NATIONAL OPTICAL ASTRONOMY OBSERVATORY

Preprint Series

NOAO Preprint No. 881

WFPC2 Stellar Photometry with HSTphot

Andrew E. Dolphin

(National Optical Astronomy Observatory)

To appear in: P.A.S.P. (October 2000)

June 2000

WFPC2 Stellar Photometry with HSTphot

Andrew E. Dolphin
National Optical Astronomy Observatories, P.O. Box 26372, Tucson, AZ 85726
Electronic mail: dolphin@noao.edu

ABSTRACT

HSTphot, a photometry package designed to handle the undersampled PSFs found in WFPC2 images, is introduced and described, as well as some of the considerations that have to be made in order to obtain accurate PSF-fitting stellar photometry with WFPC2 data. Tests of HSTphot's internal reliability are made using multiple observations of the same field, and tests of external reliability are made by comparing with DoPHOT reductions of the same data.

Subject headings: techniques: photometric

1. Introduction

With the installation of WFPC2 in December 1993, Hubble Space Telescope (HST) gained an imager capable of high-resolution stellar photometry. This advance provided a number of opportunities, most notably the ability to obtain deep photometry in crowded fields such as nearby galaxies and globular clusters. However, the severely undersampled WFPC2 point spread function (the FWHM is comparable to a PC pixel and about half a WFC pixel) produces a challenge when attempting to obtain accurate stellar photometry. As a result, issues of the number of PSF points calculated per pixel by the photometry software, and of approximations of quantum efficiency variations and charge diffusion, which cause insignificant errors of well under a percent in well-sampled data, contribute significant errors of order a percent in PC data and greater in WFC data.

Because of the vast amount of WFPC2 data available to the astronomical community, HSTphot was developed specifically for its reduction, allowing for the creation of a highly specialized (and efficient) photometry program. The package (*hstphot* and accompanying utilities) runs from the Unix command line, and has been successfully compiled and run on machines running Solaris and Linux. As a manual is available with the package, this paper is intended to describe and test HSTphot rather than to give a detailed explanation of installation and use of HSTphot. Information on obtaining HSTphot can be obtained from

sections. For example, "HSTphot" refers to the entire photometry package, while "*hstphot*" refers to the specific program that runs the photometry solution.

2.1. Image Preparation

A collection of image preparation utilities is provided with the HSTphot package, which run the necessary processing steps such as masking of bad columns, cosmic ray cleaning, hot pixel masking, etc.

The first preparation procedure is the masking of bad columns and pixels, which is done with the *mask* routine. This simple routine will read the data image (c0f) and the data quality image (c1f) provided by STScI, and proceeds to mask out all pixels that are deemed to be bad - types 1 (Reed-Solomon decoding error), 2 (calibration file defect), 4 (permanent camera defect), 16 (missing data), 32 (other bad pixel), 256 (questionable pixel), and 512 (unrepaired warm pixel). This masking will also eliminate the vignetted region in recent images (very early data quality images do not flag this region). Row 800 and column 800 are also masked out entirely. Because the saturation flag (type 8) in the data quality image is unreliable, all pixels with 3500 or more counts are set as saturated (4095 DN) to avoid ambiguity later in the reductions. All masked pixels are set to the bad data value, -100 DN, and are ignored for the remainder of the photometry.

2.2. Image Cleaning and Combination

The masked image is then ready for cosmic ray cleaning and combination, a process for which the utility *crclean* is designed. This utility uses a routine based on the IRAF task CRREJ, itself a more sophisticated version of the elementary "maximum value reject" method of combining images. *Crclean* is provided a set of images to be combined, which must be taken at the same pointing and with the same filter, and compares the images at each pixel position. All unmasked and unsaturated pixels at that position are scaled for their respective exposure times and compared. The simple procedure would be to use the either median or minimum value of the pixels at a given position as a comparison value (*crclean* provides both options for the user), and reject all values that fall more than

$$\text{max deviation} = \frac{\sigma_{threshold}\sqrt{\text{Read Noise}^2 + \text{counts/Gain}}}{\text{Exposure Time}} \qquad (1)$$

away from the comparison value. In this equation, Read Noise and counts are both expressed in DN for simplicity, and Gain is expressed in the usual e^- per DN. The value

sky-subtracted pixel values. Second, the sky-subtracted pixel value is more than seven standard deviations above the average value of adjacent pixels. The intent of this utility is not to locate and remove every hot pixel, rather only those that are sitting on blank sky and thus very easy to detect.

As likely clear by the descriptions, both cleaning stages are intentionally cautious in the pixels that are thrown away. Given the very sharp PSFs in the WFC images, this approach seems wise, as a star damaged in the cleaning process is extremely difficult to fix, while most false detections that escape the cleaning process will be identifiable in the photometry output due to unusual χ or sharpness values.

