Photometry Pipeline Documentation

Release 1.0

Michael Mommert

Jul 20, 2022

Contents

CHAPTER 1

Introduction

The Photometry Pipeline (PP) is a Python software package for automated photometric analysis of imaging data from small to medium-sized observatories. It uses Source Extractor and SCAMP to register and photometrically calibrate images based on catalogs that are available online; photometry is measured using Source Extractor aperture photometry. PP has been designed for asteroid observations, but can be used with any kind of imaging data.

CHAPTER 2

Scope and Applicability

PP has been designed to provide automated photometry for the majority of data coming from small to medium-sized observatories. It is not intended to provide high-accuracy photometry, nor is it designed to work on extremely sparse or crowded fields. PP requires a field of view of a few arcminutes to ensure that sufficient background stars are available for registration and photometric calibration. For a field several arcminutes across, calibrated photometric uncertaintes are usually better than 0.05 mag, if sufficient non-saturated background stars with cataloged brightness are available. Feel free to try PP on your data, but please be aware that it has its limitations.

The following features are currently available as part of PP:

- automated aperture photometry based on a curve-of-growth analysis
- photometric calibration in ugriz, BVRI, JHK based on catalog coverage
- full support of moving target photometry
- target identification based on tabulated positions for fixed targets and moving targets, target identifier for moving targets
- support of Gaia (DR1) astrometry for image registration
- Python 2 and 3 compatibility (thanks to boada)

CHAPTER 3

Status

The development of PP has stalled; please be aware that tailored support can only be provided in exceptional cases.

CHAPTER 4

Contents

4.1 Installation and Setup

4.1.1 Installation

General Installation Instructions

PP is available from github. You can get the source code by typing into your terminal:

```
git clone https://github.com/mommermi/photometrypipeline
```

This will create a `photometrypipeline/` directory in your current directory.

Software Requirements

PP only runs on Python 3. It furthermore requires requires git, a number of non-standard Python modules (available from the Python Package Index through pip):

- numpy
- scipy
- astropy
- astroquery (version >= 0.3.9)
- matplotlib
- future
- skimage
- pandas

and some freely available software:

- imagemagick
- Source Extractor
- SCAMP (please download the latest development version)

Setup

In order to be able to use PP anywhere on your machine, you have to add the full path of the `photometrypipeline/` directory to your *PYTHONPATH* and *PATH* variables, and you have to create a *PHOTPIPEDIR* variable on your system that points to the same directory (include these commands in your `.bashrc`, `.cshrc`, or `.profile` file.)

Installation Instructions for Ubuntu 16.04+

Clone the PP github repo:

```
git clone https://github.com/mommermi/photometrypipeline
```

Install software requirements for SCAMP, Source Extractor and imagemagick:

```
sudo apt-get install -y \
       build-essential \
       libssl-dev \
       libffi-dev \
       git \
       sextractor \
       wget \
       imagemagick \
       curl \
       libplplot-dev \
       libshp-dev \
       libcurl4-gnutls-dev \
       liblapack3 liblapack-dev liblapacke liblapacke-dev \
       libfftw3-3 libfftw3-dev libfftw3-single3 \
       libatlas-base-dev \
       scamp
```

Install Python modules:

```
pip install --upgrade --user numpy scipy astropy astroquery matplotlib pandas future␣
↪scikit-image
```

Add these lines to the `.bashrc` file in your home directory and replace `<path>` with the actual path to the PP directory:

```
# photometry pipeline setup
export PHOTPIPEDIR=<path>/photometrypipeline
export PATH=$PATH:~<path>/photometrypipeline/
```

Kudos to towicode for figuring out the SCAMP requirements.

Installation Instructions for Mac OS Catalina

Install Anaconda: download Anaconda from https://www.continuum.io/downloads In your terminal window type one of the below and follow the instructions:

```
https://repo.anaconda.com/archive/Anaconda3-2020.07-MacOSX-x86_64.sh
```

Install AstroConda:

```
$ conda update conda
$ conda config --add channels http://ssb.stsci.edu/astroconda
```

Add packages to base environment:: $ conda install stsci wcstools

Install extra packages:: $ conda install -c astropy pyephem astroquery astroplan astroscrappy ccdproc $ conda install -c conda-forge uncertainties lmfit pysftp scikit-image cfitsio ghostscript openorb $ conda install -c anaconda future pillow wget gfortran_osx-64 $ conda install -c bioconda graphicsmagick

Install sextractor and scamp:: $ conda install -c conda-forge astromatic-scamp $ conda install -c conda-forge astromatic-source-extractor

Update pip and install dev version of astroquery:: $ pip install –upgrade pip $ pip install –upgrade –pre astroquery

Install homebrew:: $ ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)" $ brew doctor

Install extra software:: $ brew install imagemagick

(Thanks to Nick Moskovitz for these instructions!)

Legacy: Installation Instructions for Mac OS (Sierra)

Install Anaconda: download Anaconda from https://www.continuum.io/downloads In your terminal window type one of the below and follow the instructions:

```
Anaconda3-4.4.0-MacOSX-x86_64.sh
```

Install Homebrew:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/
↪master/install)"
```

Install MacPorts by downloading the installer from their website: https://www.macports.org:

```
sudo port -v selfupdate
```

Test if gcc will run by typing `gcc` in a bash terminal. If prompted to install command-line tools, follow the instructions to do so

Update pip:

```
python -m pip install --upgrade pip
```

Install extra python modules in Anaconda:

```
pip install --upgrade --user numpy scipy astropy astroquery matplotlib
pip install astropy pandas
conda update astropy
```

Install SExtractor:

```
brew install brewsci/science/sextractor
```

Install SCAMP:

```
brew install brewsci/science/scamp
```

Install extra software:

```
sudo port install wget
sudo port install imagemagick
```

Install PP:

```
git clone https://github.com/mommermi/photometrypipeline
```

Add to `\/.bash_profile` file by replacing `<username>` with your system user name and `<PyVersion>` with the Python version you are using:

```
export PATH="$PATH:/home/<username>/.local/bin"
export PATH="$PATH:/Users/<username>/photometrypipeline"
export PATH="$PATH:/Users/<username>/Library/Python/<PyVersion>/bin"
```

Kudos to Annika Gustafsson and Colin Chandler for producing this summary and Kathryn Neugent for providing corrections.

4.1.2 Update your Version of PP

In order to update your version of PP, simply change into `photometrypipeline/` and type:

```
git pull
```

You should do this regularly as PP is still under constant development.

4.1.3 Example Data

The PP github clone comes with some sample data that can be used to test if the pipeline works properly. The data were taken with the VATT4k camera on the VATT and can be found in `example_data/vatt4k`. In order to run the pipeline on these images, copy them to a new directory, change there, and run `pp_run mscience*fits`. If everything works out properly, the results (`photometry_3552.dat`) should resemble those in `example_data/vatt4k/LOG`.

Telescope Setup

PP critically relies on information provided in the FITS image headers to handle data properly. While the FITS format is standardized, header keywords are not, leading to additional complications in the interpretation of FITS files. In order to be able to work with a multitude of different telescopes and instruments, PP comes with guidelines of how to read FITS files coming from different telescopes/instruments. These guidelines are imprinted in the `setup/telescopes.py` file. In order to prevent compatibility issues, you should not change this file directly. Instead, please create and use a `setup/mytelescopes.py` as described below. You can implement as many telescopes as you want in this file. The advantage is that the file will not be changed as a result of git pull requests.

The '*telescope file*' includes for each telescope/instrument combination a dictionary (`*_param`) that translates general descriptions for FITS header keywords into specific keywords used by the respective telescope/instrument combination. For example, the telescope pointing RA keyword might be named `RA` for one telescope, but `TELRA` for another – PP will refer to either of those as `ra`. The *telescope file* catches these degeneracies and allows the pipeline to understand images coming from a variety of telescopes. The meanings of the individual keys in this dictionary are explained in the comments of the respective key. Furthermore, each telescope/instrument combination must have parameter files

for Source Extractor and SCAMP (SWARP is currently not supported). Mask files are used by Source Extractor to mask certain regions of the image detector – mask files are only required if field vignetting or image artifacts (e.g., high noise levels in certain areas of the detector) strongly affect the detection of sources in the field.

If you want to include you own telescope into the *telescope file*, follow these steps:

1. Download the mytelescopes.py file into your `setup/` directory and duplicate the `mytelescope_param` dictionary. Change the `MYTELESCOPE` identifier of the duplicate and give it a unique name (e.g., `42INCH_CCD`).
2. Look at the image header of one of your science images and identify the different fields of the `*_param` file. Replace the dictionary item values accordingly.
3. In the `setup/` directory, copy the Source Extractor (`.sex`) and SCAMP (`.scamp`) parameter files from either telescope and name them after your telescope (e.g., `42inch_ccd.scamp`).
4. Add your telescope's identifier to the `implemented_telescopes` list in `setup/mytelescopes.py`, as well as the `telescope_parameters` dictionary. Finally, add your telescope's identifier to the `instrument_identifiers` dictionary: the value is your telescope's identifier, the key is the `INSTRUME` header keyword (this is present in most FITS data).
5. Run *`pp_prepare()`* over one of your images. Check with *ds9* or some other tool if the image orientation provided by *`pp_prepare()`* is correct. If not, play with the *flipx*, *flipy* parameters in your *telescope file*.

4.2 Quickstart

4.2.1 Prerequisits

Image data should be properly reduced before using the pipeline for best results, including cropping the data section. Bias subtraction and flat fielding improves photometry results but is not absolutely necessary. PP's ability to provide astrometric and photometric calibration puts some constraints on the way data is stored: data from separate fields, as well as data using different instrument settings (e.g., different binning modes) should be stored in individual directories, which in turn should be separated by filters:

