University of Nevada, Reno - Department of Computer Science and Engineering

in collaboration with

University of Nevada, Reno - Department of Physics

Guide to Photometry+

v3.0

Presented by the Photometry+ Development Team

Last Updated April 7th, 2021

Introduction

Differential photometry is the tool astrophysicists use to calculate the apparent brightness of a star, a process that involves counting the relative amount of photons that hit a pixel of a CCD from a certain area in the sky, and using those counts to estimate magnitude. To calculate the magnitude of a target star, several reference stars of known magnitude are used for comparison. The process of finding not only the counts in a target star, but also in several reference stars, is a long one. Doing this process by hand can take up to a half hour to do just one piece of data, and to create a light curve many pieces of data must be analysed.

That is why Photometry+ was created. Photometry+ seeks to not only make this process fast, but to make it easy and user-friendly as well. Photometry+ automates the process of performing differential photometry now, and in the future will allow users of all skill levels to perform this essential skill through the employ of cutting edge human-computer interaction (HCI) research. This document contains documentation, help, and guidance on how to use version 3.0 of Photometry+.

Returnable Objects

In order to keep returned data clean and accessible, Photometry+ often returns data in the form of object instances (see Glossary for more information). The data in all instances returned by Photometry+ can be accessed in Python in the following way, where the objectName is the variable name of the instance, and desiredData is the object characteristic you want to access:

objectName.desiredData

In some cases, an array of objects will be returned. This works the same way as above, however the instance in the array that will be accessed, i, will need to be specified:

objectName[i].desiredData

In most cases, the data will not need to be accessed directly but rather passed into a different Photometry+ method (described more in the Method section) in its entirety. This section is a reference for the few occasions when the data will need to be accessed individually.

Star Object

The Star object is an Object that contains information about a certain star. The data contained in an instance of the Star object is as follows:

Variable Name	Type	Description
id	string	The id is the identifier used to distinguish this star from other stars. It can be a name or something simple, like “Ref1”.
ra	float	The ra is the right ascension of the star in decimal degrees.
dec	float	The dec is the declination of the star in decimal degrees.
radius	int	The radius is the radius of the star being used for photometry, this can also be set by the user on a per star basis.
counts	float	The counts represent the number of photons captured within the star’s radius per photometric image.
magnitude	float	The magnitude contains the brightness of a star in units of magnitude (see Glossary for more information).
targetMagnitude	float	The targetMagnitude is the magnitude a reference star has calculated for a target star chosen by the user.
error	float	The error is the measure of the uncertainty in the targetMagnitude calculation.

Star objects are returned in some form from findOtherStars, readFromFile, calculateMagnitudeAndError, removeReferenceOutliers, createCMD, and findAllStars. Additionally, Star objects are used in the Photometry objects.

Photometry Object

The Photometry object is a data type that contains information about the results of the photometry process. This is the “final result”of performing differential photometry on an image. The data contained in an instance of the Photometry object is as follows:

Variable Name	Type	Description
fileName	string	The fileName is the name of the photometric image used to do differential photometry.
JD	float	The JD is the Julian date of the observation.
magnitude	float	The magnitude is the final calculated magnitude of the

		target star specified by the user.
error	float	The error is the measure of the uncertainty in the magnitude calculation.
referenceStars	Star	The referenceStars is an array of Star objects representing the reference stars used to calculate the magnitude of the user-specified target star.

Photometry objects are returned from runPhotometry, getFiles (in an incomplete format), and runFiles.

Settings Object

The Settings object is very important, and it will be discussed in its own section. The settings object is never returned or passed into any of the methods in Photometry+, but rather is used as a global variable to keep track of user preferences.

Settings

There are a wide variety of settings that can be used to change how photometry is done, and the settings are controlled by a global variable internal to the program. The settings are changed using a method called “changeSettings” which is designed to allow the user to permanently set their preferences, change preferences as needed, and maintain previously used preferences while the program is running. To change a setting, a user would use the following syntax, where settingToChange is the setting to be set, and newSetting is the new state of that setting:

```
photometry.changeSettings(settingToChange=newSetting)
```

The user can also change multiple settings at once, as shown below:

```
photometry.changeSettings(settingToChange1=newSetting1, settingToChange2=newSetting2)
```

Additionally, all settings can be retrieved through methods named “get[setting name here]”, as shown below for the setting “settingToGet” (note the capitalization):

```
settingVariable = getsettingToGet()
```

