STScI | SPACE TELESCOPE SCIENCE INSTITUTE

Instrument Science Report ACS 2017-08

A Study of PSF Models for ACS/WFC

S.L. Hoffmann, J. Anderson

October 26, 2017

Abstract

This study compares PSFs generated with Tiny Tim against an empirically-derived, effective PSF (Anderson & King, 2006) for the Hubble Space Telescope Advanced Camera for Surveys/Wide Field Channel imaging. We manipulate the Tiny Tim PSF FITS files into a format that can be utilized by the effective PSF FORTRAN photometry code. Then we perform PSF photometry on globular cluster NGC 6397 and analyze the photometry and astrometry results. We measure a value of 0.227 ± 0.032 *for a quality-of-fit metric of the Tiny Tim PSF and a corresponding* 0.117 ± 0.021 *for the effective PSFs, an improvement of a factor of approximately two. We find that the effective PSF models outperform the Tiny Tim PSFs in every measurement of stellar sources in this field.*

1 Introduction

Tiny Tim is a program that creates point spread functions (PSFs) for the various instruments of the *Hubble Space Telescope* (*HST*). Released in 1992, it has served as the primary PSF generation tool for the lifetime of *HST* (Krist et al., 2011). It is hosted and maintained by STScI (main page here: `http://tinytim.stsci.edu`, software download here: `http://tinytim.stsci.edu/sourcecode.php`). It includes updates over the years as new instruments were installed. The history and wide usage of the Tiny Tim PSF model make it a useful benchmark for comparisons with other *HST* PSF models.

Anderson & King (2006, hereafter: AK06) detail the creation of an effective PSF model for ACS/WFC. The effective PSF is constructed by analyzing empirical data, or in this case, observations of stellar sources with ACS/WFC, instead of using instrumental models of the telescope. While AK06 explain their PSFs in depth, they do not make a comparison to Tiny Tim because the PSF models in AK06 are formatted to run with a FORTRAN photometry code, img2xym_WFC.09x10.F, which is also presented in that work and can be found at `http://www.stsci.edu/~jayander/CODE`. The Tiny Tim output was not compatible with this code, and neither was the specific format of the AK06 PSFs compatible with other photometry codes. Therefore, the main challenge of this study was to generate Tiny Tim PSFs that the AK06 FORTRAN code img2xym can access in order to complete this comparison.

In this report, we detail the methodology of how we generate Tiny Tim PSFs in section § 2, explore the results of the PSF photometry (section § 3) and astrometry (section § 4) with both the effective AK06 and Tiny Tim PSFs, and finally summarize our results in section § 5.

2 Creating Tiny Tim PSFs

In order to compare the PSFs of AK06 and Tiny Tim, we required the two types of PSF models to use the same point source photometry code, otherwise our results may have been sensitive to different measurement methods. To this end, we generated Tiny Tim PSFs in a format compatible with the AK06 FORTRAN photometry code img2xym. We use version 7.1 of Tiny Tim in this analysis which follows the same procedures and gives the same results as the most recent 7.5 version because the updates focus on improvements to other instruments, mainly Wide Field Camera 3 (WFC3), and not ACS/WFC.

We started by producing Tiny Tim PSFs at 9×5 fiducial positions across each detector, as depicted in Figure 1 (a reproduction of AK06 Figure 2). First, `tiny1` prompts a list of options available in Tiny Tim and produces a parameter file used to construct the PSF model based on the responses given. We show this list with the inputs we provided to `tiny1` below[1]. We conducted this comparison in F606W because it is one of the most common filters and has the most data available, but this procedure will work for different filters if selected in `tiny1`. The typical jitter for the telescope is 3 to 5 mas as reported in the ACS Data Handbook (Lucas et al., 2016). We added 4 mas of jitter to the PSF by including that parameter in the command calling `tiny1`.

Inputs given to `tiny1`:

- Instrument and camera: 15 - ACS/WFC
- Detector: 1 or 2, depending on position
- Position: Figure 1 gives all input positions
- Filter: F606W

[1]For more information, see the Tiny Tim User Guide here: `http://tinytim.stsci.edu/static/tinytim.pdf`

- Spectrum: Select from a list (option 1), then choose an A0V star (number 7)
- PSF diameter: recommended $3.0''$
- Secondary mirror despace: $-3.0\mu m$, typical value from Figure 9 of Lallo et al. (2006)

We chose A0V for the spectrum input from an estimate of the $V - I$ color of this field for the magnitude range of interest based on Color-Magnitude Diagrams of Anderson et al. (2008). In other works, this would change based on the color of the objects of interest and the science goals of the research.

