based on engineering data. 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 pipeline solution, these solutions are constructed from polynomials plus various look-up tables to account for detector artifacts or filter artifacts. 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.). As a consequence, we adopted a format for the distortion solution that makes both directions as easy and as fast as possible.   The "standardized" format (called STDGDC, from standard geometric distortion correction) provides a similar interface for all detectors. The solution is provided in a multi-extension FITS file, where each extension is an image.

The first extension is the size of the detector (1014×1014 pixels for WFC3/IR, 4096×4096 for WFC3/UVIS and ACW/WFC). The value of each pixel in the first-extension image is the x coordinate of that pixel's mapping into the distortion-corrected reference frame. Similarly, the second-extension image contains 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 obtain the distortion correction at any non-integer location in the pixel grid. There is no need for higher-order interpolation of the pixel grid, since it is tabulated every pixel.

The third extension, also the size of the original detector, provides 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 the result is that flux is not preserved. The pixel-area correction must be added to the `flt`/`flc`-measured magnitudes.

The fourth and fifth extensions provide the inverse distortion solution. These images are 1148×1015 pixels in size[19] 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-

---

[19] Note that the vertical dimension here of 1015 is very close to the original 1014-pixel size of the detector. This is related to the fact that the free parameters in the solution are chosen to orient the corrected frame with y and its pixel scale.

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 have negative pixel locations in an image, so we use the LTV1 and LTV2 image-header 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 significantly faster than inverting a polynomial and several other layers of solution.

Appendix H: Limited-Use Early Version of `hst2collate`

The `hst1pass` routine was envisioned to be the first part of a suite of routines that could do PSF-based HST analysis. The second routine in the suite would be hst2collate. Its goal is to cross-identify stars in output lists from `hst1pass` and distill the lists into common catalogs.

H.1 The Current Limited Functionality of hst2collate

The purpose for `hst2collate` is to allow curation of a catalog in a wide variety of situations. Basically, it will take a set of star lists from a number of images and it will (1) find the transformation from each frame into a (specified) reference frame and (2) collate together the stars into the reference frame in order to produce an average catalog of sources. The ultimate goal is to allow this to be done in a flexible reference frame, but for the purpose of some basic capability now, we decided to streamline the functionality so that the `hst2collate` tool could be used in concert with `hst1pass`. In this Appendix, we detail how to use it in its limited functionality, then describe some possible future functionality.

One reason it has been easier to do astronomical analysis on the `drz` images is because it's an enormous amount of work to take model PSFs and use them to measure stars in `flt` or `flc` images, and then collate together lists from different distorted frames into a common reference frame, all of which requires access to the forward and reverse distortion solutions. The `hst1pass` routine addresses only the first part: measuring stars in images. The next step requires combining and distilling various unordered lists of stars into a single catalog. This is the most basic function of `hst2collate`. The combination of the `hst1pass` and `hst2collate` should make possible a large number of analyses.

Another application that `hst2collate` can address is artificial star tests, which serve several purposes in astrometry. They can serve to show what kinds of measurement errors (systematic and random) are involved in measuring real stars. They can serve to show what fraction of stars slip through our finding algorithms, and thus provide completeness tests. With accurate PSF models, precise positions and accurate distortion solutions, it is possible to insert realistic artificial stars into consistent locations in the `flt/flc` images with high fidelity such that they can be drizzled in a manner that is identical to that for real stars. This will allow users to demonstrate for themselves how pitfall-fraught it is to do precise PSF analysis on drizzled images, since the star profiles vary considerably with pixel phase and the exact location of the input pixels relative to output pixels.

Finally, if internal proper motions between epochs can be ignored in the collation phase, it is possible to use `hst2collate` to relate images taken at very different times, thus enabling time-series analyses. One can even use the results to examine proper motions, though care must be taken in regards to the motion of the reference frame and aspects of the motions that could be degenerate with the transformations.

H.2 Input Parameters for `hst2collate`

This routine currently accepts only a few input parameters. It takes one reference-frame specification (i.e., **REF=iassocn_drz.UVW**), which would come, e,g,, from a run of

`hst1pass` on a drizzled association product with aperture photometry. In the future versions of the code, the reference frame will be specifiable by other means (such as RA and Dec), but at the moment, it requires the reference frame and the contributing frames to both have **UVW** coordinates in roughly the same reference frame.

The routine then takes a list of `hst1pass` output files from `flt/flc`-type images. These files can have many columns, but they must at the very least have **U** and **V** (where **U** and **V** are in the same WCS reference frame as the reference list) and **u**, **v** and **w**, the distortion-corrected positions and instrumental magnitudes. The `hst1pass` output files also contain quite a bit of information about the images themselves, the coordinate range of good pixels, the exposure time, the date of exposure, the distortion solution used, etc., some of which the `hst2collate` routine uses as well.

With no other parameters or flags set, the routine will read in the reference list of stars and each of the input lists of stars. It will cross-identify stars in the reference list and the individual star lists and use them to determine the transformation from the **uv** frame of each exposure into the **UV** reference frame so that the stars will be mapped to the exact same pixel space (frame) as the measured reference stars. Dithering and *ab initio* WCS information is not always perfect, so this allows the exposures to be mapped as closely as possible to a known master frame.

Although we used the positions of stars in the reference frame for the transformations, centroid positions measured in drizzled images are not as accurate as those in `flt/flc` images made with accurate PSF models, so we use them only for the initial mapping. In addition, the star finding in `drz` images is sometimes not as robust at in a `flt/flc` image, since there can be transformation issues into the drizzle frame that blurs it out and/or introduces artifacts.

Therefore, after this initial mapping, `hst2collate` reconstructs the master-frame catalog using the individual exposure lists. Assuming that no additional runtime parameters are set, the routine selects as valid catalog sources all detections that are found in at least 50% of the possible images. It then determines the master-frame predicted position for that source for each image where it was found and constructs an average **UV** position and flux for it (by default in the **W** photometric system). In the next step, it iterates, taking the average **UV** position and average **W** flux, re-determines the transformations, then re-determines the averages and standard deviations about those averages. In this way, the system is tied to the original frame, but the transformations take advantage of the higher precision of the individual measurements. **USE CASE #3** shows an example of the standard deviation for **U**, **V**, and **W** from a set of exposures. The typical RMS error in measuring a stellar position in the `flt/flc` images is about 0.01 pixel and 0.01 magnitude for the brighter stars, with errors that go up as the S/N goes below 100.