2.3. Background Determination

The sky or background determination is the final mandatory pre-photometry step of HSTphot, and is done with the *getsky* utility. Obviously this is a task that could also be accomplished within *hstphot*, but given the occasional need to re-run *hstphot* with different detection parameters it is preferable to have the sky determined only once. The sky value is calculated at each pixel, using the robust mean of pixel values inside a square "annulus" centered on that pixel. The adoption of the square annulus, rather than the more typical round shape, was made for computational ease (the less multiplication, the faster the program runs). Given that the inner "radius" is sufficiently far from the pixel in question ($\sim$8 FWHM) that the shape is of little consequence. For the PC, the inner square is 33 pixels on a side and the outer square 45 pixels, thus giving a maximum of 1064 pixels used for the sky determination. The WFC sky calculations use squares of half this size (a 23 pixel outer square and a 19 pixel inner square), with a maximum of 240 pixels used. The robust mean of all unmasked and unsaturated pixels within this area, using a recursive rejection of pixels more than 2.5σ below and 1.75σ above the mean value, is computed. Convergence is determined when a pass rejects no pixels, and the mean sky value from that final iteration is set as the sky value for the pixel. In order to ensure a smoothly-varying background, the sky image is boxcar smoothed to determine the sky values that will be used by *hstphot*.

A few comments regarding the sky calculation process are in order. The sky value can be calculated in one of three ways: a single calculation before the photometry process, a calculation immediately preceding the photometric measurement of a star, and a calculation simultaneous with the photometric measurement. The final choice is the most appealing, as the χ^2 fitting procedure should have little trouble in distinguishing the flat sky from the variable stellar PSF and therefore one can determine the "true" sky value underneath each star with ease. However, as demonstrated by Stetson (1987) and confirmed in my

when beginning photometry on an image.

Because of the undersampled PSFs, it was decided to calculate the synthetic PSFs for a variety of subpixel centerings, with a spacing of every 0.2 PC pixels (25 total centerings) and 0.1 WFC pixels (100 total centerings). These PSFs were calculated using Tiny Tim PSFs, which were generated with subsampling settings of 0.1 PC pixels and 0.05 WFC pixels for additional resolution. Next, a charge diffusion correction was applied to the Tiny Tim PSF, which was equivalent to smoothing the subsampled PSF with a Gaussian kernel of $\sigma = 0.32$ pixels. (The value of 0.32 pixels was determined by trial-and-error, with this value producing the lowest *hstphot* median χ value for high signal-to-noise data. The value of 0.32 is probably accurate to within 0.05 pixels.) Finally, a subpixel quantum efficiency variation of roughly a 10% efficiency decrease from center to corner was applied, with this value determined in the same way. The choice of the QE fluctuation amount turned out to have very little impact on the quality of the fits, and was thus difficult to determine accurately, with any value between 5% and 15% returning indistinguishable median χ values.

This process was repeated at 64 positions per chip (every 100×100 pixels), thus generating a total of 1600 PSFs on the PC and 6400 PSFs per WFC. Again, it should be noted that the use of Tiny Tim PSFs is not new to HSTphot; CCDCAP (Mighell & Rich 1995), for example, uses Tiny Tim PSFs to determine aperture corrections for small-aperture photometry. However, HSTphot provides what is, to my knowledge, the most elaborate application.

With the grid of PSFs by chip position and subpixel centering calculated, it is worth examining the effect of using these quantized grids rather than an analytic function (such as what is used by DAOPHOT and DoPHOT). It should first be noted that the errors discussed here are of PSF shape rather than the total PSF size. Thus, while a PSF whose total number of counts was in error by 1% would create photometry with an error of 0.01 magnitudes, a PSF whose central pixel was in error by 1% but whose total size was correct would create a much smaller photometric error. The number of counts in a simple χ^2 PSF fit minimization is

$$\text{counts} = \frac{\sum Residual \times PSF/\sigma^2}{\sum PSF^2/\sigma^2}, \tag{3}$$

which can be simplified to

$$\text{counts} = \frac{\sum Residual}{\sum PSF} \tag{4}$$

and

$$\text{counts} = \frac{\sum Residual \times PSF}{\sum PSF^2} \tag{5}$$

exist, as the limit of constant noise is only reached if the star has zero counts. Any actual detectable star will contribute some amount to the noise, and thus have a smaller random scatter than the 0.004 magnitudes calculated here. Finally, it is worth reiterating that the brightest stars are unaffected by these considerations.

Within *hstphot*, the PSFs are modified further to compensate for the geometric errors of geometric distortion and the 34th row error. Both of these factors decrease the effective pixel sizes, and thus it is necessary to magnify the PSFs accordingly. The geometric distortion pixel sizes are calculated via the Holtzman et al. (1995a) distortion correction equations; the 34th row error (noted by Shaklan, Sharman, & Pravdo 1995) is characterized by row heights calculated from data supplied by Ron Gilliland. The PSF magnification process is quite simple, with the effective pixel width or height used to determine the fraction of the light from a given row or column that should be moved into the adjacent row or column. To compensate for the fact that the actual PSF value on the outer edge of a pixel is much less than the average PSF value within a pixel, and thus the amount of light transferred should be less, the PC and WFC transfer amounts are multiplied by 0.70 and 0.65, respectively. This value is determined for the typical central pixel, with a worst-case error of a 0.3% in the corrected value of the central pixel, or a maximum magnitude error of 0.002 magnitudes in our hypothetical zero count star. Again, the detailed correction of pixels other than the central pixel are of less interest, given that the PSF^2 term in the photometric solution will make such errors negligible.

