DOLPHOT/WFC3 User's Guide

version 2.0

Andrew Dolphin

adolphin@rtx.com

October 2011

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

Contents

1 Introduction

This manual describes installation and use of the WFC3 module for DOLPHOT. The WFC3 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 thus serves as an WFC3 adaptation of HSTphot.

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

Andrew Dolphin
adolphin@rtx.com

2 Installation

Installation of the WFC3 module is done in addition to the regular DOLPHOT installation. See the DOLPHOT manual for installation instructions for DOLPHOT.

The WFC3 module requires approximately 69Mb for the sources, binaries, and pixel area maps (PAMs), plus 50Mb per set of IR PSFs and 346Mb per set of UVIS 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 WFC3 module (as well as the PAMs and PSFs) should be expanded from the same directory from which the DOLPHOT distribution was expanded. Note that most WFC3 PAMs and PSFs are unzipped into a directory named "dolphot1.1/wfc3/data". To move these to the correct location, the following command should be used after unzipping all needed PSFs and PAMs:

```
mv dolphot1.1/wfc3/data/*.fits dolphot2.0/wfc3/data/
```

2.1 Makefile

The Makefile is the same as the regular DOLPHOT installation, but includes several options used by the WFC3 module. To enable the WFC3 module, the three lines beginning with the WFC3 definition should be uncommented.

No customization of the Makefile within the "wfc3" 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 "wfc3/wfc3ttpar.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 WFC3 module to run correctly, data should be in the format they are obtained from the STScI archive. Because drizzled images produce suboptimal photometry (because of resampling of the images), support exists only for _CRJ and _FLT 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 WFC3 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 and/or to cosmic ray-clean your images. Instructions on multidrizzle can be found in the WFC3 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 WFC3 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   *wfc3mask*: Mask Bad Pixels and Multiply PAM

```
wfc3mask <fits files>

wfc3mask -keepcr <fits files>

wfc3mask -exptime=<exptime> <fits files>

wfc3mask -ncombine=<ncombine> <fits files>

wfc3mask -maskflat <fits files>
```

```
wfc3mask -usewht <fits files>
```

WFC3mask takes as input FITS files in the format used by the STScI archive. (IR images are 3-extension files; UVIS images are 6-extension files. See the instrument handbook for details on what is contained in the extensions.) *WFC3mask* 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 **-keepcr** flag will cause wfc3mask to keep pixels containing repaired cosmic rays (they are masked by default). This is useful if there are more than two subimages; otherwise it is generally best for the pixel to be ignored.

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.

The **-ncombine** flag will override the NCOMBINE setting in the header. This is used to determine the correct readout noise, which is set to the single-image read noise times the square root of NCOMBINE.

The **-maskflat** flag will cause pixels to be masked if the data quality image contains the "Bad or uncertain flat value" flag. This is not used by default since, in practice, pixels with this flag set are usually good.

The **-usewht** flag will use the weight extension instead of the context extension when masking drizzled images. This option is available because multidrizzle can be run in a manner in which the context extension is not generated.

Before running *wfc3mask*, make sure you have backed up the original images, since *wfc3mask* 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 data quality and noise extensions are deleted when writing the data back to disk. (The data quality image is no longer needed; inconsistent routines in the STScI pipeline render the noise images useless.) The output image will thus have one extension if it is an IR or drizzled image, or two if a UVIS image. Note that UVIS2 is the first extension and UVIS1 is the second extension.

3.3   *splitgroups*: Split UVIS Chips

If you are using a drizzled reference image, you will need to split the UVIS image files (which contain both chips) into two 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> < σ_low > < σ_high >
```

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

$r_{in} = 15$

$r_{out} = 35$

$step = 4$ for an accurate sky map, or $step = -128$ for a quick one.

$\sigma_{low} = 2.25$

$\sigma_{high} = 2.00$

Recommended parameters for IR data are as follows:

$r_{in} = 10$

$r_{out} = 25$

$step = 2$ for an accurate sky map, or $step = -64$ for a quick one.

$\sigma_{low} = 2.25$

$\sigma_{high} = 2.00$

Since I always use FitSky=1, 2, or 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 *wfc3fitdistort*: Calculate alignment

```
wfc3fitdistort <camera> <filter> <Xref> <Yref> -file=<input file> -XY
```

This utility provides reasonable initial guesses for the alignment process. The *camera* flag should equal 0 for the IR, 1 for UVIS1, and 2 for UVIS2. (Note again, UVIS1 is the second chip in the FITS file, and UVIS2 is the first!) 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 *wfc3makepsf* 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, if aligning WFC3/UVIS images, the transformation should be similar between the two 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 WFC3 module of DOLPHOT will be invoked whenever processing images that have been run through wfc3mask. When using the WFC3 module, several parameters are disabled or have restricted ranges.

img_aprad is set to 0.5 arcsec

img_RPSF and img_RAper cannot exceed 41 pixels (UVIS) or 19 pixels (IR)

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 general-use settings for other parameters follow. Note that these were optimized across a variety of environments ranging from uncrowded to extremely crowded. For uncrowded fields, other parameter options involving larger photometry radii (`img_RAper` and `img_RChi`) and other sky options (`FitSky = 1` or `FitSky = 3`) can produce very good results.

Recommended values for other parameters for WFC3/UVIS are as follows:

```
PSFPhot = 1

PSFPhotIt = 2

Force1 = 0

FitSky = 2

img_RAper = 3

img_RChi = 2

img_RPSF = 15

img_RSky2 = 4 10

img_RSky = 15 35

img_apsky = 15 25

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

SigPSF = 5.0

PSFres = 1
```

For WFC3/IR, recommended values that differ from the above are as follows:

```
img_RAper = 2

img_RChi = 1.5

img_RPSF = 15

img_RSky2 = 3 10

img_RSky = 8 20

img_apsky = 8 20

SkipSky = 1
```

Generally, `Align = 2` and `Rotate = 1` provide the best alignment. If using `FitSky = 1` or `Fitsky = 3`, the parameters are largely the same except that suggested `img_RAper` values are 8 pixels for UVIS and 5 pixels for IR.

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 six new parameters that can be used:

```
WFC3useCTE = 0

FlagMask = 4

CombineChi = 1
```

```
WFC3IRpsfType = 0

WFC3UVISpsfType = 0

InterpPSFlib = 1
```

WFC3useCTE determines whether CTE corrections should be applied by DOLPHOT (1=yes, 0=no). If you are using CTE-corrected images (which is recommended), this should be set to zero since CTE corrections are inherent in the input data.

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). Note also that this will result in larger uncertainties for combined (but not individual image) magnitudes and normalized count rates, as the individual image uncertainties are effectively multiplied by χ when calculating combined magnitudes.

If WFC3IRpsfType is set to 1, the PSF library using the Anderson PSF cores (without spatial variation) will be used on WFC3/IR data, if available for the filter. The WFC3UVISpsfType flag works similarly for UVIS data. If WFC3IRpsfType is set to 2, the PSF library using the 2015 Anderson PSF cores (with spatial variation) will be used on WFC3/IR data. Note these are not currently recommended, due to issues seen in the DOLPHOT implementation of these PSF libraries.

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 *wfc3fitdistort* utility.

4.2   DOLPHOT Output

The DOLPHOT/WFC3 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.

Within the photometry blocks, there are also some differences. First is that calibrated instrumental magnitudes and transformed magnitudes are provided. Note that transformations are not yet supported, so the second magnitude is always 99.999 in the current DOLPHOT release. Also, a calibrated count rate and uncertainty is provided (effectively the instrumental magnitude and uncertainty expressed as a count rate). Also note that no CTE corrections are currently available.

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 PSF Utility Reference

The following programs are utilities included with the WFC3 module. Since there is little need for computing custom PSFs, no documentation is provided – use at your own risk!

5.1 *wfc3makepsf*: Compute PSF for Filter

```
wfc3makepsf <filter> <chips>
```

5.2 *wfc3showpsf*: Display PSF for Filter

```
wfc3showpsf <filter> <chip> <X/256> <Y/256> <dX> <dY> <<flags>>
```

5.3 *wfc3ttpar*: Make Tiny Tim Parameter Files and Script

```
wfc3ttpar <filter>
```

6 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 WFC3/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 acsmask and aligned to the reference using acsfitdistort, while the WFC3/IR images would be preprocessed with wfc3mask and aligned using wfc3fitdistort.

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, WFC3/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, especially if WFC3/IR data is being used. 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 WFC3.

7   Recipe for WFC3 Photometry

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

1. Run wfc3mask on all WFC3 images.

2. For WFC3/UVIS image, use splitgroups to split into *.chip1.fits and *.chip2.fits files.

3. Run calsky on each image to generate *.sky.fits files.

4. If not using UseWCS of 1 or 2 in DOLPHOT, run wfc3fitdistort to compute the alignment of each image to the reference.

5. Run dolphot.

6. If artificial star tests are to be run, use wfc3fakelist to generate the star list and re-run dolphot with the FakeStar parameter used to trigger artificial star mode.

8   Changes to DOLPHOT/WFC3

October 16, 2011 release

1. Initial release

February 25, 2012 release

1. Incorporated Jay Anderson's PSF libraries for WFC3/UVIS and WFC3/IR

April 8, 2012 release

1. Incorporated new zero points from STScI.

June 17, 2012 release

1. wfc3mask modified to correctly parse astrodrizzle outputs

January 19, 2013 release

1. Added support for selected IR and UVIS subarrays

February 8, 2013 release

1. Bug fix in wfc3mask

March 6, 2013 release

1. Added support for all WFC3 imaging apertures / subarrays

January 13, 2014 release

1. Updated acsmask to allow DETECTOR and APERTURE flags to be in the data table instead of the header for drizzled images.

2. Minor robustness update to transformations

July 27, 2014 release

1. Added support for new STScI format of FLT/CRJ FITS files

October 31, 2014 release

1. Added support for new STScI format of UVIS FLT/CRJ FITS files

November 9, 2014 release

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

August 15, 2015 release

1. Added support for Jay Anderson's new (spatially varying) IR PSF library

January 2, 2016 release

1. Fixed bug in wfc3mask affecting drizzled images made from large numbers of input images

2. Added support for new 8-extension format

January 10, 2016 release

1. Added CombineChi flag

April 17, 2016 release

1. Incorporated new UVIS zero points and encircled energy corrections

August 26, 2017 release

1. Incorporated UVIS zero points from WFC3 ISR 2017-14

February 3, 2018 release

1. Improved calculation of combined magnitudes and uncertainties