The basic output of `hst2collate` is a list of the catalog stars, along with their average positions and the standard deviations about those averages and the `iabcdefgq_mat.UVuvWw` files that provide an implicit way to map the positions in the individual frames into the reference frame. As the suffix suggests, the file consists of a list of stars for which we have coordinates in the reference frame and in the distortion-corrected frame of the particular exposure. These "point associations" provide an implicit way to construct a linear transformation from frame to frame. The stars listed in the mat file have been selected to have consistent positions in the two frames; stars with inconsistent positions are removed from the list. At this point, all positions in the list are considered to have the same error, but a future version may take into account the

position quality of each association. These _mat files can then be used in hst1pass if one wants to inject artificial stars at consistent locations in the reference frame.

H.3 Additional Outputs from `hst2collate`

Although future enhancements are envisioned for hst2collate, it already has some useful capabilities. Some of the results (with hst2collate in the name) are just documentation of the steps it moves through in the process described above and these will be described in a later report. The main output of interest is hst2collate.07.avgphot2, which contains the final average positions and fluxes in the reference frame. Additional output files can be requested via additional flags on the command line:

PKMAP+ generates several FITS output files of note. The hst2collate.03.cvg_map.-fits image indicates how many of the input exposures overlap the reference frame at every particular location in the reference frame, as determined from the "boundary" information in the hst1pass file. The other images *pk?_map.fits indicate how many stars were found in each pixel. The #1 file maps one star to one pixel, the #2 file maps one star to 2×2 pixels, and the #3 file is a 3×3 convolution of the #1 file. These maps are used to determine how many times a given star in the field was found. Stars that are found in a minimum number of images in a minimum fraction of the total possible (**cvg**) are recorded in the final catalog.

LNK+ generates a _lnk file (in addition to the _mat file) for each input frame. The _lnk file has as a suffix `.UVWuvXY_????`, where **UVW** correspond to the position from the input frame mapped into the reference frame. The **uv** and **XY** correspond to the average **UV** position mapped in to the distortion corrected and raw image frame. The next column is the catalog number and the remaining columns are repeats of the corresponding line from the hst1pass output file for that image. These linked files allow one to examine time-series photometry and astrometry, among other things.

QBAR+ will have hst2collate search for **q** in the input files and will output that along with the other parameters in the average.

I+ will have hst2collate skip the new-catalog construction; it will use the catalog in the **REF** file. The positions of the stars can still be improved, but stars will not be added or subtracted, so that the numbering and ordering is preserved.

ZPU specifies how to set the zeropoint for the W photometric system.   For example, the command **ZPU=EXPT100** will normalize the photometry to a 100-second exposure time.

H.4 A Vision for Future Routines

Currently hst2collate operates on only one filter at a time[20]. In the future, it will have hooks to deal with multiple filters at the same time, for both finding and for collating.

---

[20] See **USE CASE #3**, though, for an example of how to generate CMDs with hst2collate.

Also, currently, it must have access to a specific `_drz` frame as a touchstone for the matching. In principle, that shouldn't be necessary. It should be possible to take in RA and Dec positions from the `hst1pass` output files and collate things that way into an arbitrary frame. That won't take too much work to implement and will allow much larger mosaics.

A third routine `hstNstack` could stack images into frames in a manner similar to `astrodrizzle`, but using an iterative approach and making use of the transformations developed in the MAT files. This would "liberate" the collation phase from requiring a specific drizzled image to map into. The goal of such a stack would not be high-precision analysis, but merely as a visualization tool. This routine could also construct raw "peak" maps (see Anderson & King 2008), which can be useful for faint-object detection. Finally, an eventual fifth routine could perform simultaneous multi-passs finding and photometry, appropriate for fields that are more crowded than is optimal for `hst1pass`.

Please e-mail help@stsci.edu for any feedback on these or other potentially helpful routines.

Appendix P: PSF Optimization

The `hst1pass` routine runs quite well with simple library PSFs available in the PSFs directory of the website. While it is possible to improve the PSF either via perturbation or via solving for focus, differential photometry and differential astrometry is not generally significantly improved. We recommend that users start with the simplest possible analysis, and only then include more sophisticated treatments if the examined analysis warrants. The main impact of improved PSFs is in absolute photometry.

The primary cause of PSF variation comes from telescope breathing. As HST crosses into and out of the earth's shadow or changes its orientation with respect to the Sun (thereby intercepting more/less insolation), it heats up and cools down the truss, and HST's focal length varies accordingly. Engineering models for breathing exist, but they are far from perfect. The main impact of breathing is to vary the fraction of a star's flux that lands in the core; the shape of the core does not change dramatically and thus there is only a small impact on astrometry. Breathing generates a larger impact on photometry. The zeropoint of one exposure relative to another similar exposure can vary by ±0.03 magnitude due to breathing effects[21].   As a consequence, when combining photometric observations, it is often useful to allow a zeropoint shift from exposure to exposure. Use of a perturbation PSF or a focus-diverse PSF often reduces the zeropoint variation to ±0.005 magnitude, though it does not improve differential photometry and astrometry significantly.

Most observations also contain some amount of nominal tracking jitter, which can be analyzed with the nominal PSFs. Of course, when the jitter is unusually large, it can affect the PSFs. When a perturbation PSF is determined, it often shows a deficit at the center and increase in flux in lobes on two sides of the center. Perturbing such a PSF can often improve differential photometry up to a point, but if the jitter is too large, it may require a different kind of PSF analysis than `hst1pass` can provide.

There are three main things a PSF helps measure: astrometry, photometry, and stellarity (i.e., quality of fit). Astrometry should largely be the same whether one uses the unchanged STDPSF, whether one uses a perturbation, or whether one allows for focus variations with STDPBF. Photometry and quality-of-fit can be affected by the choice of PSF.

---

[21] Note that the shift brightness is a function of the size of the aperture. The `hst1pass` routine uses a 5×5-pixel aperture. For stars measured with a 1×1-pixel aperture, the zeropoint variation would be closer ±0.2 magnitude. If we were using a 5-pixel-radius aperture, then the zeropoint variation would be below 0.005. Relatively large, e.g. 10 pixels in radius, are largely immune to breathing variations, however, difficult to use in crowded fields.

To evaluate measurement quality based on different PSF treatments, we reduced 170 WFC3/UVIS exposures taken through F606W at the center of NGC5139, Omega Centauri. We concentrate here on the brightest three magnitudes of unsaturated stars so that we can examine PSF issues without worrying about S/N. We collated together the star lists from the 170 images and constructed an average F606W magnitude for each star referencing the catalog to a common exposure time of 60s. We then examine how the measurements of the stars in each exposure varied with respect to these average values.

