

DOLPHOT/Euclid User's Guide

version 2.1

Andrew Dolphin

adolphin@rtx.com

January 2026

<http://americano.dolphotsim.com/dolphot>

Contents

1	Introduction	2
2	Installation	3
2.1	Makefile	3
2.2	PSFs	3
3	Preprocessing Steps	4
3.1	Create/Select Reference Images	4
3.2	<i>euclidprocmosaic</i> : Convert Mosaic to DOLPHOT Usable Format	4
3.3	<i>euclidmakepam</i> : Create Pixel Area Map	5
3.4	<i>nispmask</i> , <i>vismask</i> : Convert to DOLPHOT Usable Format	5
3.5	<i>calcsky</i> : Calculate Sky for Image	6
4	Running DOLPHOT	7
4.1	DOLPHOT Parameters	7
4.2	DOLPHOT Output	9
5	Recipe for Euclid Photometry	10
6	Changes to DOLPHOT/Euclid	11

1 Introduction

This manual describes installation and use of the Euclid module for DOLPHOT. At present, this module uses DOLPHOT's inherent PSF calculation capabilities, but plans are to implement lookup tables computed using empirical PSFs.

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 Euclid module is done in addition to the regular DOLPHOT installation. See the DOLPHOT manual for installation instructions for DOLPHOT.

The Euclid module requires less than 1Mb for the sources and binaries. 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 Euclid module (as well as the PSFs) should be expanded from the same directory from which the DOLPHOT distribution was expanded.

2.1 Makefile

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

No customization of the Makefile within the “euclid” subdirectory is needed.

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 Euclid module to run correctly, data must be in the default format created by the Euclid processing pipeline. Photometry is executed only on non-Mosaic files (comparable to HST flt/flc or JWST Stage 2 images). Mosaic files (comparable to HST drz/drc or JWST Stage 3 images) can be used as alignment reference images as they produce suboptimal photometry due to resampling of images. Superior photometry can be obtained by providing the stack of individual images used to create the mosaic to DOLPHOT individually.

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.

If not, you would need to select the best existing mosaic image, or create one of your own, for the field you are studying. In the case of NISP, it is important that the mosaic reference image have the plate scale similar to NISP, not the much higher resolution VIS plate scale, which creates very blurry mosaic images if used on NISP data.

Once a reference frame is ready, you may want to co-add the individual exposures taken at the same pointing using the DOLPHOT utility *imcombine*. (This can also improve cosmic ray rejection.) 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 acceptable to photometer all of the Stage 2 images separately.

3.2 *euclidprocmosaic*: Convert Mosaic to DOLPHOT Usable Format

```
euclidprocmask <flags> <fits files>
```

euclidprocmosaic take as input mosaic files from the Euclid pipeline, which are generally undistorted (no TPV or SIP coefficients), background subtracted, and lacking photometric parameters (zero points, exposure times, etc). To be useful for photometry, even as an astrometric reference frame for alignment of individual exposures, the user needs to provide several parameters via command line flags prior to the filename

-scale=# scales the image so that X counts has $\sqrt{X}$ counts of Poisson noise. This would be the integration time multiplied by the gain (about 3.5 for VIS), if the mosaic image is in units of ADU/sec and native resolution of the camera (the default for VIS). **-bgadd=#** adds the background level in electrons **-rdnoise=#** provides the readout noise in electrons (about 3.2 for VIS)

3.3 *euclidmakepam*: Create Pixel Area Map

```
euclidmakepam <fits input> <PAM output>
```

euclidmakepam takes as input FITS files in the format created by the Euclid pipeline, which are 144*3 (432) extension VIS images and 16*3 (48) extension NISP images, the same files input into *nispmask* and *vismask*. The output is the pixel area map for that image, scaled to average to 1.0 across all chips/quadrants of that image. The output is an optional input to *nispmask* and *vismask*. Note this step is only required if you wish to apply a PAM from one FITS for use on another; the mask utilities calculate PAMs by default.

3.4 *nispmask*, *vismask*: Convert to DOLPHOT Usable Format

```
nispmask <flags> <fits files>
```

```
vismask <flags> <fits files>
```

nispmask and *vismask* take as input FITS files in the format created by the Euclid pipeline, which are 144*3 (432) extension VIS images and 16*3 (48) extension NISP images, with each set of three extensions consisting of a science image, an RMS image, and a data quality flag image for each detector / quadrant. Exposure time, readout noise, and gain are read from the FITS headers, WCS and distortion are read and converted to the TAN-SIP convention, and data quality flags are applied to flag bad or saturated pixels.