```
-+- all_data -+
              |
              +- field_1 -+- filter_1
              |           +- filter_2
              |           +- filter_3
              |
              +- field_2 -+- filter_1
              |           +- filter_2
             ...
```

Separate fields are defined as having gaps between individual frames that are comparable to, or larger than, the field of view. Series of frames that were tracked on a moving target can be put in the same directory if the total track is smaller than 3-5 times the size of a single-frame field of view.

Moving objects are currently only identified based on the images' `OBJECT` keywords. The object name should be as simple as possible, consisting either of the bodies official number or designation; please use either a blank or an underscore to separate the designation's year from the identifier.

4.2.2 Running PP

PP can be run in a fully automated or semi-automated mode, providing different levels of user interaction.

Fully Automated Mode

In the directory tree example above, PP can be run in different places, treating the data differently. If you want to run PP only on data for one field and filter, you can change in that directory and **run PP locally on all fits files in that directory**:

```
cd all_data/field_1/filter_1
pp_run *fits
```

If your data are organized as shown in the example above, you can **run PP from any higher level directory to analyze all underlying directories in a consecutive way**:

```
cd all_data
pp_run all
```

Passing `all` signalizes PP to walk through underlying directories, starting from the current one. What happens is that PP creates a PP subprocess for each data directory. In case you want PP to only run on a subset of fits files starting with a certain prefix, you can use option *-prefix*, e.g.,

```
pp_run -prefix reduced all
```

is the equivalent of using only files that are included in `reduced*.fits`.

Semi-Automated Mode Walkthrough

This section describes the individual steps PP takes to analyze the data. While `pp_run` performs these steps automatically, each of the following functions can be called manually, which allows to tweak the analysis process. If you intend to perform the analysis fully manually, please note the individual functions have to be called in the following order:

- *pp_prepare()*: prepare the input images and implant rough WCS information into the image header
- *pp_register()*: use `SCAMP` to register all input images based on the implanted rough WCS information; different catalogs are tried until all images have been registered ; this function calls *pp_extract()* automatically
- *pp_photometry()*: derive instrumental magnitudes using a curve-of-growth analysis, or a manually provided aperture radius
- *pp_calibrate()*: photometrically calibrate instrumental magnitudes and create a *SQLite* database file for each image; note that this function has to be called even if you plan on using instrumental magnitudes only (use the *-instrumental* option)
- *pp_distill()*: extract target information from the photometry databases created by the previous task; see the function reference for the different options of target identification

Manual Target Identification

In case the target has no identifier, or positions/ephemerides cannot be obtained automatically (e.g., space debris, newly discovered asteroids, etc.), or you want to verify the calibration accuracy using a manually selected control star in the field, the target has to be identified manually from the image data using *pp_manident()*.

Image data are at minimum required to have passed *pp_prepare()*, *pp_photometry()*, and *pp_calibrate()*; *pp_manident()* may also be called after a full *pp_run()* call. In order to identify the target in all images, *pp_manident()* allows you to browse through all images and click on the target. The trajectory of the target is fit using a spline function. Quitting *pp_manident()* creates a `positions.dat` file, which can be used as input for *pp_distill()* using the *-positions* option.

The manual target identification also allows the user to extract photometry from images with highly trailed background stars. In that case, the resulting photometry will consist of instrumental magnitudes. Hence, `pp_register()` does not have to be called and `pp_calibrate()` should be called using the *-instrumental* option. Positions used in the target identification and listed in the final photometry file are based on the rough WCS information implanted by `pp_prepare()` and should not be trusted!

4.2.3 PP Diagnostics

PP generates by default significant amounts of diagnostic information on each run. These information can be accessed in the individual directories where the data resides with any web browser, e.g.,