Next, `tiny2` produces FITS images of the undistorted PSF models, and finally `tiny3` distorts the PSFs in each resultant image to account for detector effects, including the off-axis position of ACS in the focal plane of the telescope. We also command `tiny3` to subsample the models by a factor of 4 to match the AK06 convention. This means that for each native ACS/WFC pixel, `tiny3` creates smaller pixels, a factor of 4 in each direction for a total of 16, which result in a corresponding increase in resolution. A word of caution to readers: AK06 use the term "supersampled" while the Tiny Tim documentation uses "subsampled" when referring to this greater resolution. The terms are completely interchangeable in this particular context. We will use subsampled in this work to reduce confusion when readers consult the Tiny Tim documentation.

Tiny Tim does not convolve the PSF with the charge diffusion kernel when producing subsampled models. The charge diffusion is a physical effect of the detector in which charge migrates into adjacent pixels and causes blurring. Krist (2003) measured the kernel in the native ACS pixel scale which cannot be directly applied to the subsampled PSF models. However, the charge diffusion has an appreciable effect on the PSF and ignoring it would result in an inaccurate comparison because the charge diffusion exists in the empirical data used to build the effective PSF of AK06.

We resolved this by applying the charge diffusion kernel to subsets of the PSF image from Tiny Tim with the same pixel phase. The pixel phase refers to the location within a pixel where the star is centered. The Tiny Tim PSFs are subsampled by a factor of 4 which means each native ACS pixel is now divided into 16 pixels. We select all the pixels that match a particular pixel phase. This arrangement of the subsampled pixels places neighboring pixels in an order which mimics that of the natural ACS resolution. We convolved the charge diffusion kernel and then repeated this 15 times to correct all pixel phases corresponding to the the 4×4 subsampled PSFs. This iteration means that the entire PSF model has been convolved with the charge diffusion kernel.

The charge diffusion kernel is dependent on the location on the detector, and therefore each PSF must be convolved with a different kernel. `tiny3` puts the appropriate kernel in the header of the final FITS product for each PSF. Figure 2 displays an example of the AK06 PSF models, the subsampled Tiny Tim models with no charge diffusion, and the Tiny Tim models with the convolution applied to each pixel phase. The blurring effect of the charge diffusion kernel is clearly visible between the two Tiny Tim models presented.

We then took all the Tiny Tim PSF models and created a single FITS image that combined the PSFs in the same format as Figure 3 from AK06. For each PSF model, we centered the peak of the PSF and kept 50 pixels in each direction, and then placed this

section of the PSF in the order based on the input position as listed in Figure 1. This became a 901×1001 pixel FITS image that contained all PSF models identical to the AK06 PSF format and compatible with the AK06 FORTRAN code img2xym. Figure 3 shows the combined, final version of the Tiny Tim PSF models.

3 Photometry Comparison

We performed PSF photometry with the AK06 FORTRAN code img2xym on a single CTE-corrected F606W exposure of a field of globular cluster NGC 6397 located 5′ SE from the core (j97102x6q_flc.fits, GO-10424, PI: Richer) observed in 2005 and reduced with version 8.3.4 of CALACS. The img2xym code identifies objects of interest with a simple set of finding criteria and then determines the position and flux utilizing the PSF model indicated by the user. First we ran the AK06 code with the PSFs empirically developed for that code, and then we repeated the run with the exact same input parameters but changed the PSF models to the Tiny Tim output we developed. Figure 4 shows the resulting PSF quality-of-fit (QFIT) metric as a function of instrumental magnitude for the AK06 PSF ("EFFPSF") on the left and for the Tiny Tim PSFs ("TTMPSF") on the right. QFIT is the absolute value of the residuals of the fit to a star's inner 5x5 pixels divided by the flux of the star (Sohn et al., 2012). It is essentially the fractional error in the PSF fit to the star.

In Panels A and B of Figure 4, we attribute the large cloud of points above the main distribution to galaxies and spurious detections (such as cosmic rays), and in Panels C and D, we visually select the well-behaved sequence of stars for further analysis, shown in red. We isolate and zoom in on these sources in Panels E and F, and we can see a qualitative improvement in the fits to the stars using the empirical PSF models by visual inspection. We calculated the statistics for instrumental magnitudes of $-14 < M_{F606W} < -10$ (a range from just below saturation at -14 down to a S/N ~ 100 at -10) and found that the effective PSF from AK06 has a median QFIT value of 0.117 ± 0.021. The Tiny Tim PSF fits obtained are about a factor of two worse with a median QFIT value of 0.227 ± 0.032. We note that we did not apply a perturbation to optimize the PSF fitting by matching the focus variation of this particular image, and therefore more precise photometry might be attained. It is not necessary in this case because we are interested in the relative comparison between the unmodified PSF models.

