DOLPHOT/WFPC2 User's Guide

version 2.0

Andrew Dolphin

adolphin@raytheon.com

October 2011

http://purcell.as.arizona.edu/dolphot/

Contents

1 Introduction

This manual describes installation and use of the WFPC2 module for DOLPHOT. The WFPC2 module replaces the analytic PSF model with a lookup table computed using Tiny Tim PSFs. It also includes built-in CTE corrections and photometric calibrations, and is intended to replace HSTphot.

If you have any problems installing or using DOLPHOT or the provided utilities, please let me know.

Andrew Dolphin
adolphin@raytheon.com

2 Installation

Installation of the WFPC2 module is done in addition to the regular DOLPHOT installation. See the DOLPHOT manual for installation instructions for DOLPHOT.

The WFPC2 module requires approximately 1Mb for the sources and binaries, plus 97Mb per set PSFs. Note that the PSFs are distributed separately, as you only need PSFs for the filters that you will actually be using.

The .tar.gz file for the WFPC2 module (as well as the PSFs) should be expanded from the same directory from which the DOLPHOT distribution was expanded. Note that most WFPC2 PSFs are unzipped into a directory named "dolphot1.1/wfpc2/data". To move these to the correct location, the following command should be used after unzipping all needed PSFs:

```
mv dolphot1.2/wfpc2/data/*.fits dolphot2.0/wfpc2/data/
```

2.1 Makefile

The Makefile is the same as the regular DOLPHOT installation, but includes several options used by the WFPC2 module. To enable the WFPC2 module, the three lines beginning with the WFPC2 definition should be uncommented.

No customization of the Makefile within the "wfpc2" subdirectory is needed.

If you plan to compute your own PSFs, uncomment out the TTDIR line and specify where Tiny Tim is installed. (Tiny Tim is an HST PSF simulator written by John Krist and Richard Hook, and can be obtained at http://www.stsci.edu/software/tinytim/tinytim.html.) Also, if you will be computing your own PSFs, and csh is not located in /bin/csh (type "which csh" to find out), then you will need to edit the file "wfpc2/wfpc2ttpar.c", changing "/bin/csh" to the correct location of csh.

After editing the makefile, you should be able to type "make" to compile DOLPHOT without any error or warning messages. Should you need to later recompile the programs, typing "make clean" will remove the original compilations to ensure that all programs get recompiled.

2.2 PSFs

You can either make your own PSFs or use the precomputed ones. Generally there is no need to generate your own, as precomputed PSFs are available for all filters. Instructions for computing PSFs are not currently available.

3 Preprocessing Steps

For the WFPC2 module to run correctly, data should be in the format they are obtained from the STScI archive (both the multi-extension and 4x800x800 image formats are supported). Because drizzled images produce suboptimal photometry (because of resampling of the images), support exists only for _C0F or _C0M datasets. DOLPHOT includes multiple-image support (including offset and rotation), so photometry can be run on multiple images of the same field, eliminating the primary purpose for photometering drizzled data.

This section takes you through the necessary steps (in order) for obtaining the best possible photometry from DOLPHOT.

3.1 Create/Select Reference Images

If you have only one exposure per filter to photometer, and all images are taken at the same pointing, you can skip this step. Simply use your deepest image as the reference image in DOLPHOT.

It is possible that the WFPC2 reduction pipeline has already drizzled your images into a single drizzled image per filter (more likely, there are multiple drizzled images per filter). If this is the case, you can also forgo running drizzle, and use the deepest drizzled image as your reference frame.

Otherwise, it is likely that you will want to use multidrizzle to create a clean, deep reference frame. Instructions on multidrizzle can be found in the WFPC2 data handbook, and are beyond the scope of this manual. However, it should be noted that multidrizzle does not work well on images of greatly differing exposure times. Thus, if you have short and long exposures, they should probably be drizzled separately. Once you are done, the deepest drizzled image will be used as your reference frame.

Once drizzling is complete, you may want to co-add the _FLT images taken at the same pointing using the DOLPHOT utility *imcombine*. (The combination of drizzle and *imcombine* is often superior to the CR-cleaned _CRJ images that come from the WFPC2 pipeline, and is thus preferable.) When you use *imcombine* for this purpose, make sure that the registration factor and sigma clipping are set sufficiently high that no additional cleaning is done, beyond the cleaning done by multidrizzle. It is also perfectly acceptable to photometer all of the _FLT images separately.