```
firefox all_data/field_2/filter_3/diagnostics.html
```

If you ran PP with the *all* argument (see above), a file `summary.html` will be generated in the root directory (`all_data`), which provides links to the individual `index.html` files.

More information on the diagnostic output is available here: *Diagnostics*.

4.2.4 Results

PP derives the calibrated photometry for the target that it finds in the `OBJECT` header keyword, as well as one rather bright 'control star' that is used to check the consistency of the photometric calibration. Results are written to files `photometry_<objectname>.dat` in the respective filter directory.

Although PP is designed to run mostly automatically, some common sense is required to make sure the results are reliable.

4.3 Supported Observatories

The pipeline is currently set up to work on data from the following observatories/instruments:

Telescope	Instrument	PP Keyword
Apache Point ARC 3.5m	AGILE	ARC35AGILE
Apache Point ARC 3.5m	ARCTIC	ARC35ARCTIC
Apache Point ARC 3.5m	SPICAM	ARC35SPICAM
Calar Alto 1.23m	DLR-MkIII	CA123DLRMKIII
CFHT	Megaprime	CFHTMEGAPRIME
CTIO 0.9m	CFCCD	CTIO09
CTIO 1.0m	Y4KCam	CTIO10
CTIO 1.3m	ANDICAM (CCD)	CTIO13CCD
Danish 1.54m	DFOSC (imaging)	DFOSC
Discovery Channel Telescope	Large Monolithic Imager	DCTLMI
Gemini North 8.2m	GMOS	GMOSN
Gran Canaria 1.5m	MUSCAT2	TCS15MUSCAT2
GTC	OSIRIS	GTCOSIRIS
INT	WFC	INTWFC
IRSF 1.4m	SIRIUS	IRSFSIRIUS
KMTnet SAAO	— (*)	KMTNETS
KPNO 4m	Mosaic 1	KPNO4MOS1
KPNO 4m	Mosaic 3	KPNO4MOS3

Continued on next page

Table 1 – continued from previous page

Telescope	Instrument	PP Keyword
KPNO 4m	NEWFIRM	KPNO4NEWF
LCOGT 1m (LSC)	SBIG (kb78)	LCOSBIGKB78
LCOGT 1m (LSC)	Sinistro (fl03)	LCOSINFL03
LCOGT 1m (CPT)	Sinistro (fl06)	LCOSINFL06
LCOGT 1m (LSC)	Sinistro (fa03)	LCOSINFA03
LCOGT XXXm (XXX)	Sinistro (fa15)	LCOSINFA15
LCOGT 1m (COJ)	Sinistro (fl11)	LCOSINFL11
LCOGT 1m (CPT)	Sinistro (fl16)	LCOSINFL16
LCOGT 2m (COJ)	Spectral (fs01)	LCOSPECFS01
Lowell 31”	NASACAM	LOWELL31
Lowell 42”	NASA42	LOWELL42
Lowell 42”	SITE	LOWELL42SITE
Lowell 24”	LONEOS	LONEOS
Magellan	IMACS long camera	MAGIMACSL
Magellan	IMACS short camera	MAGIMACSS
McDonald Struve 2.1m	CQUEAN	STRUVECQUEAN
MMT	MMTCAM	MMTCAM
NEXT 0.6m	FLI	NEXT
Nordic Optical Telescope	ALFOSC	NOTALFOSC
Observatoire Haute- Provence 1.2m	CCD	OHP120
Palomar 60-inch	optical facility camera	P60OPT
Palomar 60-inch	SED Machine	P60SEDM
San Pedro Martir 84cm	Mexman (E2V CCD)	MEXMAN
Steward Observatory 90” Bok Telescope	Spacewatch Camera	STEWARD90SCC
SOAR 4.1m	Goodman (**)	SOARGOODMAN
Sutherland 40-inch	SHA	SL40IN
Telescopio Nazionale Galileo	DOLORES	TNGDOLORES
UKIRT	WFCAM	UKIRTWFCAM
Vatican Advanced Technology Telescope	VATT4k	VATT4K
Very Large Telescope	FORS2	VLTFORS2
WIYN 0.9m	Half Degree Imager	WIYN09HDI
ZTF	Mosaic (***)	ZTFMOSAIC
Generic Telescope	any	GENERIC

(*): This camera is a multi-detector instrument; it is recommended to split multi-extension FITS frames from this instrument into individual single-extension FITS images and to run the pipeline on these individual FITS images.

(**): SOAR Goodman image header have a number of non-standard keywords that force astropy to crash; in order to use PP, please remove header keywords `PARAM0`, `PARAM61`, `PARAM62`, and `PARAM63` prior to running PP, e.g., using the `delhead` command provided by WCSTools.

(***): wcs provided with telescope data is usually excellent; in this case I suggest skipping registration and to use `pp_run` with the `-keep_wcs` option.

It is recommended to stitch images from cameras with multiple amplifiers together using the correct orientations. Furthermore, it is recommended to flatten FITS files with multiple headers by simply combining the individual headers under a single extension.

4.4 Supported Reference Catalogs

PP is currently able to use the following catalogs for astrometric and photometric calibration:

Catalog Name	Catalog Type	Registration?	Filter Bands	Comments
Gaia DR1 (Gaia)	astrometric	yes	G	all-sky catalog, excellent astrometry, G < 20 mag
Gaia/TGAS (TGAS)	astrometric	yes	G	all-sky catalog, excellent astrometry, G < 12 mag
USNO-B1.0	astrometric	yes	R2	good astrometry, V < 21 mag
2MASS (2MASS)	astrometric/photometric	yes	J, H, Ks, K* (Vega)	all-sky NIR catalog, good astrometry
URAT-1 (URAT-1)	astrometric/photometric	yes (SCAMP trunk.r345)	g, r, i (SDSS AB); B, V, R*, I* (Vega)	good coverage over the Northern hemisphere, photometry from APASS (see below)
Sloan Digital Sky Survey Release 9 (SDSS-R9)	astrometric/photometric	yes	u, g, r, i, z (SDSS AB); U*, B*, V*, R*, I* (Vega)	excellent photometry, Northern hemisphere, patchy coverage
AAVSO Photometric All Sky Survey Release 9 (APASS9)	photometric	no	g, r, i (SDSS AB); B, V, R*, I* (Vega)	good coverage, good photometry for stars with V<17
Pan-STARRS DR1 (PANSTARRS)	photometric	no	g, r, i, z, y (SDSS AB); B*, V*, R*, I* (Vega)	good coverage, good photometry for stars with V<20; currently, only cone searches with radius < 0.5 deg supported
SkyMapper DR1 (SkyMapper)	photometric	no	u, v, g, r, i, z (SDSS AB)	good coverage in the Southern hemisphere

The catalog name in brackets is the identifier used by PP; e.g., if you want to use URAT-1 for the registration of your images, use option *-cat URAT-1* with the `pp_register` command. The "Registration?" column identifies if this catalog is supported by SCAMP in order to use it for image registration. All filter bands marked with an asterisk (*) are obtained through transformations from other bands; the respective photometric system is shown in brackets.

4.4.1 Supported Catalog Transformations

PP supports the following catalog transformations:

- ugriz -> BVRI: Chonis & Gaskell 2008
- JHKs (2MASS) -> JHK (UKIRT): Hodgkin et al. 2009
- Pan-STARRS grizy -> SDSS griz + BVRI: Tonry et al. 2012

Independent checks indicate that these transformations are reliable and accurate. More quantitative results coming soon...

4.5 Functions

The individual pipeline functions are introduced and explained below. All functions presented here can be called from the terminal.

pp_run ([*-prefix string*][, *-target string*][, *-filter string*][, *-fixed_aprad float*][, *-solar*][, *-rerun_registration*][, *-asteroids*][, *-keep_wcs*], *images*)
serves as a wrapper for all the individual pipeline processes

Parameters

- **-prefix** – (optional) the prefix of all science data images if *pp_run* is called using `images = all`
- **-target** – (optional) the target name to be used in all images, this overwrites the `OBJECT` keyword in the FITS headers; note to replace blanks with underscores if the target's name is a designation
- **-filter** – (optional) manual filter name override for the photometric calibration step
- **-fixed_aprad** – (optional) use this fixed aperture radius for all data instead of finding the aperture radius in a curve-of-growth analysis
- **-source_tolerance** – (optional) this parameter defines the level at which sources are rejected in the image registration process; see *pp_register()* for details.
- **-solar** – the photometric calibration (*pp_calibrate*) is only using stars with solar-like colors (see *pp_calibrate* documentation for details); if the calibration using *-solar* fails (too few matches with reference catalog), the calibration is automatically repeated without the *-solar* option
- **-rerun_registration** – (optional) for some data, the image registration has to be run twice; this option will rerun the registration step, if not all frames registered successfull during the first try.
- **-asteroids** – (optional) make use of `-asteroids` option of *pp_distill()*
- **-keep_wcs** – (optional) skip registration and keep wcs information
- **images** – images on which the pipeline is supposed to run, wildcard symbols (`'*'`, `'?'`) can be used; or, by using `all`, PP runs on all FITS files in underlying directories (the range of images can be limited by using the *-prefix* option)

The use of *pp_run* is discussed in the *Quickstart* reference.

This wrapper should work successfully for most data sets. If the analysis fails at some point - or provides inadequate results - every single step in the pipeline can be run manually, providing the possibility to manually tweak the process parameters.

The following functions describe the individual pipeline processes in the logical order:

pp_prepare ([*-ra degrees*] [, *-dec degrees*] [, *-flipx*] [, *-flipy*] [, *-rotate degrees*] [, *-target string*] [, *-keep_wcs*], *images*)

prepares image files for use in the pipeline

Parameters

- **-ra** – (optional) manually sets the frame center RA
- **-dec** – (optional) manually sets the frame center declination
- **-flipx** – (optional) forces the image's x-axis to be flipped in wcs coordinates relative to the respective *Telescope Setup* setting
- **-flipy** – (optional) forces the image's y-axis to be flipped in wcs coordinates relative to the respective *Telescope Setup* setting
- **-rotate** – (optional) rotates the image's orientation in the sky (East-of-North) relative to the respective *Telescope Setup* setting
- **-target** – (optional) the target name to be used in all images, this overwrites the `OBJECT` keyword in the FITS headers; note to replace blanks with underscores if the target's name is a designation

- **keep_wcs** – retain original wcs header information and use that as initial seed for the image registration process
- **images** – images on which *pp_prepare* is supposed to run

This function prepares the image data by creating necessary FITS header keywords (e.g., the observation mid-time `MIDTIMJD`, the pixel scale `SECPIX`, ...), and by including fake wcs information that is required by SCAMP. Existing wcs information are deleted, as they might cause confusion with those information generated by *SCAMP*.

The diagnostic output of this function is a list of all frames with information on observation midtime, target name, filter, airmass, integration time, and field of view. A thumbnail image of each input image is available from this table.

pp_extract ([*-snr float*][, *-minarea integer*][, *-paramfile path*][, *-aprad float*][, *-telescope string*][, *-ignore_saturation*][, *-nodeblending*][, *-quiet*], *images*)
wrapper for Source Extractor

Parameters

- **-snr** – (optional) minimum SNR of sources to be extracted, default: 1.5