4 Astrometry Comparison

We also tested the astrometry of the two PSFs. We performed photometry on ten F606W calibrated images (observation sets j97112020_asn.fits and j97113020_asn.fits where each contains five images) from the same NGC 6397 observation program as before using both the TTMPSF and EFFPSF models.

We analyzed the photometry results by measuring the coordinate transformations between the ten images with stars that do not have any large or anomalous shifts in positions between frames. With this information, we removed global shifts in the positions and calculated an average position and residual for each source. We derived the RMS with the

residual to describe the typical measurement errors which we display in Figure 5 as a series of RMS versus instrumental magnitude plots. We find that the EFFPSF results in the left column show a tighter relation when compared to the TTMPSF results in the right column.

This is a similar result to the single-image analysis in § 3. The cloud of galaxies and spurious detections that appeared in Figure 4 was naturally removed when we required consistent positions between different frames. Additionally, we expected to see an improvement in the scatter of the magnitude distributions due to the increase in the number of images used to calculate the errors, and indeed, we demonstrate a decrease in the scatter and offset of the instrumental magnitudes in the bottom row of Figure 5.

Lastly, we examined the pixel phase to determine if the location on the pixel where the peak of the source falls changes the reported position in a systemic way. The photometry measures subpixel precision, and we calculated the pixel phase by subtracting the integer value of the position and then shifting the values by 0.5 as seen in this example formula for the X position: phase(x) = x−int(x+0.5). The function int() in python truncates the number after the decimal and returns the integer value. The 0.5 shift is a matter of convention to plot the phase from -0.5 to 0.5 in order to have $(0, 0)$ representing the center of the pixel.

We plot the pixel phase against the residuals in X and Y for each PSF model in Figure 6 for sources with an instrumental magnitude of -12 to -11. We restrict the magnitude range in order to see the effect of the pixel phase without additional trends due to differences in S/N of the sources. The residuals show no trend as a function of pixel phase for the EFFPSF in the left column. However, in the TTMPSF in the right column, we see structure in the residuals that indicate systematic shifts in the reported position based on the pixel phase. This type of bias in the pixel phase is often the result of the PSF model being sharper or duller than the true PSF in the image (Anderson & King, 2000).

5 Conclusion

We developed a set of Tiny Tim PSF models compatible with the PSF photometry code described in AK06 for a comparison of the two PSFs. We analyze unsaturated point sources in an uncrowded field of globular cluster NGC 6397 with a typical dynamic range for ACS/WFC instrumental magnitudes of $-14 < M_{F606W} < -10$.

We note that this experimental setup does have limitations. We chose the NGC 6397 field because it contains a large stellar population with a wide range of magnitudes, but we did not examine sources beyond the saturation limit of the detectors. Tiny Tim allows users to change the size of the PSF during construction, and therefore it is reasonable to assume that an extended PSF from Tiny Tim would be better at fitting the diffraction spikes associated with these luminous objects. Additionally, optimizing more of Tiny Tim's parameters following a procedure similar to Biretta (2014) could potentially result in a better fitting PSF. Finally, we did not test the photometry on any stellar populations with crowding as that is beyond the scope of this ISR.

With these caveats, we conclude that the empirically derived effective PSF models from AK06 provide better fits to stars for both the photometry and astrometry measurements than the corresponding Tiny Tim PSFs.

Acknowledgements

This research made use of Astropy, a community-developed core Python package for Astronomy (Astropy Collaboration et al., 2013), as well as the Python-based data interface Glueviz, which can be found at `http://glueviz.org/`.

References

Anderson, J., & King, I. R. 2000, PASP, 112, 1360

—. 2006, PSFs, Photometry, and Astronomy for the ACS/WFC, Tech. rep., STScI

Anderson, J., King, I. R., Richer, H. B., et al. 2008, AJ, 135, 2114

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, A&A, 558, A33

Biretta, J. 2014, Improved TinyTIM Models for WFC3/IR, Tech. rep., STScI

Krist, J. 2003, ACS WFC & HRC field-dependent PSF variations due to optical and charge diffusion effects, Tech. rep., STScI

Krist, J. E., Hook, R. N., & Stoehr, F. 2011, in Proc. SPIE, Vol. 8127, Optical Modeling and Performance Predictions V, 81270J

Lallo, M. D., Makidon, R. B., Casertano, S., & Krist, J. E. 2006, in Proc. SPIE, Vol. 6270, Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series, 62701N

Lucas, R. A., et al. 2016, ACS Data Handbook v.8.0 (STScI)