These settings are maintained between method calls, but not separate runs of the program. The settings that can be changed start on the next page (the default setting is bolded):

Name	Description	Possible Settings
subtractBiasFromDarkFlag[1]	Flag to determine whether to subtract the bias frame from the dark frame prior to subtracting both from the main file in calibration.	0 = leave dark and bias frames as they are originally **1 = subtract the bias frame from the dark frame prior to calibration**
useDarkFlag	Flag to determine whether the dark frame is used in calibration.	0 = do not use dark file in calibration **1 = subtract dark file from main file during calibration**
useBiasFlag	Flag to determine whether the bias frame is used in calibration.	0 = do not use bias file in calibration **1 = subtract bias file from main file during calibration**
calibrationOutputFlag	Flag to determine whether the results of calibration is output as a .fits file.	**0 = calibrated file not outputted as a .fits file** 1 = calibrated file outputted as a .fits file after calibration
calibrationFlag	Flag to determine whether to calibrate a main file prior to calculating the magnitude of any stars.	0 = calibration not performed before photometry **1 = calibration performed before photometry**
blankPerStarFlag	Flag to determine whether the counts in a pixel of blank sky are determined for the entire image or individually for each star.	**0 = calculate blank sky counts for entire image** 1 = calculate blank sky counts around each individual star
catalogChoice	When automatically finding reference stars for differential photometry, catalogChoice determines what Vizier catalog is used to find the magnitudes for reference stars.	0 = use SIMBAD[2] **"II/336/apass9" = use APASS catalog** Any other Vizier catalog designation = magnitudes are pulled from the designated Vizier catalog
filterChoice	When automatically finding reference stars for differential photometry, it is important that the filter used for the image is chosen using filterChoice.	**"V" = Johnson V filter** "B" = Johnson B filter "g" = g' filter "r" = r' filter

[1] Note that all calibration frames from the Great Basin Observatory have the bias included in the dark frame, and will need this flag to be set to 1 for accurate calibration.

[2] Note that due to the variance in where SIMBAD magnitudes are from, using SIMBAD may introduce error into the calculation.

		"i" = i' filter Other filter = when using non-supported filters, make sure that the format of filterChoice matches the format for that filter used by the catalog set in catalogChoice, non-supported filters will not be formatted automatically by Photometry+
lightCurveLineFlag	Flag to determine whether light curves are printed with a smooth line in between each point (non-standard) or without a line.	**0 = generate light curve without line** 1 = generate light curve with smooth line
showLightCurveFlag	Flag to determine whether created light curves are displayed on the screen after being created.	**0 = do not show light curve on screen after creation** 1 = show light curve on screen after creation
printLightCurveFlag	Flag to determine whether created light curves are printed to a file after creation.	0 = do not print light curve to a file after creation **1 = print light curve to a file after creation**[3]
showCMDFlag	Flag to determine whether created CMDs are displayed on the screen after being created.	**0 = do not show CMD on screen after creation** 1 = show CMD on screen after creation
printCMDFlag	Flag to determine whether created CMDs are printed to a file after creation.	0 = do not print CMD to a file after creation **1 = print CMD to a file after creation**[4]
errorChoice	When calculating the magnitude of a star, errorChoice determines how the uncertainty of the calculation is calculated.	**"STD" = calculate error with standard deviation** "JKF" = calculate error with jack knife method (Andserson, 2000) "WMG" = calculate error with weighted magnitudes

[3] Note that if both showLightCurveFlag and printLightCurveFlag are set to 0, the light curve will be created and then deleted without ever being output in any format.

[4] Note that if both showCMD and printCMD are set to 0, the CMD will be created and then deleted without ever being output in any format.