3.2 *wfpc2mask*: Mask Bad Pixels and Multiply PAM

```
wfpc2mask <fits file> <data quality file>

wfpc2mask -exptime=<exptime> <fits file> <data quality file>
```

WFPC2mask takes as input FITS files in the format used by the STScI archive. The data quality images (c1f.fits) are also required. *WFPC2mask* masks out all pixels flagged as bad in the data quality image, and multiplies by the pixel areas so that the resulting images are approximately in units of electrons on the raw image.

The `-exptime` flag will override the exposure time in the header. This is needed for images combined in the drizzling process, as the exposure time in the FITS header is generally just copied from one of the images.

Before running *wfpc2mask*, make sure you have backed up the original images, since *wfpc2mask* will alter them!

A masked or saturated pixel is skipped by all other HSTphot routines - it is not used in sky determination, photometry, aperture corrections, etc.

The output image will be in the four-extension format (even if the original image was in the 4x800x800 image format), with PC1 in the first extension and WFC2-4 in extensions 2-4, respectively.

3.3 *splitgroups*: Split UVIS Chips

If you are using a drizzled reference image, you will need to split the image files (which contain all four chips) into four files, one for each chip.

Assuming you have put your data into a working directory that contains only the data you will be photometering, typing `splitgroups *.fits` should convert all of your FITS files to single-chip format.

3.4 *calcsky*: Calculate Sky for Image

`calcsky <fits base>` $< r_{in} >$ $< r_{out} >$ `<step>` $< \sigma_{low} >$ $< \sigma_{high} >$

Calcsky is described in the DOLPHOT manual. Recommended parameters for WFPC2 data are as follows:

$r_{in} = 10$

$r_{out} = 25$

$step = 2$ for an accurate sky map, or $step = -50$ for a quick one.

$\sigma_{low} = 2.25$

$\sigma_{high} = 2.00$

Since I always use FitSky=1 or FitSky=3 in my photometry, the negative values of *step* are sufficient.

Note that running *calcsky* is currently mandatory, even if using FitSky settings of 3 or 4.

3.5 *wfpc2fitdistort*: Calculate alignment

`wfpc2fitdistort <camera> <Xref> <Yref> -file=<input file> -XY`

This utility provides reasonable initial guesses for the alignment process. The *camera* flag should equal 0 for the PC, 1 for WFC2, 2 for WFC3, and 3 for WFC4. The X and Y parameters should be the dimensions of the reference frame (normally a drizzled frame).

The input file can be omitted, in which case *wfpc2makepsf* takes its input from standard input.

The -XY flag is optional, and will compute the X,Y shift only (instead of computing X,Y shift, magnification, and rotation).

The input contains the X and Y positions of stars on the image and on the reference image, with one star per line and a format of Ximage Yimage Xref Yref. At least three stars must be used. As a rule of thumb, four stars (one in each corner of the chip being aligned) generally provides excellent results. The output will give the dx, dy, scale, and rotation, which you can place into the dolphot parameter file.

When entering the positions, note that DOLPHOT coordinates are shifted by (-0.5,-0.5) relative to IRAF coordinates. The center of the lower-left pixel has a position of (0.5,0.5). If the data image and reference image have roughly the same orientation (which is typical), the offset of 0.5 doesn't matter since both sets of coordinates are offset by roughly the same amount in the same direction.

As a sanity check, the transformation should be similar between the four chips.

The resulting shift, magnification, and rotation should be given to DOLPHOT, either on the command line ("img1_shift = 30 40" and "img1_xform = 0.9994 0 1.35" for a shift of dX=30 and dY=40, magnification of 0.9994, and rotation of 1.35 degrees) or as part of the parameter file.

Note that, if using UseWCS=1 or UseWCS=2 in the DOLPHOT parameters, this step can often be skipped. In this case, the alignment lines should be "img_shift = 0 0" and "img_xform = 1 0 0".

4 Running DOLPHOT

4.1 DOLPHOT Parameters

The WFPC2 module of DOLPHOT will be invoked whenever processing images that have been run through wfpc2mask. When using the WFPC2 module, several parameters are disabled or have restricted ranges.

img_aprad is set to 0.5 arcsec

img_RPSF and img_RAper cannot exceed 16 pixels. If running photometry on the PC only, these can be 28 pixels.