It should be noted that the expansion of PSFs for these geometric errors is a necessity for obtaining accurate PSF-fitting photometry, in addition the the normally-recommended step of multiplying the image by the effective pixel area map (cf Holtzman et al. 1995a). While field-varying PSFs from DAOPHOT or DoPHOT should compensate for the smoothly-varying geometric distortion, the 34th row error needs to be compensated similarly, something which, to my knowledge, is not done in any other photometry package. The use of an uncorrected PSF on an uncorrected image will produce photometry for a star on an affected row in which the shape matches well (thus producing negligible error for faint stars) but the total number of counts detected is 2% too large for bright stars, giving a magnitude error of -0.02 magnitudes. If the data are multiplied by the row size image, the total number of counts are correct (giving correct photometry of bright stars) but the central pixel has an error of 3%, producing a magnitude error of -0.02 magnitudes for the limiting-case faint star. Thus, while the latter case is preferable since the larger random errors should minimize the effect of the systematic +0.02 magnitude shift, in either case one will see a relative error of a few percent between bright and faint stars. Naturally, such an effect only matters significantly for roughly 3% of the stars, but it is my intention that known (and correctable) systematic errors at more than the 1% level be eliminated.

As noted in section 2.3, *hstphot* allows the user to either accept the *getsky* sky image or determine a sky value adjustment before each photometry measurement. If the δsky option is used, the sky modification is calculated immediately beyond the photometry radius, at a distance of roughly 5.5 pixels from the star in PC images and 4 pixels in FC images. This value, again calculated using a robust median routine, is subtracted from the residuals during the photometry solution. Because a sky level determined so close to the star will invariably measure some of the starlight as well, the PSFs are also adjusted by subtracting the robust mean in the same region from all PSF values during the photometry solution.

The quality-of-fit parameter that is maximized in the search for the star's center is based on the detected signal, formal error, and χ determined at each point. These values are defined as follows, calculated over a circular aperture with an effective radius of 3 pixels in the PC and 2 pixels in the WFCs. (These radii contain ~80% of the total starlight, using the encircled energy measurements of Holtzman et al. 1995a.)

$$\text{signal} = (\sum_{x,y} R_{x,y} \times PSF_{x,y}/\sigma^2_{x,y} \times wt_{x,y})/(\sum_{x,y} PSF^2_{x,y}/\sigma^2_{x,y} \times wt_{x,y}), \tag{8}$$

$$\text{error} = (\sqrt{\sum_{x,y} PSF^2_{x,y}/\sigma^2_{x,y} \times wt^2_{x,y}})/(\sum_{x,y} PSF^2_{x,y}/\sigma^2_{x,y} \times wt_{x,y}), \text{ and} \tag{9}$$

$$\chi^2 = [\sum_{x,y}(R_{x,y} - \text{signal} \times PSF_{x,y})^2/\sigma^2 \times wt_{x,y}]/(\sum_{x,y} wt_{x,y}). \tag{10}$$

$R_{x,y}$ is the residual after subtraction of the sky, $PSF_{x,y}$ is the value of the PSF at the trial chip position and subpixel centering, and $\sigma_{x,y}$ is the expected uncertainty of the measurement at that pixel,

$$\sigma^2_{x,y} = R_{x,y} + sky_{x,y} + \text{Read Noise}^2. \tag{11}$$

In order to prevent χ^2 from becoming extremely large for bright stars because of PSF errors which cause $R_{x,y} - \text{signal} \times PSF_{x,y}$ to grow proportionally to the star's brightness, a factor of $c^2\text{signal}^2 \times PSF^2_{x,y}$ is added to $\sigma^2_{x,y}$ for the determination of χ, with c values of 0.19 in the PC and 0.25 in the WFC providing a median χ near one in the final photometry. This addition is similar to the change from Equation 1 to Equation 2 in the cosmic ray rejection algorithm. Finally, the $wt_{x,y}$ is a weighting factor equal to

$$wt_{x,y} = R_{eff} + 0.5 - \sqrt{(x - x_c)^2 + (y - y_c)^2}, \tag{12}$$

but not allowed to exceed one or drop below zero. R_{eff} is the effective radius (3 PC pixels or 2 WFC pixels), and x_c and y_c are the X and Y positions of the trial position for the star's center.

which this is the case, the PSF residual (described in the following section) is calculated. The solution is considered completely converged after a user-defined maximum number of iterations have run or after no stars are flagged following an iteration.

This solution technique, like the initial star detection process, turned out to be very similar to that used by DoPHOT. However, there are a few significant differences that should be noted. DoPHOT uses a nonlinear χ^2 minimization routine that determines the best position and brightness through a single minimization. HSTphot, on the other hand, maximizes its goodness-of-fit parameter by calculating that value directly at a range of positions. Given that the topology of the search space is well-behaved for a star, such a method is advantageous in that the determination of the goodness-of-fit at a single position is straightforward (using the equations in the previous section) and storing the previous trial fit values in memory will make it trivial to avoid duplicate measurements at the a position. Additionally, the fact that all data are similar allows HSTphot to use a search stepsize that is sufficiently fine to allow good fits while sufficiently large to permit a rapid convergence. The issue at stake is one of efficiency rather than accuracy, with HSTphot reducing data at the same or greater speed than DoPHOT.

An additional difference is seen when comparing HSTphot and DoPHOT with DAOPHOT. While the first two packages share a one-star-at-a-time solution technique, DAOPHOT will solve for all overlapping stars simultaneously. The speed penalty here is considerable, with an order of magnitude difference in running speed between DoPHOT and DAOPHOT reported by Schechter et al. (1993), thus raising the question of whether or not a simultaneous solution of neighboring stars is actually necessary. Given the facts that the cores of WFPC2 PSFs are extremely narrow and that an iterative solution can continue until the faint neighbor stars have converged properly, it would seem that an iterative one-star-at-a-time solution will produce equally good photometry of neighbor stars as will the more sophisticated DAOPHOT method. This assumption is verified through a comparison of HSTphot and DAOPHOT photometry presented by Dolphin (1999) for the WLM globular cluster, as well as a more recent comparison of photometry of the Cassiopeia dwarf spheroidal galaxy presented by Dolphin et al. (1999), with HSTphot producing sharper CMDs with more stars in both cases. Although it is not claimed that the one-star-at-a-time iterative method produces better photometry, these examples demonstrate that it will not significantly hurt the photometry, even in the case of the very crowded WLM globular cluster field.