		file name = enter a .csv file name that contains stars of known magnitude to calculate error based on calculated magnitudes for them 0 = do not calculate error
consolePrintFlag	Flag to determine whether updates on the program's progress are printed to the console or not.	0 = do not print updates to console **1 = print updates to console**
readInReferenceFlag	Flag to determine whether reference stars are found or read in from a file.	**0 = find reference stars automatically** file name = enter a .csv file name that contains reference stars to use directory name = detect reference star .csv file in this directory
readInRadiusFlag	Flag to determine whether radius for reference stars read in from file are pulled from the file or are just set to be the target star radius.	**0 = all stars use target star radius** 1 = radius for stars are pulled from .csv file
fwhmFlag	Flag to determine how target star radius is chosen.	0 = detect star radius automatically **1 = use three times full-width half-maximum as radius if available** Any other number = this number will be used as the target star radius
printReferenceStarsFlag	Flag to determine if found reference stars are printed out to a .csv file.	**0 = reference stars not printed to file** 1 = reference stars printed to file
astrometryDotNetFlag	Flag to determine if Astrometry.net is used to add world coordinate system data to files without it.	**0 = Astrometry.net not used** Astrometry.net API key = Astrometry.net used with this account API key[5]
astrometryTimeOutFlag	Flag to determine when to stop waiting for Astrometry.net to generate results	**0 = run Astrometry.net until results come back** Any other number = run Astrometry.net for this many iterations before aborting
removeReferenceOutliersFlag	Flag to set the program to	0 = do not remove outliers

[5] See following link for information about Astrometry.net and Astrometry.net API key: http://nova.astrometry.net/

	automatically remove outlier reference stars based on z-score.	automatically **3 = remove reference star outliers with a z-score with an absolute value higher than 3** Any other number = remove reference star outliers with a z-score with an absolute value higher than this number
outputDetailedInformation	Setting to allow the program to output files with detailed information for each result, reference star, and error star. Only usable with runFiles.	**0 = do not output detailed information** 1 = output detailed information files
oneMagnitudeCalculation	Setting for how to calculate the magnitude of the target object	**0 = calculate target object magnitude for each reference star, then average those magnitudes together** 1 = average reference star counts and magnitudes and then use those averages to calculate the target object magnitude once
projectName	Flag used by the interface to keep track of project names.	Not accessible to users, managed by the program.
rightAscension, declination	Flag used by the interface to keep track of project coordinates.	Not accessible to users, managed by the program.
main, dark, bias, flat	Flag used by the interface to keep track of project files.	Not accessible to users, managed by the program.
universalBlank	Flag used by the program to keep track of blank sky counts.	Not accessible to users, managed by the program.
coordinateChoiceFlag	Flag used by the interface for the way coordinates were entered.	Not accessible to users, managed by the program.
interface	Setting for whether the desktop or code version is being used.	Not accessible to users, managed by the program.
walkthroughMode	Setting to trigger walkthrough mode on the interface.	Not accessible to users, managed by the program.
currentErrorStarCalc	Flag for when error star magnitudes are being calculated.	Not accessible to users, managed by the program.

Important Methods

Photometry+ is used by calling various methods to perform tasks, and the methods may call other methods. In the table below, methods that are most commonly used are highlighted in green for ease of location. The methods are as follows:

Name	Description	Parameters (IN)	Returns (OUT)
calculateMagnitudeAndError	Calculates the magnitude of a target star using reference stars of various counts and magnitudes. Also calculates the error of the magnitude calculation.	**targetStarPhotons (float)** A number representing the counts in the area of the target star **stars (Star array)** An array of star objects representing reference stars for the calculation **targetStarError (float)** This is the error associated with the photon measurement for the target star **sigmaBkg (float)** This is the error associated with the measurement of the background counts	Calculated magnitude (float), error of magnitude calculation (float), stars that are visible and can calculate magnitude for the target star (array of Star instances)
calibrate	Calibrates a main file by subtracting dark frame and bias frame from the image and then dividing out a normalized flat field.	**filename (string)** The name/file path of the main file to be calibrated **dark (string)** The name/file path of the dark frame to calibrate with **bias (string)** The name/file path of the bias frame to calibrate with **flat (string)**	A calibrated, opened hdul object containing the data and header information for the calibrated main file (hdul)

		The name/file path of the flat field to calibrate with	
createCMD	Generates a CMD plot of brightness vs. color for a group of stars.	**vstars (array of Stars)** An array of stars with V magnitudes calculated for them **bstars (array of Stars)** An array of stars with B magnitudes calculated for them **chartname = "chart.pdf" (string)** The name of the chart to output to file, default is "chart.pdf" **charttitle = "CMD" (string)** The name of the chart on the chart, default is "CMD"	Star array representing V stars actually used in plot (array of Stars), star array representing B stars actually used in plot (array of Stars)[6]
createReferenceData	Separates reference stars out and plots them individually for full light curves of the target star magnitude calculated by that specific reference star.	**info (array of Photometry)** An array of photometry results representing the magnitudes and stars to decompose **targetStarName = "Star" (string)** Name of the target star for chart naming, default is "Star"	Nothing (light curves created and output as files)
getFiles	Finds all .fits or .fts files in a directory and returns them as an array of Photometry objects with only name and Julian Date.	**dirName (string)** Name of the directory to search	Array of available fits files in the directory (array of Photometry)