Sohn, S. T., Anderson, J., & van der Marel, R. P. 2012, ApJ, 753, 7

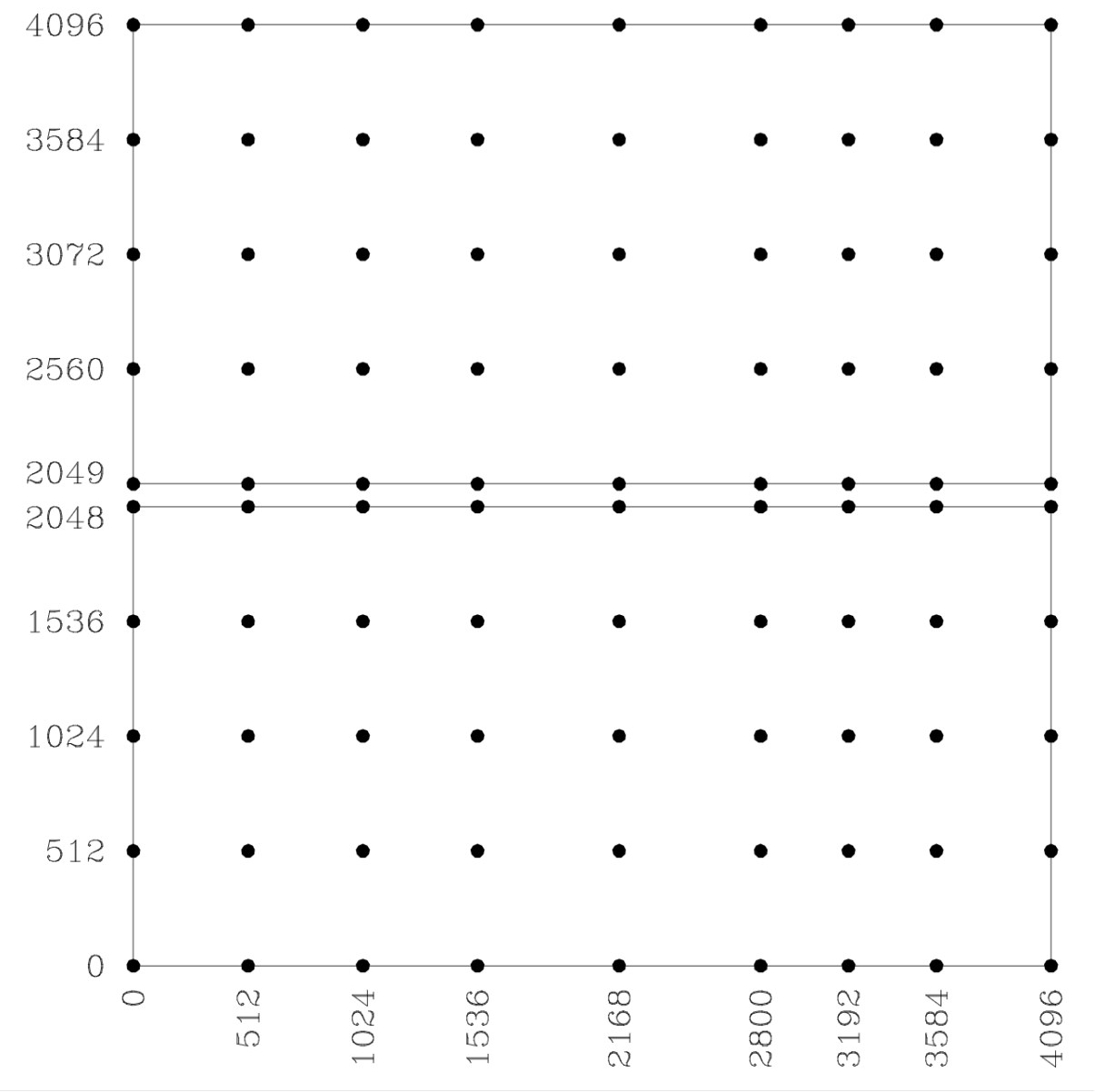

Figure 1 - A reproduction of Figure 2 from AK06 that displays all the X&Y positions in pixels on the detectors given as inputs to Tiny Tim. A PSF model is generated for every point of the grid.