- **-minarea** – (optional) minimum number of connected pixels above the SNR threshold for a valid source, default: 3
- **-paramfile** – (optional) manual override for the *Source Extractor* parameter file
- **-aprad** – (optional) aperture photometry aperture radius in pixels; if no aperture radius is given here, the default aperture radius for this telescope/instrument combination is used (see *Telescope Setup* reference)
- **-telescope** – (optional) manual override for the telescope identifier (see *Supported Observatories*)
- **-ignore_saturation** – (optional) using this option will not flag saturated sources; as a result, they are not rejected in the registration and calibration process
- **-nodeblending** – (optional) deactives Source Extractor deblending
- **-quiet** – (optional) suppress output on the screen
- **images** – images to run *pp_extract* on

pp_extract is automatically called by `pp_register()` and `pp_photometry()`. Usually, there is no reason to call this function manually.

pp_register ([*-snr float*][, *-minarea integer*][, *-cat catalogname*][, *-source_tolerance string*][, *-nodeblending*], *images*)
astrometric calibration of the input images using SCAMP

Parameters

- **-snr** – (optional) minimum SNR of sources to be extracted for the registration, default: 3
- **-minarea** – (optional) minimum number of connected pixels above the SNR threshold for a valid source, default: *Telescope Setup* setting
- **-cat** – (optional) reference catalog override for astrometric calibration (a list of supported catalogs is listed here: *Supported Reference Catalogs*); if not specific catalog is requested, those listed in the *Telescope Setup* reference are tried
- **-source_tolerance** – (optional) this parameter defines the cumulative level at which sources are rejected in the image registration process (in the following sequence, each level includes the previous rejection scheme): *none*: only flawless sources are used in the registration; *low*: sources with bright neighbors are considered; *medium*: blended sources are

considered; *high*: saturated sources are considered; the default is *high*; see the Source Extractor manual section on internal flags for details.

- **-nodeblending** – (optional) deactives Source Extractor deblending in the detection of sources in the image frames
- **images** – images to run *pp_register* on

pp_register automatically calls `pp_extract()` to identify all sources in the field of view of each image; the source catalogs are stored as `.ldac` files. The *-snr* and *-minarea* options are passed on to `pp_extract()`/*Source Extractor* in order to specify the source properties. *pp_register* utilizes *SCAMP* to match the source catalogs with astrometric catalogs as specified for this telescope/instrument combination (see *Telescope Setup* reference), or as provided by the user with the *-cat* option. Catalogs are accessed through the CDS Vizier server; the downloaded catalog is written as a `.cat` file into the working directory for later inspection. Among others, *SCAMP* outputs two diagnostic numbers: `AS_CONTRAST` and `XY_CONTRAST`. The image registration generally has succeeded if both numbers are greater than 2.5 - the higher the contrast numbers, the better the fit. Unless every image has been registered properly, each catalog is matched twice using information from the last *SCAMP* run. The routine ends if all images have been registered properly or all catalogs have been used twice. Before starting the registration process, this function will check the distribution of sources in all images in the plane of the sky. If there appear to be two or more fields that are non-overlapping and separted by at least 5 degrees, those fields with fewer sources will be rejected and considered not registered. The *-nodeblending* option will deactivate deblending that is usually performed by Source Extractor. The advantage of deactivating deblending is that bright and saturated sources are recognized as single objects instead of a group of fainter sources; this is especially useful in the registration process.

The diagnostic output of this function is a table of the *SCAMP* output parameters and a presentation of each image overplotted with the catalog sources used in the matching.

pp_photometry ([*-snr float*][, *-minarea float*][, *-aprad float*][, *-target string*][, *-background_only*][, *-target_only*], *images*)

curve-of-growth analysis of the input images and source extraction using a derived optimum aperture radius resulting in final instrumental magnitudes

Parameters

- **-snr** – (optional) minimum SNR of sources to be accounted for in the analysis, default: 2
- **-minarea** – (optional) minimum number of connected pixels above the SNR threshold for a valid source, default: *Telescope Setup* setting
- **-aprad** – (optional) if this option is used, the curve-of-growth analysis is skipped and instrumental magnitudes are derived with this aperture radius
- **-target** – the target name to be used in all images, this overrides the `OBJECT` keyword in the FITS headers; note to replace blanks with underscores if the target's name is a designation
- **-background_only** – only account for background sources in the curve-of-growth analysis
- **-target_only** – only account for the target in the curve-of-growth analysis
- **-nodeblending** – (optional) deactives Source Extractor deblending
- **images** – images to run *pp_photometry* on

pp_photometry calls `pp_extract()` with a list of 20 different aperture radii in order to establish a separate curve-of-growth for the target (if it is a moving target) and the average of all fixed sources in the images. The motivation behind this split is to identify and minimize the impact of potential trailing caused by the relative motion of the target. The optimum aperture radius is derived based on different strategies: (1) the default is to pick the smallest aperture radius at which both the target and the background fractional fluxes are greater than 70% and the difference between the target and background curves is smaller than 5% (minimizing systematic

offsets in the measured fluxes); (2) the smallest aperture radius at which the average fractional background flux is greater than 70% if the option *-background_only* is used; (3) the smallest aperture radius at which the target flux is greater than 70% if the option *-target_only* is used. These strategies have been derived empirically and lead to reliable flux measurements in most cases. The *-target <targetname>* option allows for overriding the target name in image header's `OBJECT` keyword. If the function is called with the option *-aprad <aperture radius>*, no curve-of-growth analysis is performed and the provided aperture radius is adopted as the optimum aperture radius. Finally, this function runs *pp_extract()* again over all input images using the derived optimum aperture radius resulting in a new *.ldac* file for each input image providing instrumental magnitudes for all sources in the field.

The diagnostic output of this function are two plots. The first plots shows the fractional combined flux and the fraction SNR of the target and the background sources as a function of aperture radius. The optimum aperture radius is indicated with a vertical line. The second plots shows the median PSF FWHM per frame as a function of time as derived by *Source Extractor*. The optimum aperture diameter is indicated by a horizontal line - this line should always be slightly higher than the measured FWHMs.

pp_calibrate ([*-minstars int/float*] [, *-catalog string*] [, *-filter string*] [, *-maxflag integer*] [, *-instrumental*] [, *-solar*] [, *-use_all_stars*], *images*)

photometric calibration of each input frame in one specific filter

Parameters

- **-minstars** – (optional) minimum number of reference stars used in the photometric calibration; if `int`, use at least this number of stars; if `float` use at least this fraction of the available reference stars; if this option is not used, the default is 0.5 (i.e., use at least 50% of all available reference stars)
- **-catalog** – (optional) manual override for the reference catalog; a list of available reference catalogs is available here: *Supported Reference Catalogs*) or using this routine's help function; if this option is not used, the photometric reference catalogs list in the *Telescope Setup* are used
- **-filter** – (optional) manual override for the filter used in the observations; if this option is not used, the filter name is read from the image FITS headers
- **-maxflag** – (optional) the maximum flag value for sources to be still considered in the calibration process and written into the resulting photometry database; flag values as tabulated in the Source Extractor manual; default value is 3, allowing for sources to have bright neighbors and to be blended with another source; value of 7 permits sources to be (partially) saturated
- **-instrumental** – (optional) if this option is used, the calibration process is skipped entirely and instrumental magnitudes are written to the photometry database for each image
- **-solar** – only use stars with solar-like colors; use this feature for photometry of Solar System bodies. Solar-like stars are selected based on their *g-i* and *r-i* colors, hence, this feature is currently only available for photometric calibration using the PANSTARRS, APASS, and SDSS catalogs. The threshold of solar-like colors is defined by the *_pp_conf.solcol* parameter; the default is the actual color index +- 0.2 mag.
- **use_all_stars** – if used, no quality checks are performed on calibration stars and all stars are used in the calibration.
- **images** – images to run *pp_calibrate* on

Instrumental magnitudes provided by *pp_photometry()* are matched with photometric catalogs in order to derive the magnitude zeropoint of each input image. Photometric catalogs are accessed through CDS Vizier, as specified in the respective *Telescope Setup* setting, or as specified by the *-catalog* option. If *-catalog* is not used, a number of catalogs are tried; if it is used, only one catalog is tried. If no sources are available from either catalog, the function finishes using instrumental magnitudes. Filter transformations are implemented as