Figure P.1 contains three rows of panels. In the top row, we analyzed the photometry extracted from the plain STDPSF reduction. Recall that the STDPSF accounts for spatial variations in the fluxes, but it does not account for any dynamical effects. The middle panel shows the residuals when a single perturbation across the images is used on the STDPSF. Finally, the bottom panel shows the residuals for the reduction with the STDPBF, where the focus was determined for each exposure and then the appropriate-focus PSF was used.
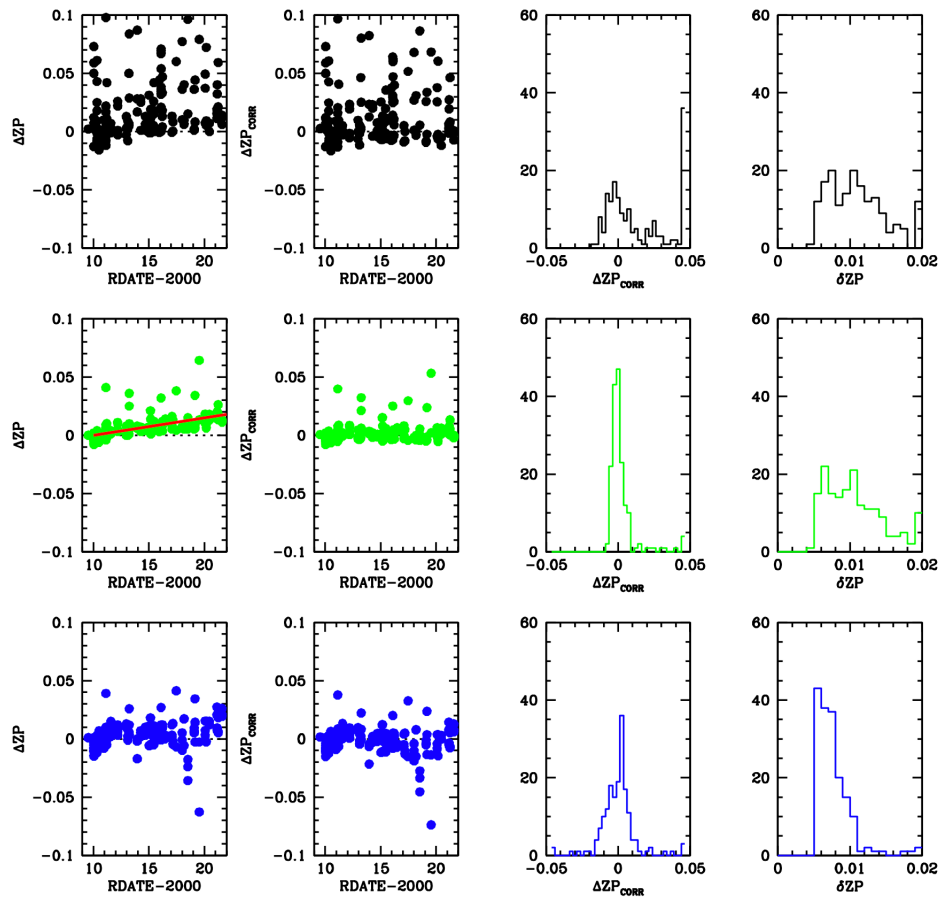
Figure P.1: Photometric residuals relative to the average for 170 F606W exposures of the center of Omega Cen. Blue: STDPSF, green: STDPSF+PERT1, blue: STDPBF(focus). Left column: raw zeropoint of each exposure. Second column: post sensitivity correction. Third column: distribution of zeropoints. Fourth column: spread about the zeropoint in each exposure.

The left panel shows the raw residuals with respect to time. There is a clear shift over time, which is related to the loss of sensitivity of WFC3 over time. We adopted a linear correction for this in the other three columns of panels. The histogram in the third column shows the distribution of zeropoints for each reduction. The histogram in the fourth column shows the consistency of photometry across the detector for each reduction.

It is clear that the plain STDPSF reduction can have large apparent zeropoint variations, typically +/- 0.02 magnitude, but it can be as large as 0.1 magnitude. The STDPBF has a tighter distribution of zeropoints, but it is not as tight as the STDPSF+PERT1. Both the PERT1 and STDPSF treatments improve the zeropoint variation considerably, but if the goal is to equalize the zeropoint then perturbation does the best job. Essentially, PERT corrects the distribution of light in the PSF to reflect the true-image PSF out to a radius of about 5 pixels. The focus-diverse PSF allows for this somewhat, but imperfectly.

The final panel shows the residuals about the zeropoint, in other words a measure of the consistency of the zeropoint across the detector. Note that the STDPSF and STDPSF+PERT1 have similar residuals across the detector, i.e., a one-size-fits-all perturbation does not change residuals across the detector. On the other hand, the STDPBF treatment *does* improve the zeropoint variation across the detector, providing more accurate relative measurements of stars in different places on the detector.

These results might encourage users to use the focus-diverse PSFs, but note that the improvement is actually quite modest. Most analyses can be done using the simple STDPSF product.

A final aspect of PSF fitting that can improve with improved PSFs is the quality-of-fit. **Figure P.2** shows the distribution of the average quality of fit for each exposure for the three different PSF reductions. The unadjusted STDPSF has a typical quality of fit of about 0.07, but if either a perturbation or a STDPBF is used, the typical quality is about 0.03, providing a better discriminant between stars and galaxies (or single stars and binaries, etc). **Figure 1** shows that impressive star-galaxy separation can be achieved even with the unmodified STDPSFs.
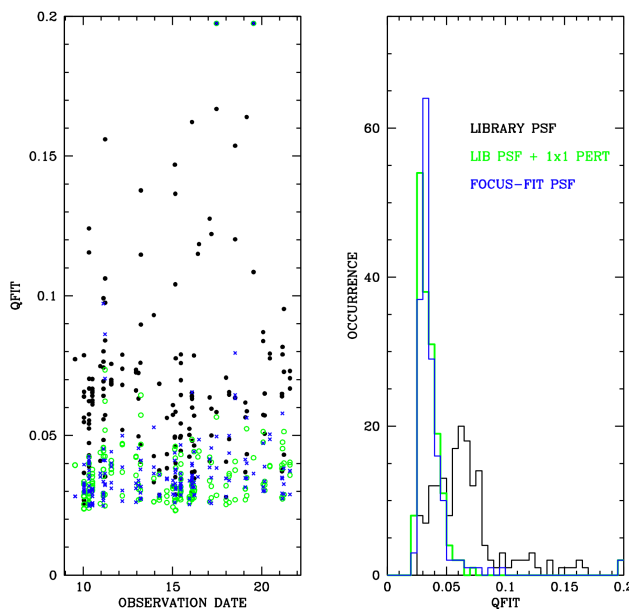
Figure P.2: The quality of fit metric for the three different PSF reductions.