[6] Both arrays returned will be in the same order and contain the same stars, just different magnitudes.

getWCS	Adds world coordinate system to the header of files without world coordinate system data already. Uses Astrometry.net.[7]	**file (string)** File name/file path for the image to get world coordinate system data for **ra = 0 (float)** Starting guess right ascension, default is 0 which means Astrometry.net is run without starting guess **dec = 0 (float)** Starting guess declination, default is 0 which means Astrometry.net is run without starting guess	Name of new .fits file with world coordinate system data in the header (string)
findAllStars	Finds all stars in an image and then calculates their magnitude based on other stars.	**mainFile (string)** File name/file path for the main image **darkFrame (string)** File name/file path for the dark frame **biasFrame (string)** File name/file path for the bias frame **flatField (string)** File name/file path for the flat field	All stars found in the image with magnitude and error (array of Stars)
findBlank	Finds the median background counts per pixel for an image based on the entire image.	**data (array of ints)** Array of integers containing the image data **r (int)** Radius of target star, used to take background counts over an area four times the size of the star	Median background counts per pixel (int)
findBlankNoRad	Finds the median background	**data (array of ints)**	Median

[7] Note that the setting astrometryDotNetFlag must be set to a valid Astrometry.net API key to use getWCS.

	counts per pixel for an image based on the entire image without using a radius in the calculation.	Array of integers containing the image data	background counts per pixel (int)
findCenter[8]	Finds true center coordinates of stars when off-center coordinates are entered.	**Y (int)** Pixel y location of current star guess **X (int)** Pixel x location of current star guess **data (array of ints)** Array of integers containing the image data	Pixel y location of centered star (int), pixel x location of centered star (int)
findOtherStars	Finds reference stars in the image and constructs them into an array of Star objects.	**Y (int)** Pixel y location of target star **X (int)** Pixel x location of target star **data (array of ints)** Array of integers containing the image data **rad (int)** Radius of target star **w (WCS parameter)** The conversion parameter for conversion of pixel to world coordinate system location and vice versa	Reference stars found with withdrawn magnitudes and calculated counts (array of Stars)
findRadius	Finds the radius of a star by	**Y (int)**	Radius of

[8] As of this version of Photometry+, findCenter is still a work in progress and is not guaranteed to work without error.

	examining the image itself.	Pixel y location of star **X (int)** Pixel x location of star **data (array of ints)** Array of integers containing the image data	selected star (int)
formatFilter	Formats desired filter (V, B, etc.) into correct format for use with Vizier and SIMBAD magnitude search.	None[9]	Three strings representing formatting for Vizier and SIMBAD (strings)
runPhotometry	Completes differential photometry for a single image and target star.	**targetStarRA (float)** Right ascension of target star in decimal degrees **targetStarDec (float)** Declination of target star in decimal degrees **mainFile (string)** File name/file path for the main image **darkFrame = "" (string)** File name/file path for the dark frame **biasFrame = "" (string)** File name/file path for the bias frame **flatField = "" (string)** File name/file path for the flat field	Photometry object containing results of differential photometry (Photometry)

[9] The setting filterChoice is used as the starting place for choosing which filter to use.

matchCal	Matches a target file with calibration files closest in date and exposure time.	**filename (string)** File name/file path for the main image **cals (array of Photometry)** Photometry objects containing names, dates, and exposure times associated with calibration files.	File name/file path for closest calibration file (string)
plotResults	Plots a light curve made from the results of photometry stored in a Photometry array.	**ans (array of Photometry)** Differential photometry results to plot in the light curve **chartname = "chart.pdf" (string)** Name of the file that the light curve should be outputted to **chartTitle = "Light Curve" (string)** Name of the light curve to be put on the plot itself	Nothing (light curves created and output as files or printed to screen based on settings preference)
plotResultsFile	Plots a light curve made from the results of photometry stored in a Photometry file.	**filename (string)** File name/file path containing differential photometry results to plot in the light curve **chartname = "chart.pdf" (string)** Name of the file that the light curve should be outputted to **chartTitle = "Light Curve" (string)** Name of the light curve to be put on the plot itself	Nothing (light curves created and output as files or printed to screen based on settings preference)