documented in *Supported Catalog Transformations*. The calibration process requires a minimum number of matched sources in the field (currently 3) and uses an iterative Chi2 fitting process as documented in Mommert (2016).

This function results in a SQLite database file (*.db*) for each image file, holding calibrated and instrumental magnitudes for all sources found in the field of view.

The diagnostic output of this function consists of a plot of the magnitude zeropoint of all input images as a function of time, as well as a table of all input images, their zeropoints, and the number available catalog sources and the number of sources used in the calibration. Furthermore, detailed information is available on each input image: all catalog sources used in the calibration are listed with their properties, a thumbnail of the image is shown with the calibration sources overplotted, and a diagnostic plot is generated. This plot shows the magnitude zeropoint and its uncertainty as a function of the number of calibration sources used; also, it shows the magnitude residuals as a function of source brightness.

pp_distill ([*-target string*][, *-offset float float*][, *-positions string*][, *-fixedtargets string*][, *-variable_stars*][, *-asteroids*][, *-reject string*], *images*)
extraction of calibrated photometry for targets

Parameters

- **-target** – (optional) the target name to be used in all images, this overrides the `OBJECT` keyword in the FITS headers; note to replace blanks with underscores if the target's name is a designation
- **-offset** – (optional) position offset to apply on target positions (e.g., Horizons position for moving targets) in arcsec; requires two floats, one for RA and one for Dec
- **-positions** – (optional) file that lists the position of the target as a function of time for all frames; exact format: image filename, ra (deg), dec (deg), observations midtime (JD); if this option is used, the header `OBJECT` keyword will not be used to identify the target
- **-fixedtargets** – (optional) file that list targets with fixed positions; exact format: target name, ra (deg), dec (deg); if this option is used, the header `OBJECT` keyword will not be used to identify the target
- **-variable_stars** – (optional) match source catalog with the VSX catalog to identify and extract variable stars
- **-asteroids** – (optional) find serendipitously observed asteroids in the image field using IMCCE's SkyBoT service; extract objects that are bright enough and have accurate orbits
- **-reject** – (optional) this option enables the filtering of data based on predefined criteria before they enter the final photometry file; a single rejection schema identifier or a comma-separated list of identifiers (no whitespaces) can be provided. Rejected observations will still show up in the final photomoetry file, but will be commented out using `#`; no diagnostic output will be generated for rejected observations. Valid identifiers are: `pos` (rejects observations with positional residuals greater than 10 arcsec). Default: `pos`
- **images** – images to run *pp_distill* on

This function will automatically read the target name from the FITS images (or use the manually provided one), pull target positions from JPL Horizons, and extract calibrated photometry from the database catalogs created with *pp_calibrate()* in to a `photometry_<targetname>.dat` file. In addition to the primary target, this function also creates a photometry output file for one relatively bright star that is present in the first and the last image of the series - this star serves as a control star to check the consistency of the derive magnitude zeropoints. If either the *-positions* or *-fixedtargets* option is used, JPL Horizons will not be queried, same if *-asteroids* is used. The latter will query the target field using IMCCE's SkyBoT service and extract asteroids from the source catalog that have positional uncertainties less than 5 pixels (un-binned) and are brighter than 90% of the sources in the field. Note that both the options *-variable_stars* and *-asteroids* will extract the source that matches the provided target position best - confusion with an unrelated source is possible.

Functions that provide additional functionality:

pp_manident ([*-zoom float*], *images*)
manual target identification

Parameters

- **zoom** – zoom factor applied to images when loaded; number greater than one will increase the size and vice versa; default zoom value is 0.5
- **images** – images to run *pp_manident* on

This function allows to manually identify a target in the images provided and creates a file with the target's position in each image; the resulting file can be used by `pp_distill()` to extract target photometry. Loading all images might take a while, the loading progress is displayed. Once all images have been loaded, the first image is displayed in a window with green circles, indicating sources identified by `pp_extract()`. You can browse between the images with the *a* and *d* keys, or display the next frame with a right-click. Left-click on the target in at least two different images (the target circle color will turn to red) will make this function interpolate the target trajectory using second-order splines or third-order splines, if more manual positions are provided. Browsing between the images will show the interpolated (or extrapolated) target position indicated with a yellow circle. If the target is incorrectly identified in some images, click on it again to mark it manually (red circle) which will automatically update the spline interpolation. Once the target is properly identified in all images, hit *q* to close the window to write the positions file (*positions.dat*). A few notes: (1) `pp_manident()` uses WCS coordinates to identify the target; the images do not necessarily have to be registered, i.e., the fake WCS information provided by `pp_prepare()` will work perfectly fine, allowing the user you to apply this function also on un-registered images; however, be aware that the coordinates listed in the *positions.dat* file might not be *real* RA and Dec; (2) `pp_manident()` relies on source catalogs created by `pp_extract()` so either `pp_extract()`, or better `pp_photometry()` have to be run over the images previously; please refer to the *Manual Target Identification* walkthrough for a recipe on how to use this function.

pp_combine ([*-comoving*][, *-targetname str*][, *-manual_rates float*, *float*][, *-method {average*, *median*, *clipped}*][, *-backsub*][, *-keep_files*], *images*)
image combination

Parameters

- **comoving** – if used, the images will be combined in the moving frame of a moving target; the target name will be taken from the `OBJECT` header keyword or the `targetname` parameter
- **targetname** – manual override for the target name if `comoving` parameter is used
- **manual_rates** – use manual rates instead of queried ephemerides; in units of arcsec per second in RA and Dec; RA rate includes factor of cosine Dec
- **method** – image combination method: [average, median, clipped] as provided by SWARP
- **backsub** – if used, the background will be subtracted from each frame prior to image combination
- **keep_files** – if used, intermediate files are not deleted
- **images** – images to run *pp_manident* on

This function allows the combination of images using different methods. The function makes use of the SWARP software. By default, images are combined in the rest frame of the background (stars are enhanced, moving objects are partially removed); the `-comoving` option enables the combination in the moving frame of one target. In the latter case, images are shifted based on target ephemerides; manual rates can be provided, too. For details on the combination process, please refer to the SWARP manual. Image files produced by `pp_combine` can be used in any other PP function.

pp_stackedphotometry ([*-comoving*][, *-filter str*][, *-method {average, median, clipped}*][, *-fixed_aprad float*][, *-snr float*][, *-solar*], *images*)
perform automated aperture photometry on stacked images

Parameters

- **comoving** – if used, the images will be combined in the moving frame of a moving target; the target name will be taken from the `OBJECT` header keyword or the `targetname` parameter
- **filter** – manual override for the filter band
- **method** – image combination method: [average, median, clipped] as provided by SWARP; default: clipped
- **fixed_aprad** – use fixed aperture radius for aperture photometry instead of performing a curve-of-growth analysis
- **snr** – minimum SNR for sources to be identified
- **-solar** – the photometric calibration (*pp_calibrate*) is only using stars with solar-like colors (see *pp_calibrate* documentation for details)
- **images** – images to run *pp_stackedphotometry* on

This function stacks the images provided in the background frame (*skycoadd.fits*) using `pp_combine`. If the `-comoving` option is used, it also creates a combined image in the moving frame of the target provided in the `OBJECT` FITS header keyword (resulting in *comove.fits*). The combination method can be set with the `-clipped` parameter; the default is a clipped average. Note that while a median combination might produce cleaner images, it does not conserve flux; hence, you are advised not to use the median here. The magnitude zeropoint for the respective filter band (override of header filter information using the `-filter` option) is then derived from the *skycoadd.fits* image using `pp_photometry` and `pp_calibrate`. The target photometry is finally extracted using `pp_distill`. If the `-comoving` option is used, the magnitude zeropoint derived from *skycoadd.fits* is applied to *comove.fits*, from which the target's instrumental magnitude is extracted in that case.

4.6 Diagnostics

PP creates extensive diagnostic output to provide the user with a comfortable way to verify the quality of the derived results. Focusing on human readability, the diagnostic output is in the form of a HTML website that is accessible with any web browser.

An example diagnostic output is provided here.

4.6.1 Overview

HTML Output

The diagnostic output for one specific data set can be accessed by opening the `diagnostics.html` website that is by default created in the respective data directory, e.g., using:

```
firefox diagnostics.html
```

Data presented in this website are all stored in a sub-directory named `.diagnostics/` in the respective data directory (note that the dot prefix makes this a hidden directory).

LOG File

All tasks performed by PP are documented using Python's logging system. The LOG file is linked from the `diagnostics.html` file (see top of the page). The LOG file can be used to check proper performance of the pipeline. Its main purpose is to simplify debbuging in case something went wrong. The LOG file is by default created in the respective data directory.

4.6.2 Diagnostics Details

The following sections discuss the output of a successful pipeline run in detail based on this example page.

Overview Information

The top of the diagnostics website lists the data directory, some general information like the telescope/instrument combination, as well as the total number of frames processed. The process LOG file (see above) is linked here, too.

All the frames considered in the process are listed in the *Data Summary* table, providing information on the date and time, target, filter, airmass, exposure time, and field of view. By default, the frame name links to a separate page with more information on that specific frame.

Registration

The first line of this section lists the astrometric catalog that was used in the registration and whether the registration was successfull, or not. The following table lists for each frame some diagnostics provided by SCAMP, including C:subscript:*AS* and C:subscript:*XY* that indicate the goodness of the frame's rotation angle and scale, and the frame's shift, respectively. σ_{RA} and σ_{DEC} provide a sense for positional uncertainties in arcsecs; $\chi^2_{\mathrm{Reference}}$ and $\chi^2_{\mathrm{Internal}}$ quantify the goodness of fit. For additional information on these parameters, please refer to the SCAMP manual. By default, the image filenames lead to a separate frame report page.

Instrumental Photometry

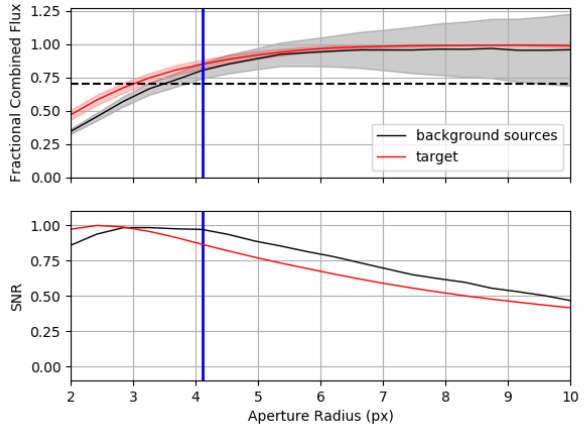
The optimum aperture size used by PP is determined using a curve-of-growth analysis. Details on how the optimum aperture was derived is provided in this section.

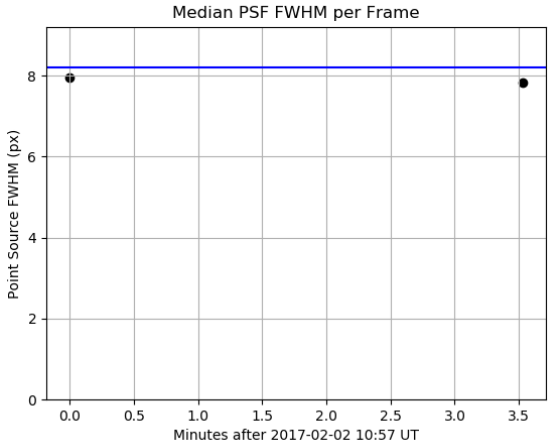
The **left plot** documents the curve-of-growth analysis. For a range of aperture sizes, the flux and SNR of both the target (red line) and the average over all background sources (black line) are determined. Note that both the flux and the SNR are fractional values - they are relative to the maximum flux (of either the target or the background stars) or the maximum SNR. In order to obtain reliable photometry, you want to include as much flux as possible but at the same time keep the aperture as small as possible, in order to minimize noise introduced by the background - this is reflected by the SNR's peak (see, e.g, Howell's *Handbook of CCD Astronomy* for a discussion). PP deliberately does not use the aperture radius that provides the highest SNR. The optimum aperture radius is required to include at least 70% of the both the target's and background sources' flux - at the same time, the fractional flux difference between the target and background sources has to be smaller than 5%. Both conditions minimize the effect of potential trailing on the photometry results. The optimum aperture radius is chosen as the smallest aperture radius that meets the conditions listed above.

The **right plot** shows the median PSF full-width-half-max (FWHM) based on all sources in the field as a function of time. The red line indicates the optimum aperture diameter for comparison. The measured FWHMs should be below the blue line, meaning that the aperture diameter is slightly larger than the image FWHM. Variations in the FWHM can be caused by seeing variations and or focus shifts. Note that in the case of badly focused images, the measured FWHM is a bad indicator of the real FWHM. By default, you can click on the data points on the FWHM plot, which will take you to the respective frame page.

Instrumental Photometry

Photometry Method	Aperture Photometry
Source Extractor MINAREA (px)	12.0
Source Extractor Detection Threshold (σ)	1.5
Aperture Radius (px)	4.11
Aperture Radius Basis	20 target detections and 20 background detections
Aperture Radius Strategy	target+background fluxes > fluxlimit, flux difference < margin





Photometric Calibration - Catalog Match