After convergence is reached or the maximum number of iterations have run, a final photometry iteration is attempted to improve the accuracy. Note that in all previous photometry stages, star positions are only determined at the PSF library grid points (every

is necessary because of the implications of equations 4 and 5 - photometry of a bright star depends primarily on the total number of counts in the PSF (which should always be accurate) while that of a faint star depends on the shape of the PSF. In concept, this modification of the library PSFs is similar to the DAOPHOT calculation of a residual used to modify its analytic PSFs. Thus, any error in the PSF caused by the assumption that the library PSFs are correct in every image will have little impact on the bright stars, but the magnitudes of the faint stars will be systematically in error. Stetson (1992) reported systematic errors of up to 0.25 magnitudes over a span of ~6 magnitudes in WF/PC data; my own experiments comparing PSF-fitting photometry using unadjusted library PSFs with aperture photometry show deviations of up to 0.15 magnitudes over a similar magnitude range in WFPC2 images, with a typical image showing systematic devaitions of 0.05 magnitudes.

Thus it is necessary to calculate a PSF residual image for each frame. A set of stars meeting the following criteria is selected as the set of PSF stars.

- -0.5 $\leq$ sharp $\leq$ 0.5
- $\chi \leq 4$
- The star is more than the PSF radius (6 PC pixels, 4 WFC pixels) away from the edge of the usable chip area
- There are no saturated or masked pixels within the PSF radius of the star.
- No brighter star is within 9 PC pixels or 6 WFC pixels of the star.
- No fainter star with at least half the brightness is within the PSF radius of the star.

The PSF residual is then calculated through an iterative process. First, the average residual around the PSF stars is determined through a robust mean at each point, comparing the residuals around the individual stars (weighted by 1/counts, of course). Simply adding the residual to the current PSF, however, will not conserve the number of counts in the PSF, as the mean residual can (and usually will) have a nonzero sum. Since the true PSF should be proportional to the sum of the current PSF and the mean residual, the corrected PSF is set to be

$$new\ PSF_{x,y} = c(PSF_{x,y} + R_{x,y}), \tag{16}$$

where $R_{x,y}$ is the mean residual at the point and c is a constant chosen to conserve the PSF size,

$$1/c = 1 + (\sum x, yR_{x,y})/(\sum x, yPSF_{x,y}). \tag{17}$$

from 0.5 to 0.75 arcsec from the star. The aperture correction for the star is simply

$$\Delta mag = -2.5 \log(\frac{C_{aperture}}{C_{hstphot}}), \tag{18}$$

where $C_{aperture}$ and $C_{hstphot}$ are the counts within the aperture measured by *getapcor* and the counts measured by *hstphot*, respectively. Another robust mean routine is used to determine the overall aperture correction for each chip, based on the Δmag values for all of the aperture stars.

2.9. Multiphot

A recent addition to the HSTphot package is *multiphot*, a program designed to simultaneously solve multiple images in a single run. Again, this concept is far from new, and is to *hstphot* what ALLFRAME (Stetson 1994) is to DAOPHOT. The primary advantages in using *multiphot* rather than *hstphot* are that it is easier to compensate for bad pixels or cosmic rays that are only in one image and that a more accurate photometric solution is obtained by reducing the number of free parameters per star from $3 \times N_{images}$ (X, Y, and counts in each image) to $2 + N_{images}$ (global X and Y positions and counts determined in each image). However, there is somewhat of a tradeoff involved - in order to prevent *multiphot* from using too much memory (*hstphot* allocates about 20 Mb per image), some simplifications had to be made, along with the use of 32-bit floating point rather than 64-bit. The simplified photometry, which affects the expected noise at each pixel, increases uncertainties by roughly 0.005 magnitudes for bright stars, while the increased roundoff error contributes another 0.002 magnitudes of uncertainty. Given a limiting accuracy of about 0.011 magnitudes determined for *multiphot*, these uncertainties will contribute little to the overall error budget.

Overall, the *multiphot* algorithms are nearly identical to those in *hstphot*. The only significant exception is the necessary one: when determining the goodness-of-fit of a trial position, a combined signal, error, and χ are used instead of just the values from a single image. (All PSF information is separate for the exposures, of course, with a different residual image calculated for each image being reduced.) However, there are a few additional complexities that must be addressed by *multiphot*.

The most significant additional problem faced by *multiphot* is that the images need to be properly aligned. To ensure accurate PSF-fitting magnitudes, the accuracy needs to be accurate to at least 0.1 WFC pixels. Thus the alignment issue requires a correction for geometric distortion (which increases the offsets between two images by up to 2% at

3. Tests of HSTphot

An F555W WFC2 image of IC 1613 is displayed in Figure 1, with the detected stars and cosmic rays subtracted from half the image in order to provide in initial "sanity check" on HSTphot. (Note that the white lines and dots are the masked bad columns and hot pixels, not residuals.) In general, *hstphot* appears to be treating objects as it should. The galaxies are left alone, aside from what are either foreground stars or clusters or HII regions in the galaxies, which indicates that the object classification scheme in *hstphot* is working properly.