printReferenceToFile	Prints reference stars to .csv file for later retrieval and review.	**stars (array of Stars)** Reference stars to be output into a file **filename = "stars.csv" (string)** Name of the file reference stars outputted to	Nothing (reference stars outputted to file)
printResultsToFile	Prints differential photometry results to a .csv file.	**info (array of Photometry)** Array of photometry results to print to file **filename = "output.csv" (string)** Name of the file to output photometry results to	Nothing (results outputted to file)
readFromFile	Reads in reference star information from a .csv file.	**fileName (string)** File name/file path to read in from **radius (int)** Radius to use for stars read in **data (array of int)** Array of integers containing the image data **w (WCS parameter)** The conversion parameter for conversion of pixel to world coordinate system location and vice versa	Reference stars from file (array of Stars)
reducedChiSquared	Calculated the χ_r^2 value of a set of photometry results.	**info (array of Photometry)** Array of photometry results to be used in calculation	The χ_r^2 value of the data (float)

removeReferenceOutliers	Examines reference stars for outliers in the calculation of the target star magnitude and removes them if they are over a certain z-score.	**stars (array of Stars)** Array of reference stars with target star magnitude calculations	Array of reference stars not removed as outliers (array of Stars)
runFiles	Performs differential photometry on all .fits files in a directory.	**targetStarRA (float)** Right ascension of target star in decimal degrees **targetStarDec (float)** Declination of target star in decimal degrees **dirName (string)** Folder name/file path for the main images **darkDirName = "" (string)** Folder name/file path for the dark frames **biasDirName = "" (string)** Folder name/file path for the bias frames **flatDirName = "" (string)** Folder name/file path for the flat fields	Array of Photometry objects containing results for every .fits file in the main directory (array of Photometry)
starCount	Sums up the counts within the area of a star.	**Y (int)** Pixel y location of star **X (int)** Pixel x location of star **data (array of ints)** Array of integers containing the image data **r (int)** Radius of the star	Amount of counts in a star's area (float), error of the counts (float), 0 or 1 flag indicating if a star possibly has another star too close to it for the calculation on 1 (int)
truncate	Truncates a number to a certain place.	**n (float)** Number to truncate **decimals (int)** Number of decimals to	Truncated number (float)

		truncate at	
worldCoordinateSystem	Returns the world coordinate system parameter of a hdul container.	**hdul (hdul)** An opened container for the .fits file information	World coordinate system parameter

Photometry+ User Interface

Video Tutorial: https://youtu.be/gPaxViG-Yd0

Starting code for user interface:

```
import photometryuserinterface
from PyQt5 import QtWidgets
import sys

if __name__ == '__main__':
  app = QtWidgets.QApplication(sys.argv)
  controller =photometryuserinterface.Controller()
  controller.show_home()
  sys.exit(app.exec_())
```

Getting Started

While most of this document is a reference, this is the section for those who need more practical help getting started. No matter how the user runs Python, these tips will help get Photometry+ running faster.

Practical Advice

The heart of Photometry+ is the photometry.py file on the github. This is where all the code and settings shown above reside and it is the start of performing photometry. If you look at photometry.py, you'll see that all methods are commented with what they do, what they take in, and what they return on top of the method. Additionally, within the method, what the method does is commented out. Don't be afraid to go to the code itself if there are questions about what a method is actually doing. To run Photometry+ as intended, you will never need to change the source code yourself. All of its functionalities can be accessed by calling its methods.

When changing settings via the changeSettings method, note that those settings will remain in place until you change them or abort the program. This can trip you up if you're using reference star files that may need to be changed in between every runFiles call. Make sure that you are aware of the settings.

Another thing to be aware of is that Photometry+ runs in decimal degrees entirely. Everywhere that the program takes in right ascension and declination, it will ask for it in decimal degrees. The formula for conversion of the normal form (hours, arcminutes, and arcseconds) to decimal degrees is:

Right Ascension, decimal deg. = Hours x 15 + (Arcminutes/60) x 15 + (Arcseconds/3600) x 15
Declination, decimal deg. = Hours + Arcminutes/60 + Arcseconds/3600

Be sure to convert your target star locations using this formula, and be sure to keep track of the final sign of the location, especially if your target has a negative coordinate.