Zero disabled (set to VEGAMAG zero points)

FPSF, SubPixel, img_psfa, img_psfb, img_psfc disabled

EPSF, PSFsol, PSFStep disabled

MinS, MaxS, MaxE disabled

Recommended values for other parameters are as follows. These refer to processing of the PC image. When processing WFC images, img_RAper, img_RSky, img_RPSF, img_apsize, and img_apsky are automatically divided by 1.5. This allows reasonably comparable photometry to be obtained for all chips with the same parameters. Also note that the `img_shift` values for all four chips are for the scale and rotation angle of the PC.

```
img_apsky = 15 25

img_RAper = 4

img_RChi = 2.0

img_RSky = 15 35

SkipSky = 2

SkySig = 2.25

SecondPass = 5

SigFindMult = 0.85

MaxIT = 25

NoiseMult = 0.10

FSat = 0.999

ApCor = 1
```

```
RCentroid = 2

PosStep = 0.25

RCombine = 1.5

img_RPSF = 10

SigPSF = 5.0

PSFres = 1
```

The best overall combination of solution parameters are `PSFPhot = 1`, `FitSky = 1`, and `Force1 = 0`. Generally, `Align = 2` and `Rotate = 1` provide the best alignment. If using `FitSky = 3`, the parameters are largely the same except that the suggested `img_RAper` value is 8 pixels.

In some cases, blends or extended objects can hamper photometry of nearby stars. Setting `Force1 = 1` will solve this, but of course will also result in false detections of hot pixels and extended objects as single stars. So one may have to choose between a very clean but slightly incomplete CMD and a complete but contaminated one.

Finally, there are four new parameters that can be used:

```
WFPC2useCTE = 1

FlagMask = 4

CombineChi = 0

InterpPSFlib = 1
```

WFPC2useCTE determines whether CTE corrections should be applied by DOLPHOT (1=yes, 0=no).

FlagMask is a bitwise mask that determines what error flags will not be accepted when producing the combined photometry blocks for each filter. Note that error flag values of eight or more (when the "extreme case") always cause the photometry to be ignored. A value of zero allows photometry with an error flag less than eight to be used. Adding one eliminates stars close to the chip edge, adding two eliminates stars with too many bad pixels, and adding four eliminates stars with saturated cores.

CombineChi also affects the combined photometry blocks. If set to zero (default), photometry will be combined weighted by $1/\sigma^2$ to maximize signal to noise. If set to one, weights will be $1/\sigma^2 max(1, \chi^2)$ to reduce the impact of epochs with bad photometry. Note that using CombineChi of one will require tuning NoiseMult so that well measured stars have $\chi = 1$ at all magnitudes (plots of chi vs. magnitude should show this).

If InterpPSFlib is set to 0, the PSF library will use the nearest X,Y position where a precalculated PSF is available rather than interpolating. The impact is $\approx 1\%$ on the PSF shape but some speed improvement.

Note that alignment is treated specially if you are using a drizzled reference image for alignment. In this case, the geometric distortion corrections are automatically applied to

the image, so that you need to provide only the offset between the distortion-corrected image center and the drizzled image. Initial guesses for the alignment parameters can be calculated using the *wfpc2fitdistort* utility.

4.2 DOLPHOT Output

The DOLPHOT/WFPC2 output is virtually identical to the regular DOLPHOT output, with a few exceptions. First, if multiple images exist per filter, it will insert extra sets of photometry for each such filter (in order of wavelength) prior to the individual-image photometry. For example, if your data consist (in order) of F555W, F439W, F814W, F555W, and F439W images, the output photometry will have the combined F439W photometry first, combined F555W photometry second, and last, the five sets of single-image photometry.

Second, two magnitudes are computed per photometry set. First is the usual instrumental magnitude, and second is a transformed magnitude. Instrumental magnitudes, transformations, and CTE corrections are made using the values posted at http://purcell.as.arizona.edu/wfpc2_calib/.

As with all photometry, you will need to trim your detection list to eliminate objects classified as non-stellar, low signal-to-noise, or bad photometry quality. See the DOLPHOT manual for further discussion.

5 Artificial Star Tests

Once the photometry has been run to your satisfaction, you will likely want to measure completeness, as well as photometric errors due to blending. This is done with fake star tests.