The stellar residuals themselves also provide a sanity check. The lack of similar features in the bright residuals is evidence that the average PSF is correct, and that the PSF residual image was correctly determined. The lack of monopole residuals in the stars likewise indicates proper PSF width and star brightness determinations. The lack of dipole residuals (highly positive on one side and highly negative on the other) is indicative of proper centering. Although this is certainly not a quantitative test, it gives evidence that, at the very least, HSTphot isn't completely off target in its methods. More detailed (and quantitative) comparisons follow in the next sections.

3.1. Photometric Reliability

In order to determine whether or not HSTphot photometry is reliable, a comparison was made between photometry of eight combined 2400s F555W images of a field in IC 1613. The data were taken with four dithering positions, essentially providing four independent sets of observations of the same field. (In fact, it should be noted that the dithering provides something of a worst-case comparison, as stars centered in one image are on the edge or corner in the others.) The data were processed in the standard way, with each image combined from two 1200s images for cosmic ray removal, and run through HSTphot independently. (Note that *Multiphot* could have been used to produce more accurate photometry, but the resulting photometry would not have been completely independent.)

The results of the comparisons between the fields are shown in Figure 2, with a 1σ line plotted through the data. *Hstphot* uncertainties for the points are shown for comparison in the bottom panel. The *hstphot* uncertainties match the measured 1σ uncertainties well, except at the bright end where there appears to be a minimum error of 0.027 magnitudes. Part of this error is due to the assumption that none of the stars in these data are variable, which is incorrect. After running a variable star analysis and retaining only the stars that are apparently non-variable, the minimum error in the HSTphot photometry drops to 0.02

5 shows all objects in one of the IC 1613 images, with cosmic rays and extended objects (as determined by *hstphot* via object type determination) plotted with circles and squares. The "stars" well above the principal trend (at sharpness $\geq$0.4) are multi-pixel cosmic rays, which were not identified by *hstphot* since its cosmic ray model is a single pixel. In general, requiring a sharpness between -0.3 and +0.3 should reject most cosmic rays and extended objects that were not classified by *hstphot*, while retaining most of the stars.

Finally is the χ column, which simply gives the quality of the fit. The bottom panel of Figure 5 shows all objects, with χ plotted against counts. The unidentified cosmic rays seen in the sharpness plot are also seen here, with χ values of greater than 3 or 3.5, and either limit thus useful for eliminating false detections. The use of other poorly-fit stars depends on the quality of the data and the application.

These plots permit the determination of selection criteria for *hstphot* output. If one needs a complete CMD, a χ threshold of 3 should keep most of the stars, but if a clean CMD is preferable a χ threshold of 1.5 will eliminate most detections. The χ limit should be relaxed in the case of very crowded images, such as what is seen in star clusters and in the denser regions of Local Group galaxies. A plot similar to that given in the bottom panel of Figure 5 will quickly determine the appropriate cutoffs for any given situation.

3.4. Comparison With DoPHOT

Comparison photometry on one the WFC2 chip of one of the IC1613 images was provided by Jennifer Christensen using DoPHOT, as described by Saha et al. (1996), with aperture corrections and calibration but the no CTE or zero point corrections applied. Likewise the *hstphot* magnitudes were given aperture corrections but no CTE or zero point corrections. Given that HSTphot is somewhat of a black box with very few adjustable parameters, my photometry using HSTphot is probably identical to what anyone else would obtain. The DoPHOT photometry package was run in the manner prescribed by Saha et al. (1996). As this method is optimized for Cepheid studies and thus uses a background region very close to the star, the δsky option was used in the HSTphot reduction as well to avoid an inherent advantage given to HSTphot by the use of a larger sky region.

The differences between the HSTphot and DoPHOT F555W and F814W magnitudes are shown in Figure 6, and show no significant differences. Overall, the median difference is +0.001 magnitudes, with a slight (0.01 magnitude) difference in the individual filters that is comparable to the uncertainty in the HSTphot aperture corrections and thus not significant. The respective CMDs are also shown in Figure 7, with the HSTphot CMD a

including sky calculation, aperture corrections, and alignment. This speed is comparable to the rate at which the STScI archive provided the data. HSTphot is also being used by the HST Snapshot Survey of Nearby Dwarf Galaxy Candidates (Seitzer et al. 1999) to provide uniform photometry of the sample of galaxies, as well as by other users.

I would like to thank Jennifer Christensen for providing the DoPHOT comparison photometry and Ron Gilliland for providing the data used for the 34th row correction. I would also like to thank Dan Zucker and Ted Wyder for help with debugging HSTphot and facilitating the port to linux. This work was supported by NASA through grants GO-02227.06-A and GO-07496 from Space Telescope Science Institute.