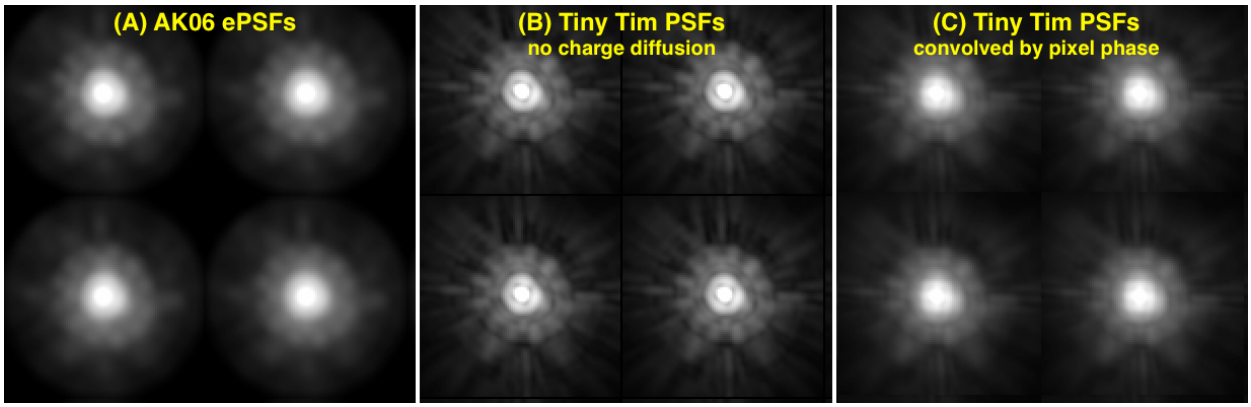

Figure 2 - PSF models for the same four positions on ACS/WFC displayed with identical scaling and stretch for comparison. Panel A shows the effective PSFs from AK06 derived from empirical data. Panel B displays the subsampled Tiny Tim PSFs without the charge diffusion kernel applied. Finally, Panel C shows the version of the Tiny Tim PSFs with the pixel phase convolution that corrects for the charge diffusion.