Many aspects of the program use a "hdul" which contains information from a .fits file. The way to access this information is to use hdul[0].header['INFORMATION'] to get information from the .fits header, such as EXPTIME for exposure time or JD for the Julian date, and to use hdul[0].data to access the actual image array data.

Photometry+ does not modify original data. Anywhere a file is created, it will be a new version of the file stored in a folder called "Output" in the directory where the program is being run.

Examples

Code Example for Astrometry.net usage:

```
import photometry.py

photometry.changeSettings(astrometryDotNetFlag="validapikey")
f = photometry.getWCS("filewithoutWCS.fits", ra=175.91034694, dec=71.68911273)
```

Result:

The variable "f" now contains a string describing the name of the new file that was created that has complete WCS information in the header.

Code example for calibration 1:

```
import photometry.py
```

```
photometry.changeSettings(calibrationOutputFlag=0)
hdul = photometry.calibrate("mainfile.fits", "darkframe.fits", "biasframe.fits", "flatfield.fits")
```

Result:
The variable “hdul” now contains an unwrapped .fits file information including the data and header information. No file is outputted.

Code example for calibration 2:

```
import photometry.py

photometry.changeSettings(calibrationOutputFlag=1)
hdul = photometry.calibrate("mainfile.fits", "darkframe.fits", "biasframe.fits", "flatfield.fits")
```

Result:
The variable “hdul” now contains an unwrapped .fits file information including the data and header information. The calibrated file is outputted to a new .fits file, leaving the original file intact.

Code example for locating reference stars:

```
import photometry.py

hdul = photometry.calibrate("mainfile.fits", "darkframe.fits", "biasframe.fits", "flatfield.fits")
w = photometry.worldCoordinateSystem(hdul)
blank = photometry.findBlankNoRad(hdul[0].data)
stars = photometry.findOtherStars(2000, 2000, hdul[0].data, 25, w)
```

Results:
The variable “stars” now contains an array of Star objects representing all stars in the image with a valid V magnitude (the default filter setting) pulled from the APASS catalog in Vizier.

Code example for printing reference stars to file:

```
import photometry.py

hdul = photometry.calibrate("mainfile.fits", "darkframe.fits", "biasframe.fits", "flatfield.fits")
w = photometry.worldCoordinateSystem(hdul)
blank = photometry.findBlankNoRad(hdul[0].data)
stars = photometry.findOtherStars(2000, 2000, hdul[0].data, 25, w)
photometry.printReferenceToFile(stars, filename="referencestars.csv")
```

Results:
A new file named referencestars.csv is created that contains the reference star information.

Code example for reading reference stars from a file:
import photometry.py

```
hdul = photometry.calibrate("mainfile.fits", "darkframe.fits", "biasframe.fits", "flatfield.fits")
w = photometry.worldCoordinateSystem(hdul)
stars = photometry.readFromFile("Output/referencestars.csv", 25, hdul[0].data, w)
```

Results:
The variable "stars" now contains an array of Star objects representing all stars in the .csv file specified.

Code example for full differential photometry of one image:
import photometry.py

```
ans = []
ans.append(photometry.runPhotometry(119.721, 16.279214, "mainfile.fits", darkFrame="darkframe.fits", biasFrame="biasframe.fits", flatField="flatfield.fits"))
```

Results:
The variable "ans" now holds an array with the [0] spot filled with a Photometry object containing the results of differential photometry on the star specified by the stellar coordinates in the main file.

Code example for performing photometry on all files in a directory:
import photometry.py

```
ans = photometry.runFiles(52.800196, 43.90441,
                "MainFolder", "DarkFolder", "BiasFolder", "FlatFolder")
```

Results:
The variable "ans" now holds an array made of Photometry objects representing photometry done on every file in MainFolder for the target star specified by the stellar coordinates.

Code example for printing results of photometry to a file:

[10] Most Photometry+ methods that take in Photometry objects take them in as an array, so it is good practice to always store Photometry results in an array.

```
import photometry.py

ans = []
ans.append(photometry.runPhotometry(119.721, 16.279214, "mainfile.fits",
darkFrame="darkframe.fits", biasFrame="biasframe.fits", flatField="flatfield.fits"))
photometry.printResultsToFile(ans, filename="results.csv")
```

Results:
A new file named results.csv is created containing the results of differential photometry stored in ans.