REFERENCES

Anderson, J., & King, I. R. 1999, PASP, 111, 1095

Dolphin, A. E. 1999, PhD thesis, University of Washington

Dolphin, A. E. et al. 1999, AAS meeting 195, 08.04

Holtzman, J. et al. 1995a, PASP, 107, 156

Holtzman, J. et al. 1995b, PASP, 107, 1065

Mighell, K. J., & Rich, R. M. 1995, AJ, 110, 1649

Saha, A., Sandage, A., Labhardt, L., Tammann, G. A., Macchetto, F. D., & Panagia, N. 1996, ApJ, 466, 55

Schechter, P. L., Mateo, M., & Saha, A. 1993, PASP, 105, 1342

Seitzer, P. et al. 1999, AAS meeting 195, 08.01

Shaklan, S., Sharman, M. C., & Pravdo, S. H. 1995, Appl. Opt., 34, 6672

Stetson, P. B. 1987, PASP, 99, 191

Stetson, P. B. 1992, in *Astronomical Data Analysis Software and Systems I*, eds. D. M. Worrall, C. Biemesderfer, & J. Barnes, ASP Conference Series, 25, 297

Stetson, P. B. 1994, PASP, 106, 250

Stetson, P. B. 1998, PASP, 110, 1448

Stetson, P. B., Davis, L. E., & Crabtree, D. R. 1990, in *CCDs in Astronomy*, San Francisco: ASP, p. 289