Both mask utilities estimate the readout noise from the RMS extension. In the case of Vis, this is almost always exactly equal to the value from the FITS header, so it is not used by default. The `-useestnoise` flag in *vismask* will apply the estimated value.

Conversely, in NISP, the estimated value is very different than what is calculated from the FITS header, so it is used by default. The `-calcestnoise` flag in *nispmask* still calculate the estimated value but will not use it for photometry.

Both mask utilities also apply a pixel area map. By default, this is calculated from the distortion coefficients in the file being processed. This behavior can be changed with two flags. The `-PAM=` flag will instead use the specified PAM file (created by the *euclidmakepam* utility. The `-noPAM` flag will disable PAM processing. Note if both this and `-PAM=` are specified, the later flag will take precedence.

Before running *nispmask* or *vismask*, make sure you have backed up the original images, since the tool will alter them!

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

All extensions other than the data extension are deleted when writing the data back to disk. The output image will thus have only 144 or 16 extensions.

3.5 *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 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 non-zero FitSky settings in my photometry, the negative values of *step* are sufficient.

Note that running *calcsky* is currently mandatory, even if using FitSky to measure sky within dolphot.

4 Running DOLPHOT

4.1 DOLPHOT Parameters

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

img_aprad is set to 10 pixels (5 for NISP)

img_RPSF and img_RAper cannot exceed 20 pixels

Zero disabled (set to VEGAMAG or ABMAG zero points)

FPSF, SubPixel, img_psfa, img_psfb, img_psfsc disabled

EPSF, PSFsol, PSFStep disabled

MinS, MaxS, MaxE disabled

Recommended values for other parameters when running with `FitSky = 1` are as follows:

`PSFPhot = 1`

`PSFPhotIt = 2`

`Force1 = 0`

`img_apsky = 20 35`

`img_RAper = 4`

`img_RChi = 1.5` (2 for Vis)

`img_RSky = 15 35`

`img_RPSF = 10`

`SkipSky = 1`

`SkySig = 2.25`

`SecondPass = 0`

`SigFindMult = 0.85`

`MaxIT = 25`

`NoiseMult = 0.10`

`FSat = 0.999`

```
ApCor = 1
RCentroid = 1
PosStep = 0.25
RCombine = 1.5
SigPSF = 5.0
PSFres = 1
UseWCS = 2
```

Running with `FitSky = 2` is recommended in crowded fields. In this case, recommended parameter changes to the above are as follows for NISP:

```
img_RAper = 2
img_RSky2 = 3 10
```

For Vis, with slightly broader PSFs, the recommendations are:

```
img_RAper = 3
img_RSky2 = 4 10
```

If using `FitSky = 3`, the parameters are largely the same except that suggested `img_RAper` values are 6 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 three new parameters that can be used:

```
FlagMask = 4
CombineChi = 0
NISPvega = 1
```

`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 χ 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 `NISPvega` is set to 0, calibrated fluxes will be provided in units of Jy, and instrumental magnitudes in units of ABmag. Otherwise, when set to 1, Vega magnitudes are used and calibrated fluxes are scaled to for a zeroth magnitude source.

Note that the DOLPHOT/Euclid module does not have default distortion parameters included, but rather relies on the WCS information in the FITS file obtained from the JWST pipeline. Thus, `UseWCS = 2` is required when reducing Euclid data.

4.2 DOLPHOT Output

The DOLPHOT/Euclid 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. Following any combined-filter photometry are the single-image photometry blocks, in order that the images are provided in the parameter files.

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).

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 Recipe for Euclid Photometry

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

1. Run `euclidprocmosaic` on the Euclid reference mosaic image
2. Run `nispmask` / `vismask` on all Euclid exposures, followed by `splitgropus`
3. Run `calsky` on each chip image to generate `*.sky.fits` files
4. Run `dolphot`

6 Changes to DOLPHOT/Euclid

January 2, 2026 release

1. Initial release
2. Caveat: Empirical/model PSFs are not currently available, so PSFs have to be estimated by DOLPHOT
3. Caveat: Encircled energy corrections are not currently incorporated
4. Caveat: Pixel area maps are currently not used by default
5. Caveat: RMS extension data are currently not used by default

April 18, 2026 release

1. Incorporated Empirical NISP PSFs and encircled energies
2. Incorporated Pixel area maps
3. Incorporated use of RMS extension data in NISP by default
4. Caveat: NISP PSFs and encircled energies are preliminary and do not include spatial variation
5. Caveat: Vis Empirical/model PSFs are not currently available, so PSFs have to be estimated by DOLPHOT
6. Caveat: Vis Encircled energy corrections are not currently incorporated