Appendix S: Saturated stars

In CCDs that have a gain setting that allows sampling of the full well[22], the electrons from saturated stars are not lost — they are preserved on the detector as the charge bleeds from the central saturated pixels up and down along the columns[23]. Gilliland (2004, 2010) first showed that it is possible to obtain high-precision photometry for these stars by simply adding up the flux in the contiguous saturated distribution (along with a 1-pixel penumbra around it that may contain bled charge, but are not themselves above the saturated threshold). In this way, fluxes for saturated stars in ACS/WFC and WFC3/UVIS `flt`/`flc` images can be recovered. In general, we find that saturated-star fluxes are repeatable to about 0.02 magnitude.

We can also take advantage of the bleed-along-y property to measure accurate positions for saturated stars as well. If we collapse the star's profile along y and collapse the PSF model along y, we can cross-fit the two and determine an accurate y position for the saturated star. We find that such y positions are good to about 0.03 pixel, which is about three times the error present in a bright unsaturated star. Once we have an accurate position along the y axis, it is easy to get a position along x by fitting the unsaturated pixels to the PSF constrained at the measured y. The position along x is generally good to better than 0.05 pixel. We find that positions measured this way are typically about 4× better than positions that are fit to the partial annulus of unsaturated pixels around the saturated cores of stars. Having independent constraints on x and y clearly provides better results than solving for both simultaneously.

The WFC3/IR channel has a CMOS detector, not a CCD. Photometry and astrometry using CMOS detectors are less impacted by saturation than CCDs for two reasons. First, most pixels that saturate during the exposure will have some reads where they are not saturated, and a slope value can still be constructed for them. In addition, pixels that saturate do not bleed into other pixels, so saturation in one pixel does not spoil the adjacent pixels. Even so, some stars in WFC3/IR images do have unusable pixels in their cores. We can fit the good pixels in an annulus to get a flux and position, generally good to 0.1 magnitude and 0.1 pixel, respectively.

The WFC/HRC and WFPC2 at most gain settings saturate the A to D converter before they saturate the detector full well. As a consequence, the number of electrons is not preserved in the image, and we have to use the unsaturated pixels to obtain flux and position estimates. In general, these fluxes are good to about 0.1 magnitude and the positions to about 0.5 pixel.

As discussed above, saturated stars can generally not be measured as well as unsaturated stars. Sometimes the measurements are comparable, such as when electrons are preserved during bleeding, but even when saturated stars can be measured only crudely, it can still be useful to include them in our star lists: being able to compare the brightest stars in two lists can help considerably with field identification. Below, we will present some test cases that we ran that show how well saturated stars can be measured in the three main detectors.

---

[22] This includes essentially all observations made with ACS/WFC (gain 2) and WFC3/UVIS (gain 1.5), and WFPC2 (with gain 15) and for WFC/HRC (with gain 4).
[23] The potential barriers are lower along the columns than across the columns, so charge bleeds up and down and not right and left.

S.1 Saturated-Star Recovery in WFC3/UVIS

Examining the WFC3/UVIS archive of F606W exposures (which contains more than 10,000 individual exposures), we identified 154 short-long pairs of exposures taken at the same dither pointing. We measured the same stars in the short and long exposures and compared the measurements. The figure below provides the photometric residuals.
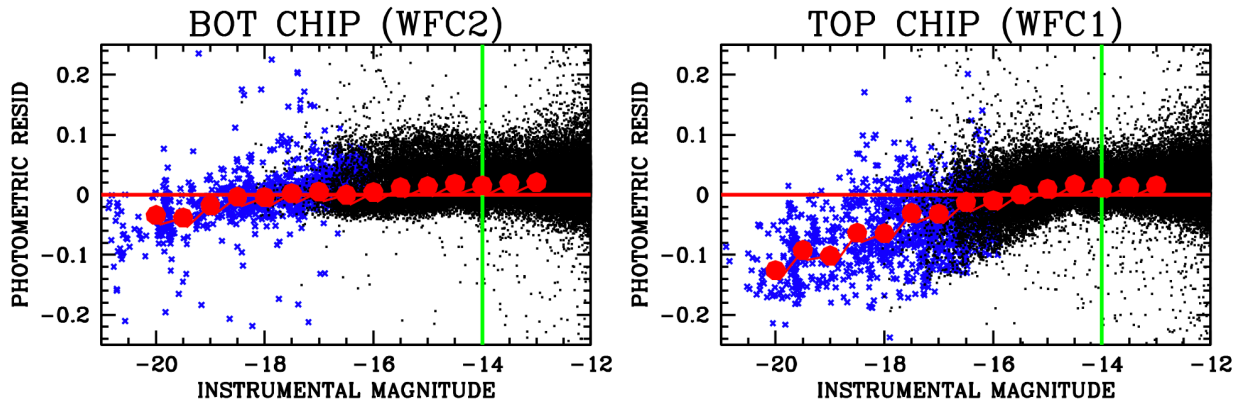

The stars are plotted against the deep-exposure instrumental magnitude, where saturation occurs at about −14 (marked by the green line). The stars shown in black are unsaturated in the short exposures, while the stars shown in blue are saturated in even the short exposures. It is clear that the photometry in the first 3 magnitudes of saturation is consistent to better than 0.02 magnitude, which provides evidence for the usability for the more saturated measurments (blue points).