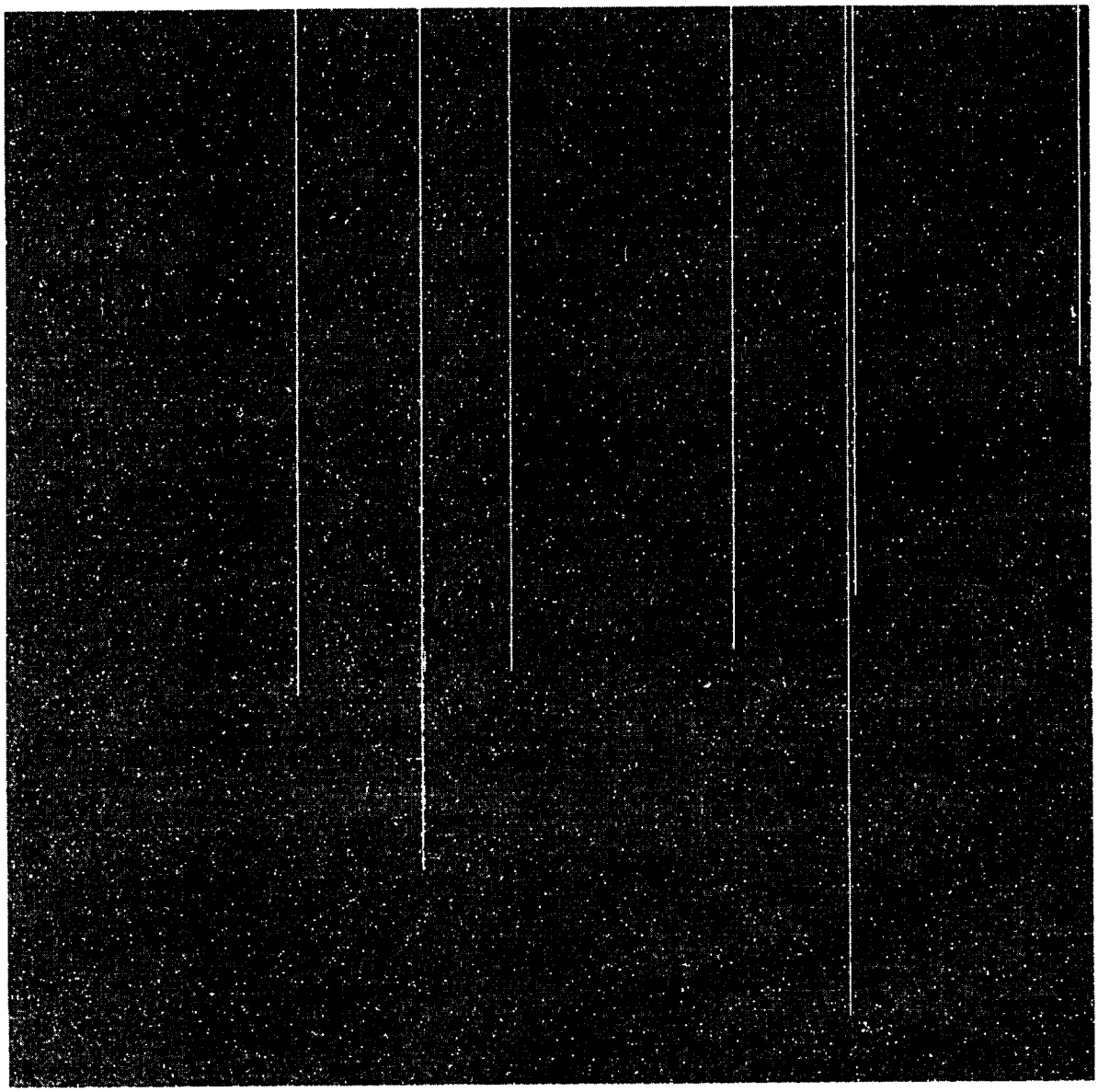

Fig. 1.— F555W WFC2 image of IC 1613, with stars subtracted from the left half

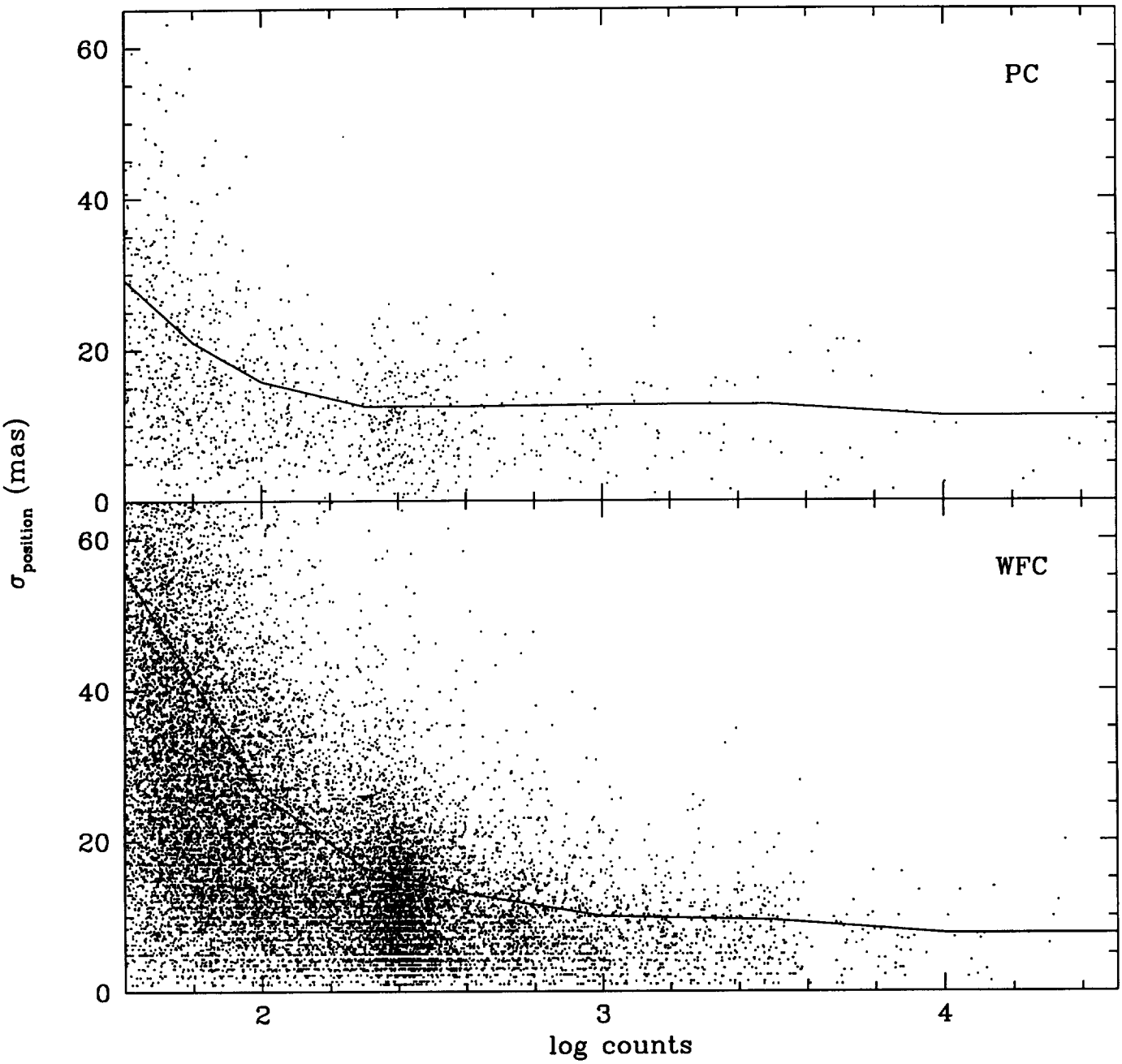

Fig. 3.— Astrometry error, based on a global astrometry solution

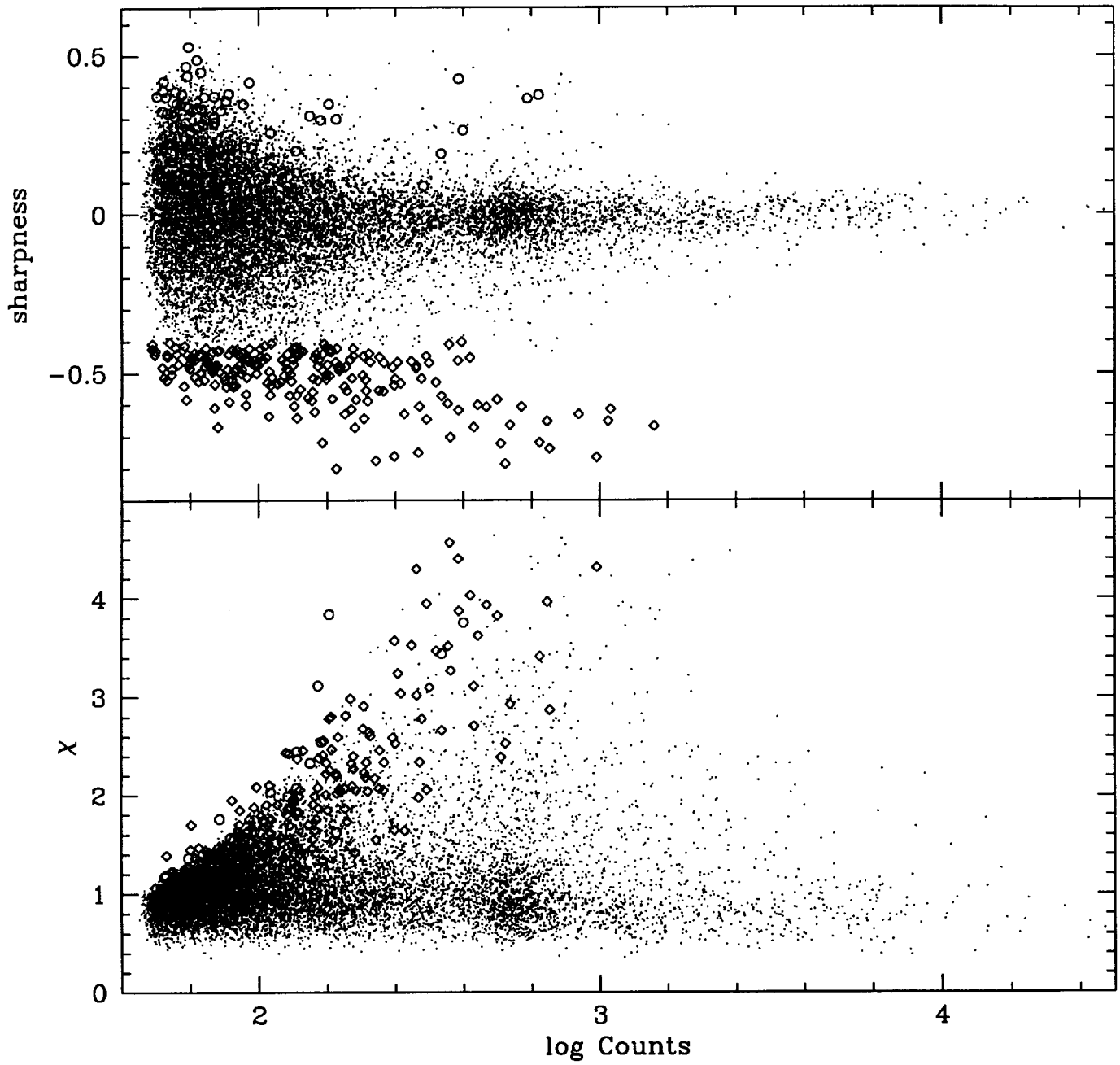