PP provides an automated photometric calibration based on a number of different star catalogs. The calibration process is summarized on the diagnostics website with a plot of the magnitude zeropoint as a function of time:

Variations in the magnitude zeropoint are due to changes in the airmass, as well as due to transparency and seeing variations (see FWHM plot above).

In addition to the overview plot, PP provides detailed information on the respective frame page as shown below:

The *Calibration Analysis* shows the magnitude zeropoint (red line) as a function of the number of background catalog stars used in the calibration. The number of background stars is reduced by rejecting the most significant outlier at a time. The blue line shows the reduced χ:superscript:*2* of the remaining data points; the dotted blue line indicates a reduced χ:superscript:*2* of 1. Currently, 50% of all background stars are rejected (vertical line) based on their weighted residuals; weights account for photometric uncertainties and catalog uncertainties. This representation shows that the magnitude zeropoint does not depend on the number of background stars used in the calibration. Further options include a *Calibration Map*, the actual image overplotted with those stars that were used in the final calibration, and the *Calibration Data Table* listing all background stars used in the final calibration.

Photometry Results

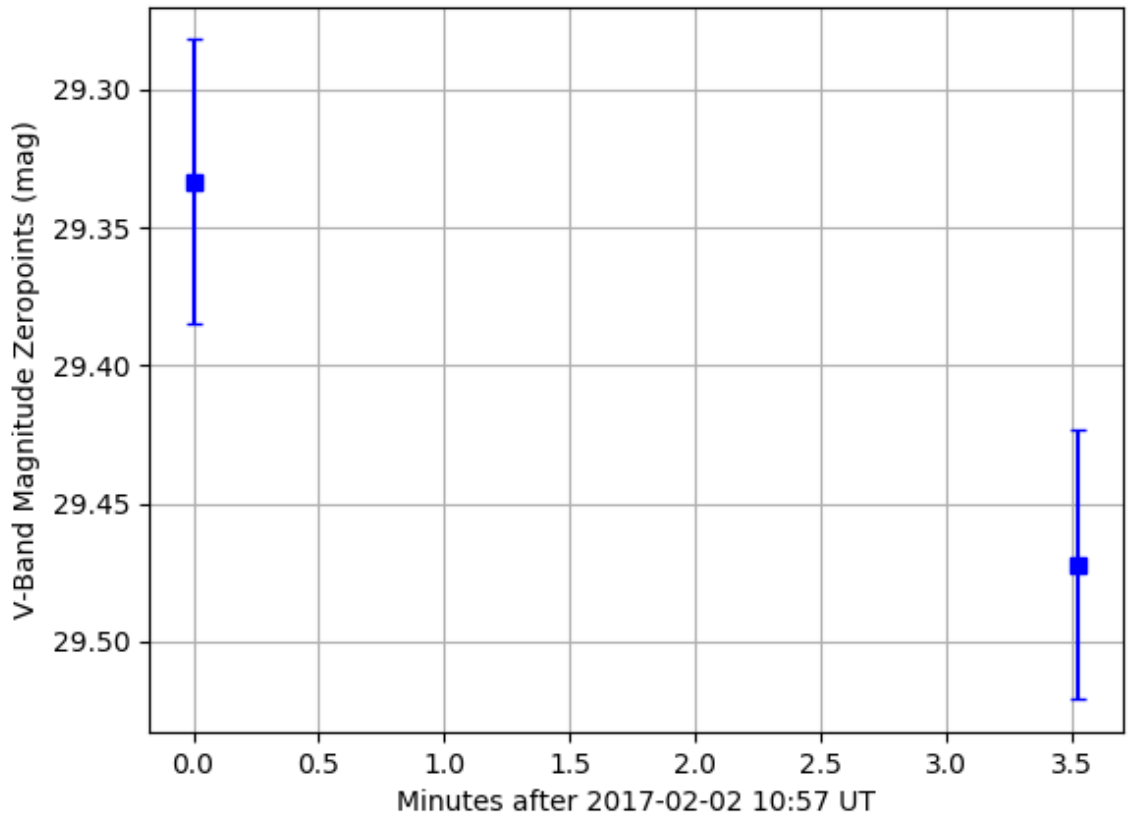
PP provides final photometry for the actual target(s) in the field, as well as for one reasonably bright star that acts as a **Control Star**. The Control Star is derived using the exact same calibrations and routines as the target, providing a good verification of the whole process. Usually, the Control Star should have a flat lightcurve that does not show significant variations. However, it cannot be ruled out that the comparison star shows intrinsic variability, or is subject to detector effects, leading to photometric variability. The comparison star is required to be present in the first and the last image of the sequence of images provided.

For each target, PP provides a GIF animation and a lightcurve showing calibrated photometry. The individual frames

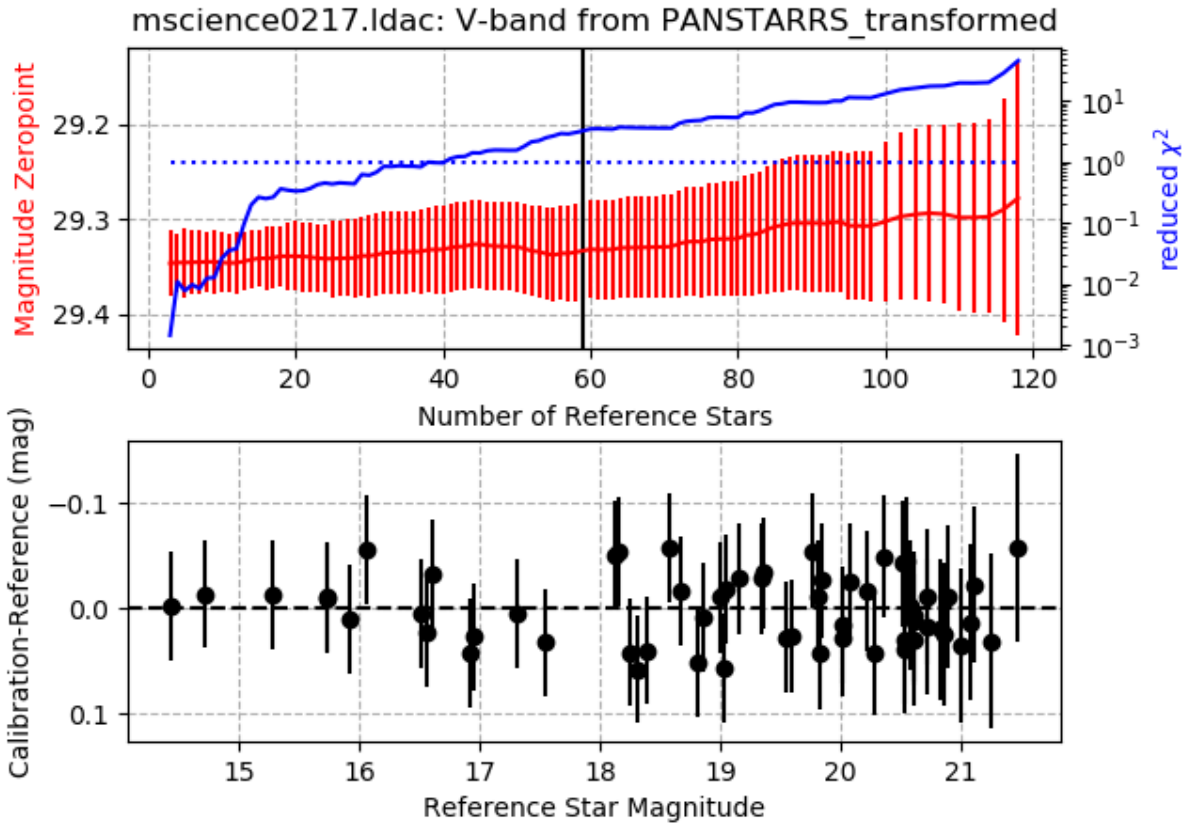
Photometric Calibration

Reference Catalog	PANSTARRS_transformed
Reference Catalog History	210 sources downloaded, 210 transformed to V (Vega)
Target Filter	V

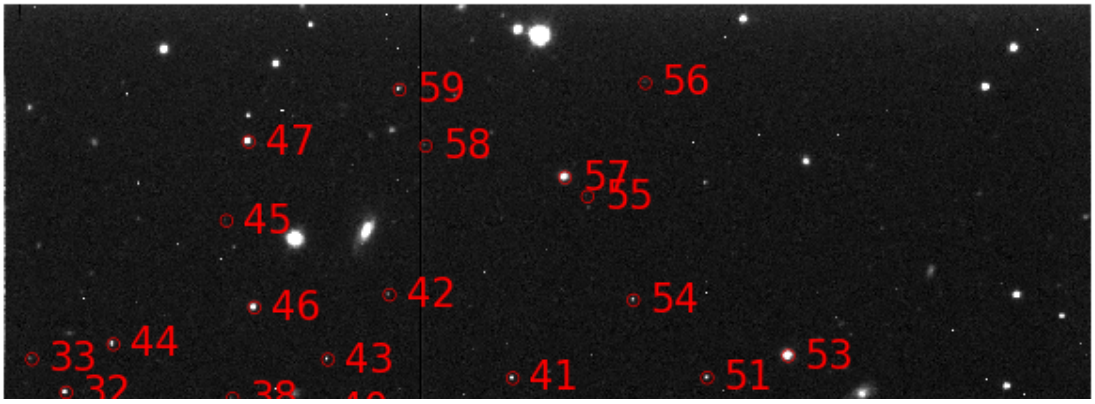
Filename	Zeropoint (mag)	σ (mag)	N^*_{used}	$N^*_{matched}$
mscience0217.fits	29.3333	0.0516	59	118
mscience0218.fits	29.4724	0.0488	59	119



Calibration Analysis

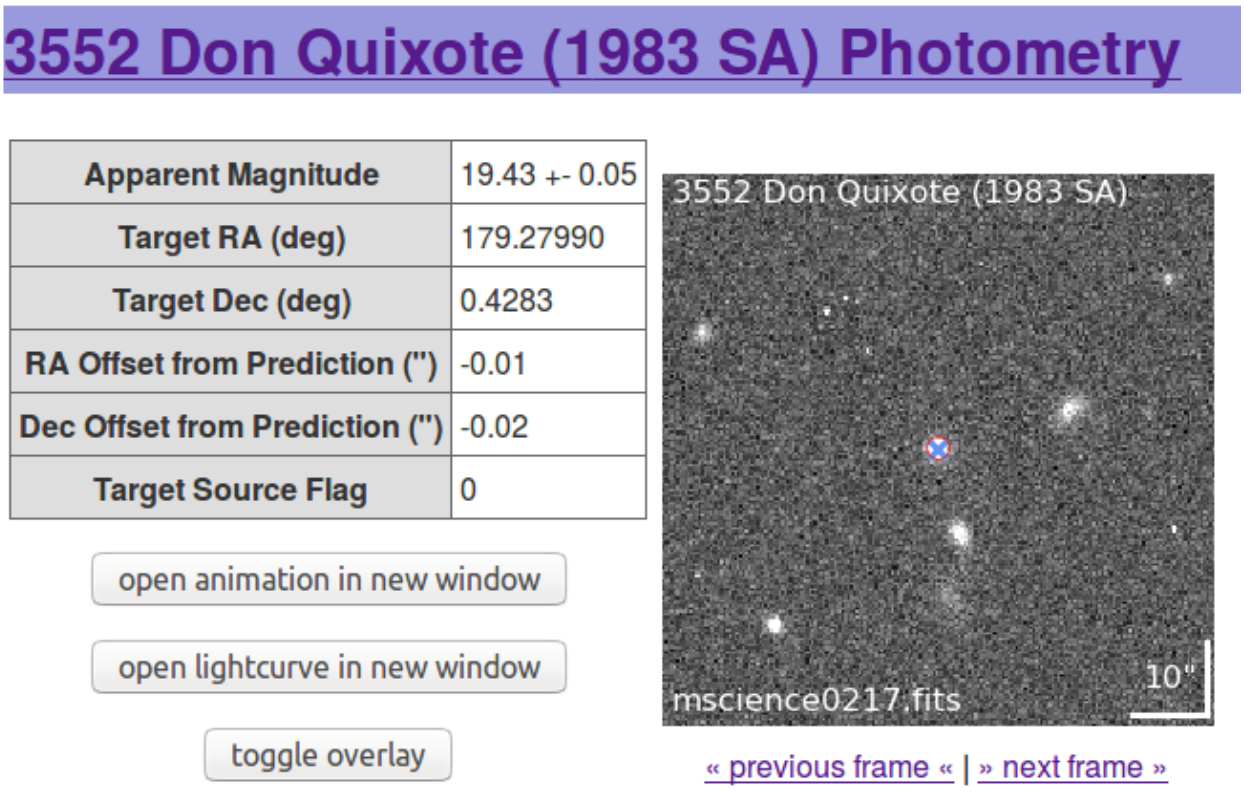


Calibration Map



in the GIF show the expected target position (blue cross) and the actual aperture placement and size used (red circle). The GIF allows for identifying target mismatches and contaminations of the photometry aperture. A is featured for each target, allowing for a quick and easy identification of corrupted frames.

The individual frame page allows for inspecting the corresponding thumbnail image in which the overlay information can deactivated:



4.7 Problems?

4.7.1 Frequently Asked Questions

General Problems

The pipeline does not run and gives me an error similar to 'pp_run: Command not found'. What is wrong?
There are two things to check: (1) did you properly setup the *PHOTPIPEDIR* environment variable (check this by typing `echo $PHOTPIPEDIR` in a terminal), or (2) the command `pp_run` uses a symbolic link to `pp_run.py`; if the former does not work, try the latter.

****Running the pipeline creates warnings (e.g., FutureWarning); what**

should I do?** Warnings are usually caused by Python module version issues and cause no harm whatsoever. In theory, there should be no warnings, but they happen occasionally. If you are haunted by some warnings, let me know and I will try to resolve the issue.

****I get an error message like 'Intel MKL FATAL ERROR: Cannot load**

libmkl_avx2.so or libmkl_def.so.‘. What do I have to do?** This seems to be a problems with your numpy or scipy installation. Try to install the latest versions of both packages. If you are using anaconda, try *conda install -f numpy* and *conda install -f scipy*.

pp_calibrate (Photometric Calibration)

I keep getting output like ‘zeropoint for image.fits: Warning: 0 reference stars after source matching for frame image.ldac‘. Wh