The two WFC3 chips are known ot have different properties when it comes to electron-preservation in the presence of saturation (Gilliland et al. 2010). We developed a simple correction to account for these trends and implement it in the `hst1pass` routine. The following figures illustrate the correction's effectiveness: we are able to measure six magnitudes of saturated stars with systematically accurate positions and fluxes good to better than 0.05 magnitude.
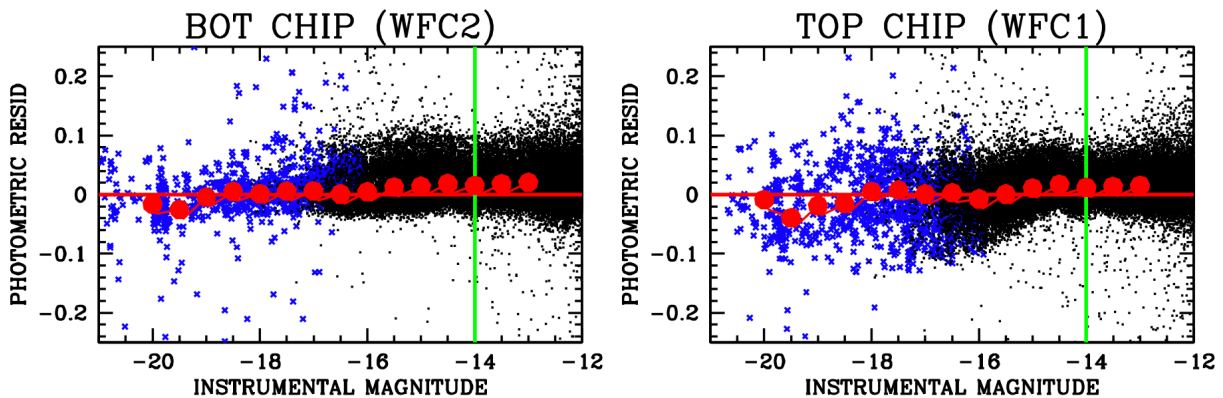
The following figure presents the astrometric offsets between the (saturated) stars in the deep exposures and the stars in the short exposures. In general, these positions are recovered with errors of about 0.05 pixel up to 4 magnitudes of saturation; beyond this level, the errors increase to tenths of a pixel.
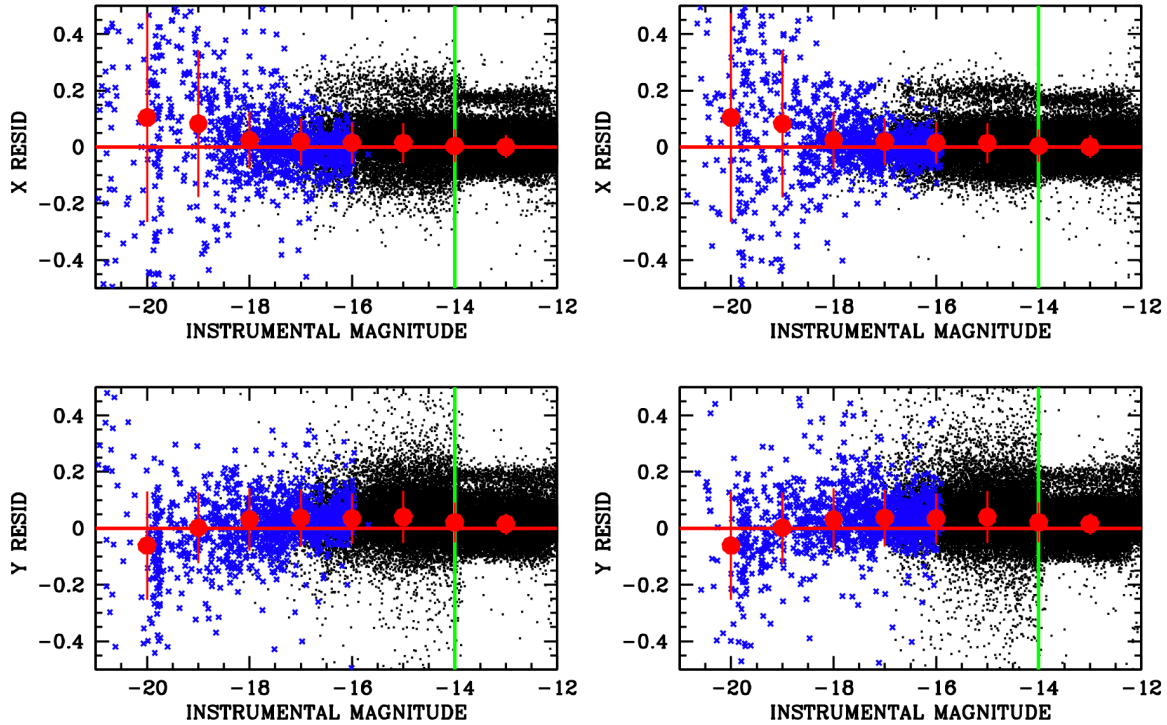
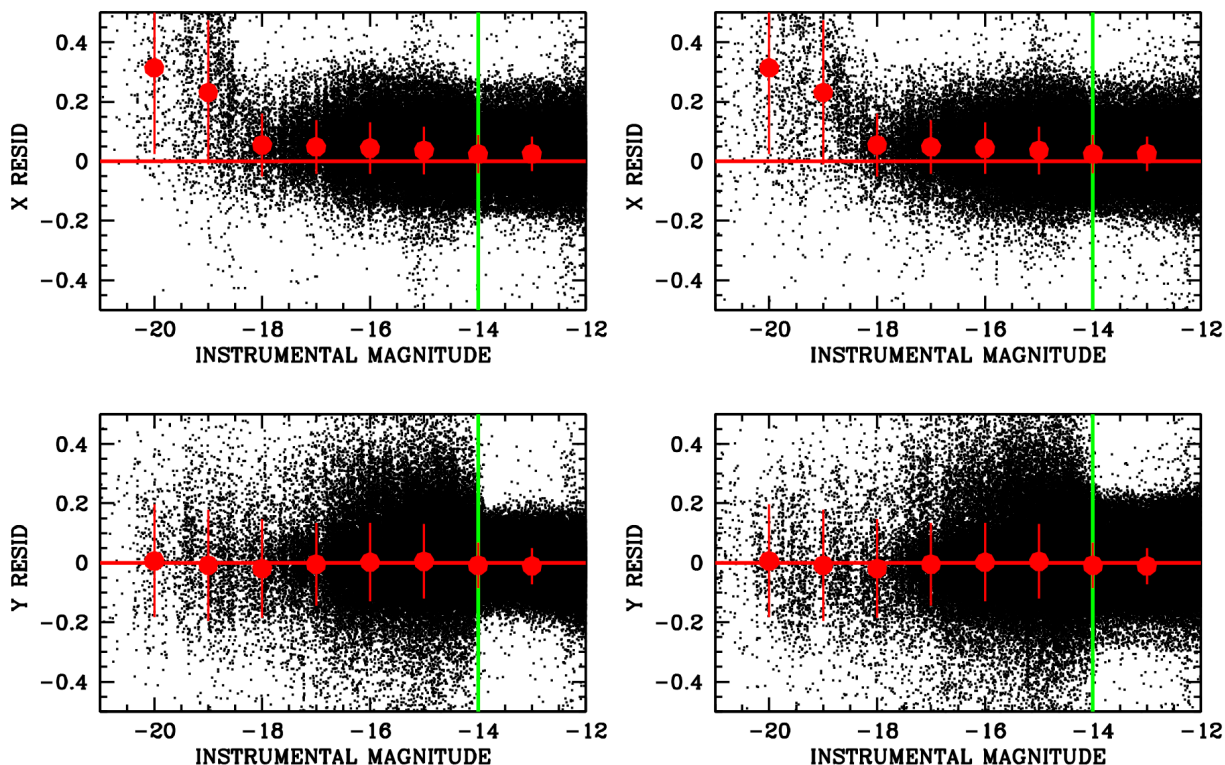## S.2 Saturated-Star Recovery in ACS/WFC