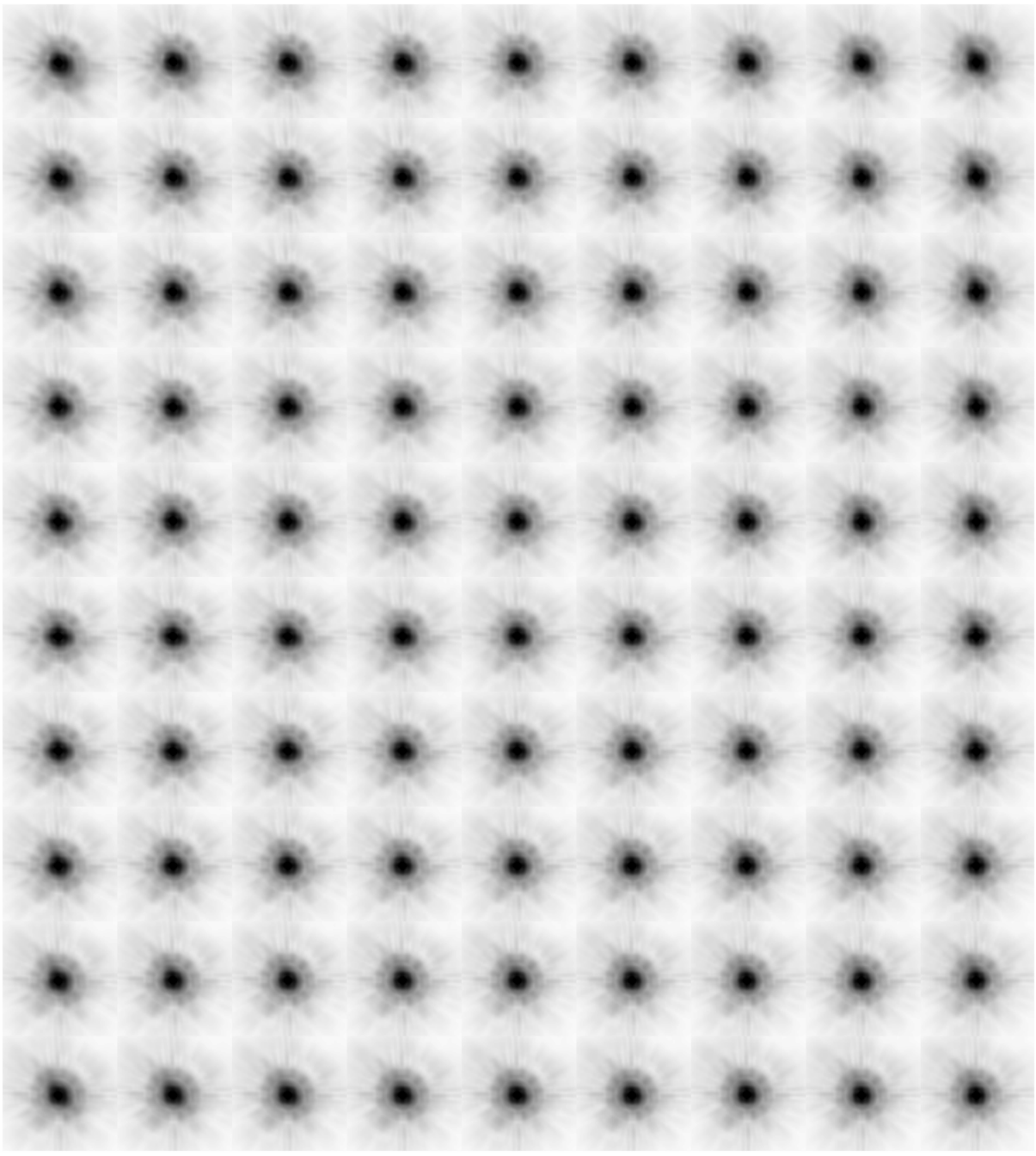

Figure 3 - The final, formatted output of the Tiny Tim PSF models. We combined the PSFs into a 901 × 1001 pixel FITS image sorted to match the locations given in Figure 1.

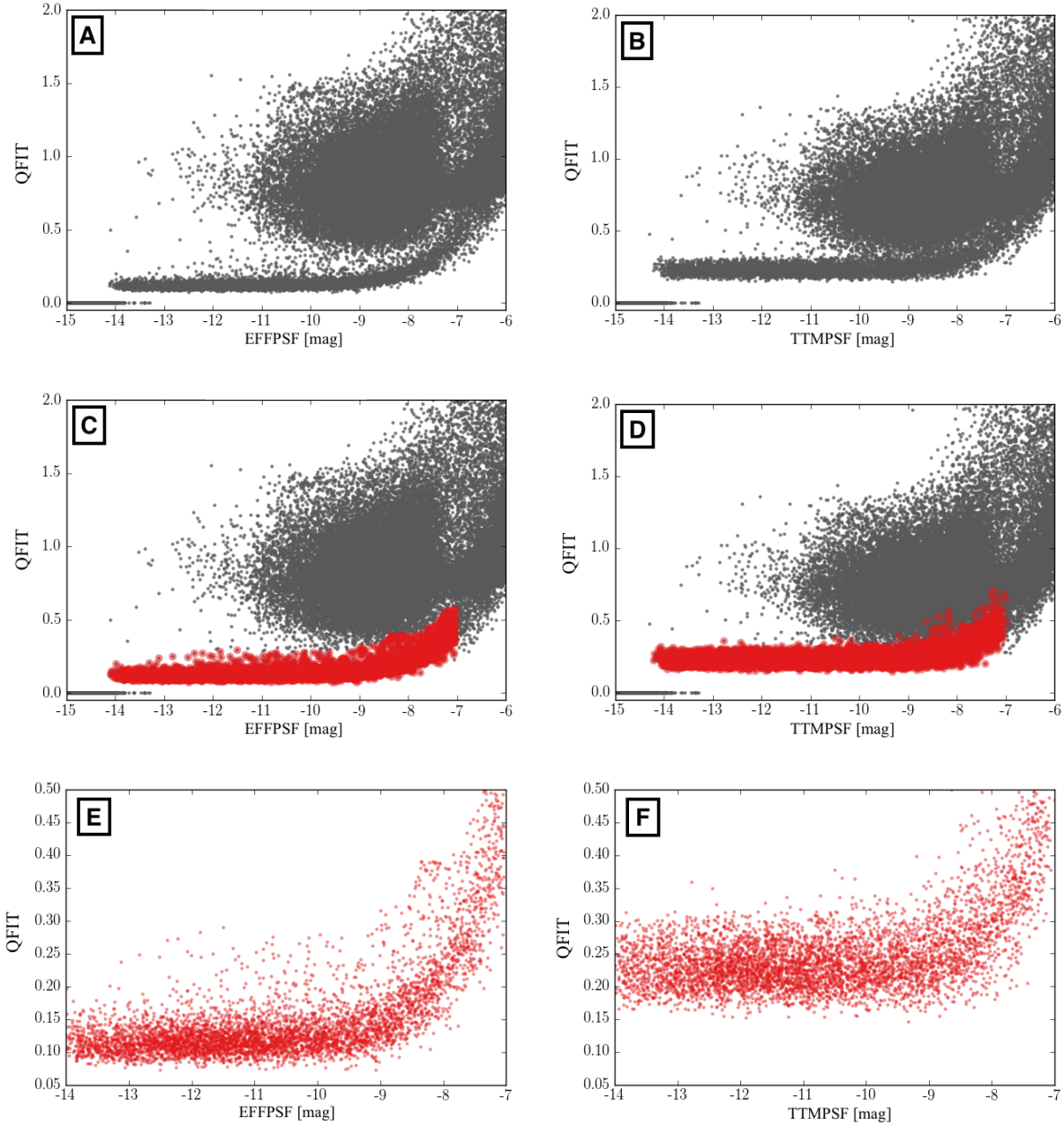

Figure 4 - We show a grid of PSF photometry instrumental magnitude versus QFIT plots from a single F606W image of a field in globular cluster NGC 6397. The QFIT parameter is an output of the AK06 code that is a proxy for the PSF-fitting error. The photometry measured with the AK06 effective PSF is labeled EFFPSF (left column), and the Tiny Tim PSF photometry is labeled TTMPSF (right column). Panels A and B show all the photometric points measured, panels C and D identify a well-behaved sequence of stars plotted in red, and panels E and F zoom in on only the well-behaved red sources to highlight the comparison of the QFIT.