Code example for plotting results of photometry from a file:

```
import photometry.py

photometry.plotResultsFile("Output/results.csv", chartname="chart.pdf", chartTitle="Chart")
```

Results:
A new file named chart.pdf is created that contains a light curve named Chart.

Code example for plotting results of photometry from a Photometry object:

```
import photometry.py

ans = photometry.runFiles(52.800196, 43.90441,
                "MainFolder", "DarkFolder", "BiasFolder", "FlatFolder")
photometry.plotResults(ans, chartname="chart.pdf", chartTitle="Chart")
```

Results:
A new file named chart.pdf is created that contains a light curve named Chart.

Debugging and Error Support

While Photometry+ seeks to prevent catastrophic errors, there may still be some problems running the program. This section helps with debugging and should provide a useful companion for running the program.

Error Messages

Photometry+ is equipped with comprehensive error reporting when the consolePrintFlag is set to 1. Not all errors cause a program abort, but it is best to be aware of them. Below is a list of all the errors that can be generated by Photometry+ and what they could mean for you:

Error Message	What it Means	Possible Solutions
Method: calibrate "Error in calibrate: Invalid filename: <filename>"	One of the files used for calibration cannot be opened. The <filename> indicates which file is having problems.	Check to see if the filename is correct, and the filepath is accurate. Check and see if the file is a valid .fits file.
Method: findCenter "WARNING: Find center aborted because of out of bounds error. Returning starting coordinates."	When looking for a center, the program encountered the edge of the image.	The target star may be too near the boundary of the image. findCenter is a WIP method, so try to eliminate it from use.
Method: findOtherStars "Connection Error: Check your internet connection before searching for reference stars"	Photometry+ is unable to connect to the internet to search the database for other stars.	Try reconnecting internet access.
Method: findOtherStars "Catalog has no valid entries. Switching to SIMBAD."	The selected Vizier catalog has no option for stars in the image, or there has been a problem finding stars in the selected catalog.	Select a new catalog, check your filterChoice setting for formatting, or use SIMBAD if accuracy is not critical.
Method: findOtherStars "WARNING: Using SIMBAD may introduce error into the calculations due to discrepancies in where magnitudes were obtained."	SIMBAD is being used and may cause errors to be higher due to inaccuracies in SIMBAD's magnitudes.	Choose a Vizier catalog to reduce error.
Method: findOtherStars "No reference stars found."	No suitable reference stars were found in the image with an appropriate magnitude using either the selected catalog or SIMBAD.	Check your settings for filterChoice, there may be incorrect formatting. Check if stars are visible in the image. Select reference stars by hand.
Method: readFromFile "Error: Invalid file for	Reference star file is unable to be opened.	Check to see if the filename is correct, and the filepath is

reading in reference stars"		accurate. Check and see if the file is a valid .csv file.
Method: reducedChiSquared "Reduced Chi Squared error: Reduced chi squared cannot be calculated for <0,1> value(s)."	χ_r^2 can only be calculated for arrays with more than one value.	Ignore if printing one photometry result to file. Check data if this error comes up for more than one Photometry result.
Method: reducedChiSquared "Reduced Chi Squared error: Reduced chi squared calculation needs valid error values."	The Photometry results passed to the function do not include error values.	Change errorChoice settings to an acceptable value. Make sure error is included in Photometry results.
Method: plotResultsFile "Error: Invalid file for plotting results"	Results file is unable to be opened.	Check to see if the filename is correct, and the filepath is accurate. Check and see if the file is a valid .csv file.
Method: calculateMagnitudeAndError "Calculate magnitude error: Magnitude cannot be calculated for any reference star"	None of the reference stars have produced a valid magnitude calculation.	Check to see if reference stars are of a magnitude not visible in the image. Check to see if reference stars are out of bounds. If using autonomously detected reference stars, consider passing in your own choices for reference stars.
Method: getWCS "Astrometry.net unavailable: Please set astrometryDotNetFlag setting to a valid Astrometry.net API Key"	The astrometryDotNetFlag is not set.	Set astrometryDotNetFlag to a valid Astrometry.net API key to use this functionality.
Method: getWCS	Login to Astrometry.net	Make sure