We performed a similar analysis for ACS/WFC to that done for WFC3 above, identifying 121 image pairs for comparison. As shown in the following plot, saturated stars are photometrically recovered with residuals better than 0.02 magnitude up to 6 magnitudes of saturation.
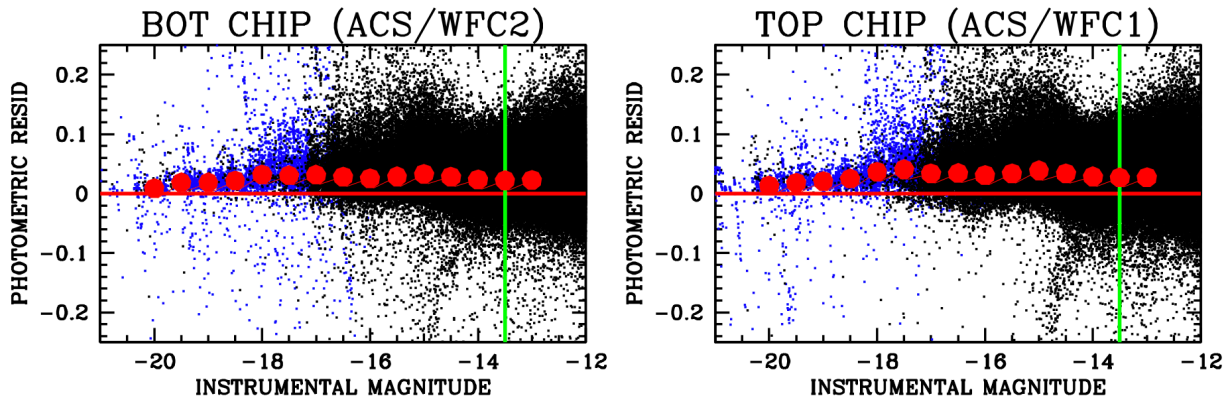
The following plot shows the astrometric residuals for measured saturated stars in ACS. In general, saturated star positions are measured to about 0.05 pixels up to four magnitudes of saturation. Stars beyond that level of saturation exhibit some systematic residuals up to half a pixel for stars that are saturated by more than 6 magnitudes (250 times saturation).

We developed a correction, implemented in `hst1pass`, so that saturated stars can even be measured reasonably well in the GAIN=1 ACS/WFC images, even though the full wells of electrons are not preserved.

The figure below shows astrometric recovery of saturated stars in ACS. There are significant trends in x for stars that are more than 4 magnitudes past saturation.

S.3 Saturated-Star Recovery in WFC3/IR

The WFC3/IR detector is not a CCD, but a CMOS detector. The downside is that charge is not preserved along bleed streaks up and down the columns. However, one benefit is that it does not have bleed streaks that can corrupt some of the closest pixels to the star's center along with pixels far away that could contain other stars. As a consequence, with WFC3/IR, we cannot use short-long pairs to examine saturation, since long and short exposures both have the same exposure time for the first read. Instead, we examined data taken through F160W of the core of Omega Cen. The images are all very similar, with exposure times between 228 s and 253 s. They were taken with a variety of pointings and orientations, so we can evaluate measurement quality by inter-comparing the observations and evaluating their consistency. We examined about 50 images taken between March 2010 and June 2011, so that the internal motions of the stars would not have a significant impact on our residuals.
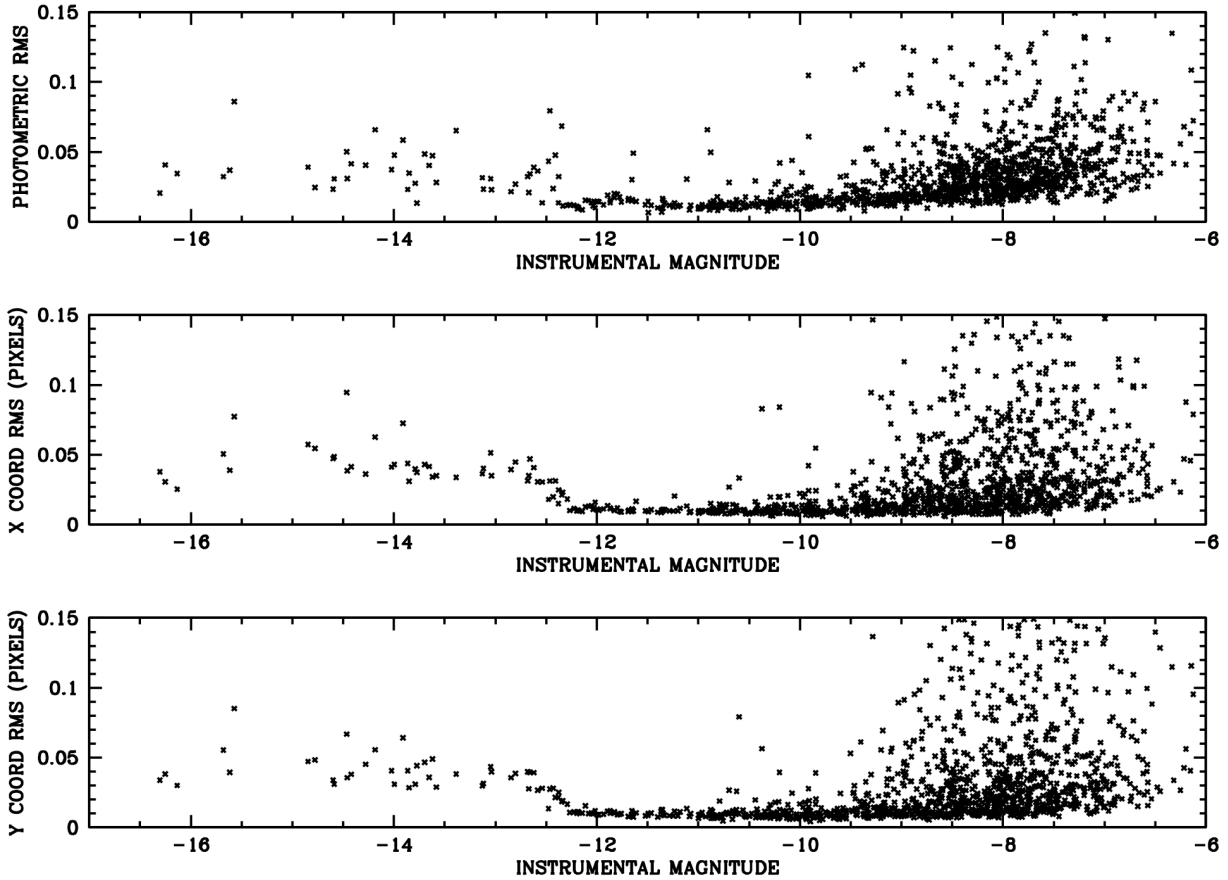
In the figure above, the resulting instrumental magnitudes are in terms of electrons per second, the units of the calibrated WFC3/IR flt images. Saturation in the first read occurs around −12.5, although there are some small features just below this where stars saturate before the last read. For the first 4 magnitudes of saturation, the photometry is generally good to about 0.05 magnitude and the astrometry is good to about 0.05 pixel.