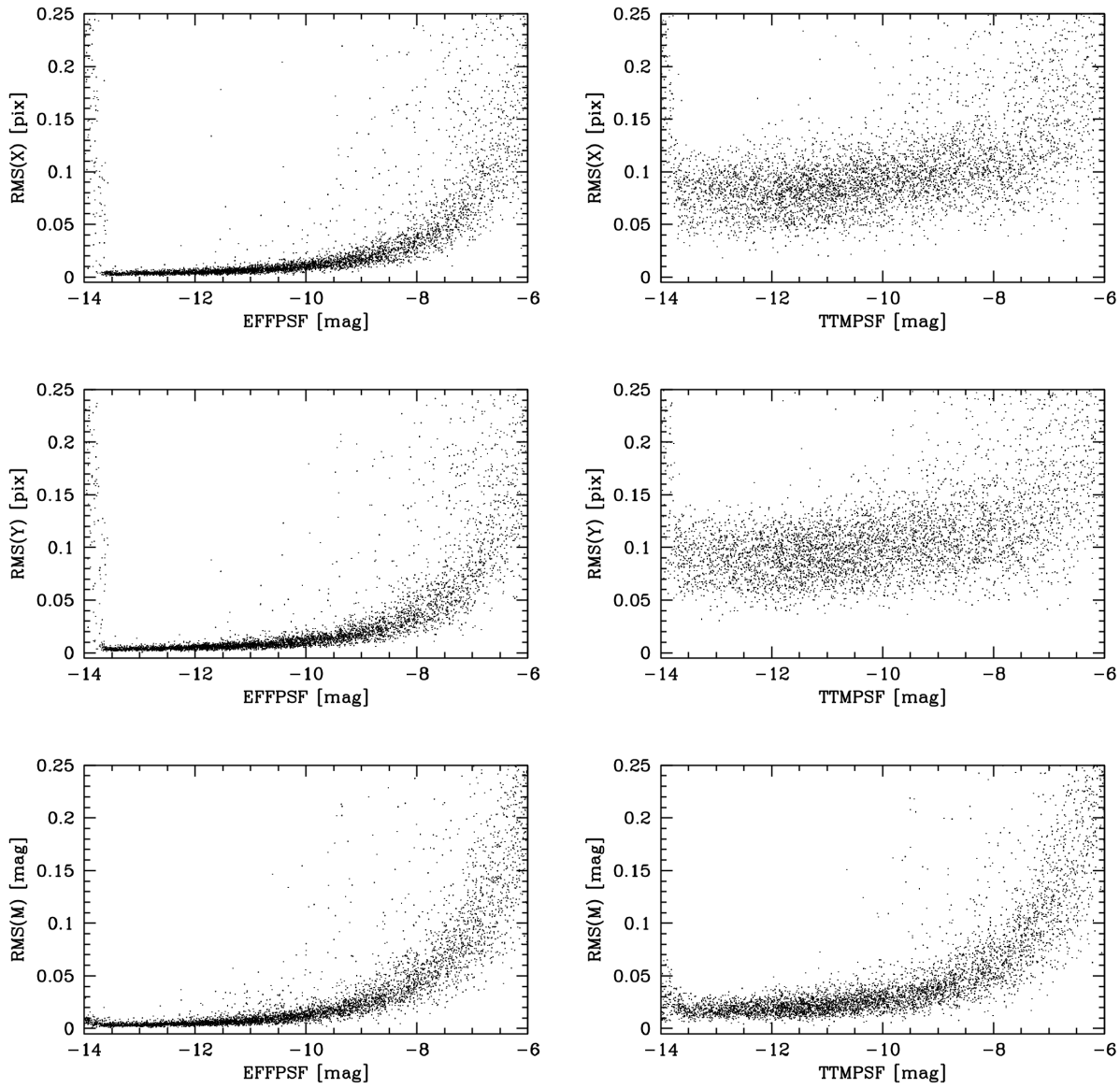

Figure 5 - Here we display the RMS calculated from the photometry of 10 F606W images of globular cluster NGC 6397 as a function of the instrumental magnitude for the X positions (top row), the Y positions (middle row), and the magnitude (bottom row). We labeled the photometry analyzed with the AK06 effective PSF models as EFFPSF (left column), and the corresponding measurements with Tiny Tim PSF models as TTMPSF (right column).