It means that none of the reference stars with measured magnitudes in your field of view could be matched with a source in your image. As a result, the magnitudes in the photometry output files are simply instrumental magnitudes, not calibrated ones. Try using a different photometric catalog in `pp_calibrate()`. If your field of view is small (<2 arcmin), there might just be no stars with known magnitudes in the field, in which case there is not a lot that can be done...

The pipeline fails to derive useful magnitude zeropoints from 2MASS data This problem might be solved by installing the latest version of astropy (currently 2.0.2).

The pipeline crashes in ‘‘pp_calibrate‘‘ with the following error message: ‘IndexError: too many indices for array‘. ** Well, this is embarassing... I am familiar with this problem, but I haven't found a way to solve it, yet. The problem is that Source Extractor runs in ‘‘pp_photometry‘‘ to produce an array of aperture photometry results for the curve-of-growth analysis. After that, Source Extractor is supposed to run then once again using the optimum aperture radius. Sometimes, this second Source Extractor runs seems to fail, causing ‘‘pp_calibrate‘‘ to fail. A manual workaround is to run ‘‘pp_photometry‘‘ again using the ‘‘-aprad‘‘ option and the optimum aperture derived from the last run. Running ‘‘pp_calibrate‘‘ again will then succeed. -Update: this problem should now be fixed. If you still encounter problems, please let me know!**

pp_distill (Target Photometry Extraction)

Why does the target photometry vary wildly, although the magnitude zeropoints are consistent with each other?

Check the offset of the photometry position to the expected target position. If the offsets are not consistent, it is likely that PP picks up noise or nearby sources (also check the target thumbnail images). This can happen if the target is very faint (in this case, lower the *snr* limit in `pp_photometry()`), or if the *snr* limit and *minarea* paramters are picked too small and too much noise is picked up. In either case, redo the `pp_photometry()` step and play with the *snr* and *minarea* parameters.

4.8 Changelog

Major changes to the pipeline since 2016-10-01 (see Mommert 2017) are documented here.

- 2018-12-02: major overhaul of diagnostic output
- 2018-11-23: implementation of `pp_setup.py`, which will eventually replace `_pp_conf.py`; photometric catalog data used in the photometric calibration can now by output as ascii table and/or into final .db file; `pillow` has been replaced by `skimage`; `pandas` is now a required Python module
- 2017-11-24: implementation of *-solar* option in `pp_run` and `pp_calibrate` to obtain photometric calibration only from stars with Sun-like colors
- 2017-10-20: implementation of `pp_stackedphotometry`, providing automated image stacking and subsequent photometry
- 2017-10-04: implementation of `pp_combine`, which enables automated image combination
- 2017-08-29: implementation of Gaia/TGAS as an alternative astrometric catalog for shallow widefield observations

- 2017-06-01: extraction of serendipitously observed targets (asteroids and variable stars) implemented in `pp_distill`
- 2017-03-19: if no filtername is provided (`None`) or the `-instrumental` option of `pp_calibrate` is used, `pp_run` will complete all pipeline tasks using these instrumental magnitudes
- 2017-03-16: implementation of Pan-STARRS DR1 for the photometric calibration; currently, PP uses a home-built access of MAST at STScI, which is limited to catalog queries with a maximum cone radius of 0.5 deg; please note that this kind of query is rather slow compared to Vizier queries of SDSS or APASS
- 2017-02-24: `pillow` is now a required python module; the pipeline now supports default distortion parameters for wide-field cameras
- 2017-02-04: catalog.data is now an astropy.table, catalog downloads using astroquery.vizier (no effect to the user), pp now supports Gaia DR1 (CMC, USNOB1, and PPMX have been removed)

CHAPTER 5

License and Contact

CHAPTER 6

Acknowledgments

If you are using PP for your research, please acknowledge PP by citing

- Mommert, M. 2017, PHOTOMETRYPIPELINE: An Automated Pipeline for Calibrated Photometry, Astronomy & Computing, 18, 47.

PP is supported by NASA grants NNX15AE90G and NNX14AN82G and has been developed in the framework of the Mission Accessible Near-Earth Objects Survey (MANOS).

CHAPTER 7

Indices and tables

- genindex
- modindex
- search

Index

P