Some observers have noted that comparing stars measured via full-ramp fits to stars measured using only the first few reads may not be completely equivalent because of jitter-related effects (M. Hosek, personal communication). In a CCD exposure, jitter effects are averaged into the PSF shape while in an IR ramp, the average jitter shift may be different in full and partial-ramp analyses. That is, the pointing that is representative of the first reset-read pair may be offset by a few hundredths of a pixel from the average based on the entire ramp. For this reason, users should take care when comparing the positions of bright stars and faint stars in WFC3/IR images. Given the up-the-ramp data, it is possible to correct for the effect but it has not been implemented in `hst1pass`.

Appendix T: Linear Transformations

Here we present the least-squares solutions for the transformation between two distortion-corrected frames, using the positions of stars found in both frames as the basis for the transformation. Specifically, for a given point (X_*,Y_*) in image 1, we seek the corresponding point (U_*,V_*) in image 2.

We begin with a star list which consists of positions for the same star in each of two images $\{X_i,Y_i\}$ and $\{U_i,V_i\}$. These star lists could have 10 members or 10,000. We assume that all of these stars are well-measured and have been verified as consistent with each other. This means that if one ``transforms'' the position of any of the stars in the list from one frame to the other by means of the prescription given below, the position will not be far off from the corresponding position in the list. We further assume that a distortion correction has been performed on both lists.

The task at hand is to take a point (X_*,Y_*) and desire to find the corresponding point (U_*,V_*) using the N stars for which we have positions in both frames. We will find the linear transformation that best relates the (X,Y) values of this list to the (U,V) values.

The problem to solve is of the form:

$$\begin{pmatrix} U - U_0 \\ V - V_0 \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} X - X_0 \\ Y - Y_0 \end{pmatrix}$$

There appear to be 8 free parameters in the fit, but in reality, there are only 6 parameters that must be solved for: A, B, C, D, U_0, and V_0, as the choice of X_0 and Y_0 is arbitrary.

The above represents the most general linear transformation possible. It allows for displacement, rotation, and an arbitrary stretch in either of the two coordinates (essentially in either system). The reason for the generalization is two-fold. First, we want to be able to treat any linear distortion that may remain after the image has been distortion corrected. Second, we want to be able to determine transformations freely from coordinate systems that may have different scales and orientations. Thus, we leave the transformation as general as linearity allows. Once we have solved for the 6 parameters, the point we are interested in will simply be:

$$U_* = u_0 + A(X_* - X_0) + B(Y_* - Y_0)$$
$$V_* = v_0 + C(X_* - X_0) + D(Y_* - Y_0)$$

Since the choice of X_0 and Y_0 are arbitrary we could choose to set them equal to (X_*,Y_*), in which case U_0 and V_0 would be equivalent to (U_*,V_*). However, it is much easier to evaluate the sums if we set them to the cetroid of the stars sin the list: $X_0 = (\Sigma X_i)/N$ and $Y_0 = (\Sigma Y_i)/N$, so

that $U_0 = (\Sigma\ U_i)/N$ and $V_0 = (\Sigma\ V_i)/N$. Let us also make a change of variables to: $x = X - X_0$, $y = Y - Y_0$, $u = U - u_0$, $v = V - V_0$, which reduces the equation to:

$$\begin{pmatrix} u - \delta u \\ v - \delta u \end{pmatrix} = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Note the solutions for U and V are completely separate and analogous. The least-squares sum we seek to minimize for U is:

$$S_U^2 = \sum \left(u_i - \delta u - A x_i - B y_i \right)^2$$

This yields three simultaneous equations for the three unknown parameters δu, A, and B.

$$\begin{pmatrix} N & \sum x_i & \sum y_i \\ \sum x_i & \sum x_i^2 & \sum x_i y_i \\ \sum y_i & \sum x_i y_i & \sum y_i^2 \end{pmatrix} \begin{pmatrix} \delta u \\ A \\ B \end{pmatrix} = \begin{pmatrix} \sum u_i \\ \sum u_i x_i \\ \sum u_i y_i \end{pmatrix}$$

recognize that $\Sigma x_i = \Sigma u_i = 0$, since they are defined to be the residuals from the mean. So that:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & \langle x^2 \rangle & \langle xy \rangle \\ 0 & \langle xy \rangle & \langle y^2 \rangle \end{pmatrix} \begin{pmatrix} \delta u \\ A \\ B \end{pmatrix} = \begin{pmatrix} 0 \\ \langle ux \rangle \\ \langle uy \rangle \end{pmatrix}$$

Then, the solution for this system is:

$$\delta u = 0.$$

$$A = \frac{\langle ux \rangle \langle y^2 \rangle - \langle uy \rangle \langle xy \rangle}{\langle x^2 \rangle \langle y^2 \rangle - \langle xy \rangle^2}$$

$$B = \frac{\langle uy \rangle \langle x^2 \rangle - \langle ux \rangle \langle xy \rangle}{\langle x^2 \rangle \langle y^2 \rangle - \langle xy \rangle^2}$$

The fact that $\delta u = 0$ simply states that if we adopt as (X_0, Y_0) the centroid of the list stars in frame 1, then the least-squares solution for (U_0, V_0) is simply the centroid of the list stars in frame 2.

Finally, the equation for U_* is:

$$U_* = \bar{U} + A(X - \bar{X}) + B(Y - \bar{Y})$$

The solution for V_* is analogous to that for U_*.

The same can be expanded to evaluate to evaluate the errors and include weighting in the calculation. Currently, all stars were treated to have equal weights. One could also examine residuals of the transformation to determine whether any particular stars might either have bad measurements in one frame or the other[24], or whether one star or another might have some other, possibly scientifically interesting, reason to be discordant with the others.