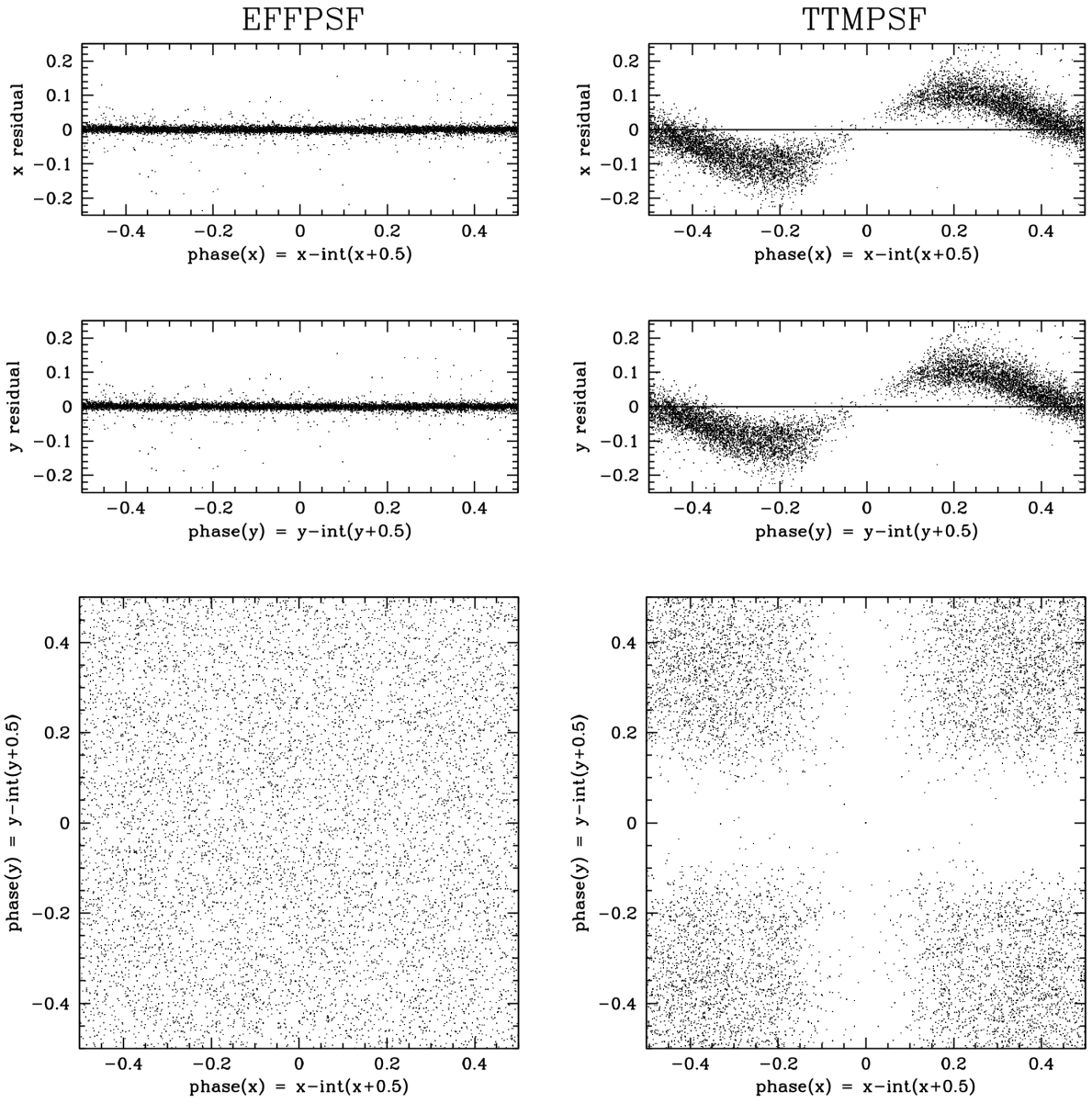

Figure 6 - We compare the position residuals of EFFPSF (left column) and TTMPSF (right column) with respect to the pixel phase for sources with magnitudes of -12 to -11. All position measurements are in pixel units, and we give the formula to calculate the pixel phase on the appropriate axis labels. The top row has the X position residuals plotted against the phase, with the Y plots shown in the middle row. The bottom row has the X phase versus the Y phase. The TTMPSF contain structure not seen in the EFFPSF demonstrating that the TTMPSF astrometry results are not pixel phase invariant.