5.1 Create artificial star list

Unlike the ACS and WFC3 packages, no utility exists to automatically generate the artificial star list. This is because the input format for artificial stars has been simplified. The file should contain one line per artificial star, containing the following data:

extension number

chip number (almost always 1)

X position

Y position

instrumental magnitude in each filter present in the data (in order of increasing filter wavelength)

5.2 Run Artificial Stars

In the same directory as *DOLPHOT* was originally run, you will now want to rerun *DOLPHOT*, while specifying the additional parameter, `FakeStars`, which is set to the filename created by *wfpc2fakelist*. If you want the PSFs to include the PSF residuals, you should also use the parameter `FakeStarPSF=1` when re-running *DOLPHOT*.

Note that running the artificial star tests can take a very long time. Once completed, you will have a `.fake` file that lists the input values for every star, as well as the complete *DOLPHOT* photometry output.

6 PSF Utility Reference

The following programs are utilities included with the WFPC2 module. Since there is little need for computing custom PSFs, no documentation is provided – use at your own risk!

6.1 *wfpc2makepsf*: Compute PSF for Filter

```
wfpc2makepsf <filter> <chips>
```

6.2 *wfpc2showpsf*: Display PSF for Filter

```
wfpc2showpsf <filter> <chip> <X/256> <Y/256> <dX> <dY>
```

6.3 *wfpc2ttpar*: Make Tiny Tim Parameter Files and Script

```
wfpc2ttpar <filter>
```

7 Multi-camera Photometry

Beginning with DOLPHOT 2.0, the capability exists to simultaneously run photometry of data taken with multiple HST cameras, such as ACS/WFC and WFPC2/IR. For the most part, this is as simple as pre-processing each image using the appropriate steps for the camera. That is, ACS/WFC images should be preprocessed with wfpc2mask and aligned to the reference using wfpc2fitdistort, while the WFPC2/IR images would be preprocessed with wfpc2mask and aligned using wfpc2fitdistort.

Three points deserve mention. First is the selection of a reference frame. While this is normally a simple matter of selecting the filter with the deepest photometry, in a multi-camera mode one must consider wavelength and pixel size. As an example, WFPC2/IR images may serve as poor reference images if combined with optical data due to the different PSF size and the fact that the IR images are generally much more crowded. Likewise, the capability exists to run simultaneous photometry of images from 275nm to 1.6 microns, so it is useful to select an intermediate wavelength where a sufficient number of stars on the reference are likely to be observed in all filters.

The second point is treatment of different PSF sizes and resolutions. The ability to set img_RPSF, img_RAper, and img_RChi by image mitigates much of this, since each image can have appropriate PSF and photometry apertures set. In practice, a good value RCombine can be very tricky to find since it is possible to have multiple UVIS or ACS detections within the size of an IR PSF.

Finally, it is worth noting that the distortion corrections are not as self-consistent between the cameras as they are within a single camera. Thus, a higher-order alignment solution (Align=4) may be needed to minimize residuals. In a similar vein, UseWCS may not prove to be a perfect solution to the alignment issues, since the WCS solution on ACS may not match that on WFPC2.

8 Recipe for WFPC2 Photometry

Assuming that an appropriate reference image exists, the recipe for generating photometry is reasonably simple.

1. Run wfpc2mask on all WFPC2 images.
2. If possible (images with same filter and similar pointings), run crrej to identify and mask cosmic rays from the images. This may need to be run multiple times for large numbers of images.
3. If desired (images with same filter and pointing, and epoch if performing variable star work), run imcombine to combine the CR-cleaned images
4. For WFPC2/UVIS image, use splitgroups to split into *.chip1.fits and *.chip2.fits files.
5. Run calsky on each image to generate *.sky.fits files.
6. If not using UseWCS of 1 or 2 in DOLPHOT, run wfpc2fitdistort to compute the alignment of each image to the reference.
7. Run dolphot.
8. If artificial star tests are to be run, generate the star list and re-run dolphot with the FakeStar parameter used to trigger artificial star mode.

9 Changes to DOLPHOT/WFPC2

October 16, 2011 release

1. Initial release

April 29, 2012 release

1. Fixed bug causing crashes

October 14, 2012 release

1. Fixed bug in wfpc2mask's processing of drizzled images
2. Fixed bug that prevents WFPC2 library from adjusting RChi on a per-chip basis.

March 31, 2013 release

1. Added feature to generate WCS estimates when running c0f format files through wfpc2mask

January 13, 2014 release

1. Minor robustness update to CTE corrections

October 3, 2014 release

1. Fixed error in wfpc2mask's recognition of single-chip drizzled images

November 9, 2014 release

1. Added ability to choose whether PSF library is interpolated or not

January 10, 2016 release

1. Added CombineChi flag