The inverse linear terms are easy to calculate:

$$A^{-1} = D / (AD - BC)$$
$$B^{-1} = -B / (AD - BC)$$
$$C^{-1} = -C / (AD - BC)$$
$$D^{-1} = A / (AD - BC)$$

With the inverse terms, we can then compute (X_*, Y_*):

$$X_* = X_0 + A^{-1}(U - U_0) + B^{-1}(V - V_0)$$
$$Y_* = Y_0 + C^{-1}(U - U_0) + D^{-1}(V - V_0)$$

There are also equations that estimate the errors in the linear transformation based on the residuals, but that is beyond the scope of the effort here.

---

[24] It's hard to tell which frame!

Appendix Z: Planned/Possible Future Improvements

While `hst1pass` was originally designed only for relatively isolated medium-to-bright point sources, it is still possible to use it for other applications given some enhancements to the code. Some of these improvements are easier than others, and initial progress has already been made for some of them. Users that would find particular improvements useful to their science goals are encouraged to send their comments, and any specific use cases, to **help@stsci.edu**. When new features are implemented, we will provide update the `NEWS.txt` file on the `hst1pass` website and the users-manual documentation will be updated as well (on the same site given in **Section 2**). Note that this ISR will not be updated going forward.

Z.1 Forced / Driven Photometry

The `hst1pass` routine is primarily designed as a find-then-phot application for moderate to high S/N isolated stars. As such, it typically searches the image for stars using specified star-finding criteria and employs a PSF to solve for the position and flux for each source.

It should be possible to run `hst1pass` in "forced photometry" mode where the user would supply a list of positions, and the routine would make its measurements at the specific input locations. This mode would be invoked with **FORCED=forcelist.xy** or **FORCED=forcelist.uv**. As with the artificial stars, `xy` positions would be assumed to be in the raw detector frame and the `uv` positions would be in the reference frame (accessed via the `_mat` file derived from `hst2collate`). The photometry mode is specified on the command line in the usual way, with PSF-fitting or aperture photometry — but the measurement would be done without solving for the position. Alternatively, a driven mode could use the list as a starting point and measure both flux and position for each star in the list.

When forced photometry is used, a flux is reported, irrespective of whether it is positive or negative. Thus, if a FORCED mode is implemented, then we would also enable the output of z, the measured flux. Currently that is not critical, since all found sources are centered on positive peaks and hence should be have a positive flux and a defined instrumental magnitude m.

A forced-mode photometry option would allow `hst1pass` to go faint and measure 1- or 2-sigma stars that cannot be found reliably in individual images. A list of such stars could be generated from a drizzled stack, even if they might be better measured by forced photometry on individual exposures. A forced photometry reduction would be able to include CTE corrections specific to each observation of each source.

Z.2 Two-Star Fitting

As the name suggests, `hst1pass` does one pass of photometry and astrometry, distinguishing it from routines that perform multiple passes through an image, first identifying the bright stars, removing them, then finding fainter stars in the subtracted residuals. Executing multiple waves of finding is well beyond the scope of this `hst1pass`. Users seeking that capability could run something akin to DAOPHOT on drizzled image stacks, perhaps using `hst1pass` to introduce artificial stars.

However, even without running multiple passes, one could consider the 2-star case separately from the crowded star-field case. The 2-star case arises quite commonly in HST images, as most stars in the field are members of binaries but not multiple systems. Therefore, one potential option would be to update `hst1pass` to search for star pairs and, when found, to analyze them with the PSF. Such an PSF-based analysis is often required to measure star separations or flux ratios for stars that are close enough to have their PSFs overlap.

Z.3 Limited Multi-Pass?

One could branch out beyond the 2-star fits to a limited multi-star fitting procedure. It would be difficult to identify multiple stars that are too close to each other, as that would require multiple dithered exposures to improve the resolution present in a single exposure. But it should not be impossible to propose an iterative scheme that finds and measures stars separated by more than 2 pixels, and includes the neighbor stars in the fits.

Z.4 Gaussian Fitting

In addition to providing a mechanism for measuring binaries, `hst1pass` could also provide a limited capability for measuring resolved objects. Such a mechanism could be useful to have within the code because it is hard to measure such things without proper treatment of the PSF. Since `hst1pass` knows how to perform accurate fits to pixels with the PSF, it could make sense to include a simplistic approach to profile fitting, perhaps even to measure a Gaussian with a point source at the center.

Z.5 Errors in Quantities

The `hst1pass` routine includes a simple scaled error estimate output quantity "e" that represents in a roughly what the fractional error in photometry or pixel location should be. More detailed errors could be calculated from the residuals to fits.

Z.6 Options Beyond PSF-Fitting 5×5 Pixels

The `hst1pass` routine by default uses a 5x5-pixel aperture when it fits PSFs to stars. There are some science cases where it may be better to use a smaller aperture, such as a 4x4-, a 3x3-, or even a 2x2-pixel aperture (the smallest aperture one can use and still measure a position).

The routine currently has the capability of measuring fluxes in apertures of different sizes, although the sky is currently measured well beyond the smallest apertures. Some applications might benefit from a tighter sky and a position measured from fewer pixels (or more pixels) in the aperture. For example, WFC3/IR has a very tight PSF and it is imaginable that a 3x3-pixel aperture might provide less neighbor-contaminated results.

Z.7 Bundles

Anderson (2014) devised a way to package multiple dithered observations of the same object into a single package (a "bundle") in order to facilitate simultaneous analysis of pixels in multiple exposures. It could be possible to have `hst1pass` create such a product, although other tools would be required to do the analysis.

Z.8 X-CTE

It should be possible to correct x-CTE in a similar manner to the y-CTE corrections that have been implemented in table form. These corrections would be much smaller and less dependent on flux, which is one reason they have not yet been constructed. But, once constructed, they should be easy to incorporate.

Z.5 jwst1pass (or st1pass?)

As of the initial writing of this ISR, it remains to be seen how well JWST data can be analyzed by the HST model. Every time a new HST camera has become available, the `img2xym` / `hst1pass` treatment has proven highly adaptable because the HST PSF and its distortion solution have been stable. In addition, since the data have been undersampled, the `hst1pass` treatment has brought significant improvement over aperture photometry or software constructed for ground-based telescopes. In principle, JWST should be amenable to the same treatment.

Z6. Possibility of running in the cloud

There an eventual possibility of `hst1pass` running in the cloud, so that users would not have to compile the FORTRAN code themselves. The idea is that users could upload images (or use images directly from the cloud archive) and specify finding/running parameters, and receive as outputs the catalogs and images generated by `hst1pass`. This functionality is not yet available, but it is being explored. We will advertise its availability on the `NEWS.text` page if it becomes available.