Fig. 5.— Sharpness and χ values for stars in an F555W image. HSTphot type 4 (cosmic rays) are plotted as circles; type 5 (extended objects) are plotted as diamonds.

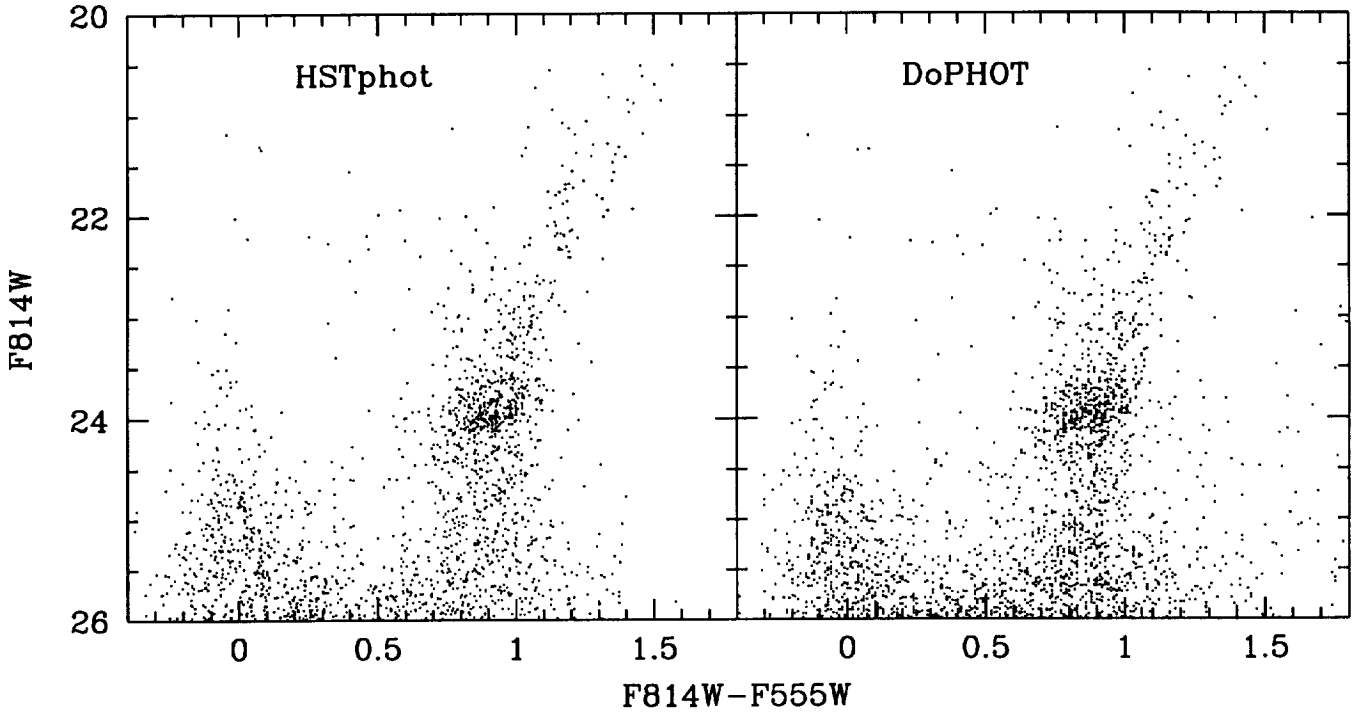

Fig. 7.— (F555W-F814W, F814W) CMDs of the WFC2 chip of an IC 1613 image, as produced by HSTphot and DoPHOT