"Astrometry.net login error: Check if login is accurate and internet connection is secure"	failed.	astrometryDotNetFlag is set to the accurate API key for Astrometry.net. Make sure the program is connected to the internet.
Method: getWCS "Astrometry.net submission job ID cannot be retrieved."	Astrometry.net jobid search has timed out.	Change the setting astrometryTimeOutFlag to be a larger number for longer searching, or 0 to search until jobid is found.
Method: getWCS "Astrometry.net failed to solve submission."	Astrometry.net failed to localize the image.	Try a more accurate starting guess location. Remove this file from the dataset.
Method: getWCS "Astrometry.net submission timed out."	The connection to Astrometry.net was ended by Astrometry.net.	Wait a bit and then try again.
Method: runPhotometry "runPhotometry on <mainFile> aborted due to calibration error"	The file was not able to be calibrated.	Check other calibration errors to see what problem happened in the calibration method.
Method: runPhotometry "runPhotometry on <mainFile> aborted due to invalid target star coordinates."	The target star coordinates are out of bounds of the image.	Make sure the target star coordinates are correct. Make sure the target star is actually visible in the image.
Method: runPhotometry "runPhotometry on <mainFile> aborted due to localization error." "Suggestion: Set the astrometryDotNetFlag to a valid Astrometry.net API Key to localize files without world coordinate system data."	The file does not have the world coordinate system data required for localization and calculation.	Set the astrometryDotNetFlag to a valid Astrometry.net API key to attempt to localize the image using the Astrometry.net service.
Method: runPhotometry "runPhotometry on <mainFile> aborted due to problem finding reference	Reference stars unable to be located in the field.	Check your settings for filterChoice, there may be incorrect formatting. Check if stars are visible in the image.

stars in field"		Select reference stars by hand.
Method: runPhotometry "Reading in reference stars from file failed, finding stars automatically:"	Reference stars unable to be read in from file, the program will try to find reference stars automatically.	Check your reference star file for formatting. Check that all reference stars are visible in the image. Output found reference stars and curate from that list instead of from scratch.
Method: runPhotometry "runPhotometry on <mainFile> aborted due to problem calculating magnitude"	Magnitude unable to be calculated.	Check to see if reference stars are of a magnitude not visible in the image. Check to see if reference stars are out of bounds. If using autonomously detected reference stars, consider passing in your own choices for reference stars. Make sure the target star is visible in the image.
Method: runPhotometry "WARNING: There may be another star near the target star."	Another star was detected in close proximity to the target star, within its radius of calculation.	Manually set a smaller radius for the star calculations. Examine data for irregularities.
Method: getFiles "No files found in directory <dirName>"	The directory listed is empty of .fits or .fts files.	Check to make sure the directory is correct and the file path to the directory is correct. Make sure the directory has valid .fits or .fts files in it.
Method: matchCal "No valid calibration files passed into matchCal"	The Photometry array representing calibration files passed into matchCal is empty.	Make sure there are calibration files in the calibration folders passed into other methods.

Method: matchCal "Error in matching calibration files: Invalid .fits file"	The main file to be matched with a calibration file cannot be opened.	Check to see if the filename is correct, and the filepath is accurate. Check and see if the file is a valid .fits or .fts file.
Method: runFiles "No <dark frame, bias frame, flat field> calibration files found" "Aborting run files due to error finding calibration files"	The directories containing calibration files are empty of .fits/.fts files.	Check to make sure the directory is correct and the file path to the directory is correct. Make sure the directory has valid .fits or .fts files in it.

Some Possible Data Problems

In addition to errors within the program itself, not all data is created equal. Some data can have problems outside of the program that can make it difficult to perform differential photometry autonomously. While this is by no means a comprehensive collection, here are some of the possible data problems that may be encountered while using the program, how to spot them, and what they may look like practically.

In Figure 1 there is a light curve of the star BG CMi. It may initially look like there are no error bars included in the graph, however they are there. The error bars are so small compared to the scale of the graph that they appear as normal dots. When it comes to a light curve with a large spread of points with very low error, the problem may be that there is another star too close to the target star. In this image, a larger star bordered on the edge of BG CMi's radius, dipping in and out of its area for calculation. This caused wildly different photon counts in the area, making the magnitude jump around as seen in Figure 1. A solution would be to manually make the calculation radius smaller, if the stars do not directly overlap.

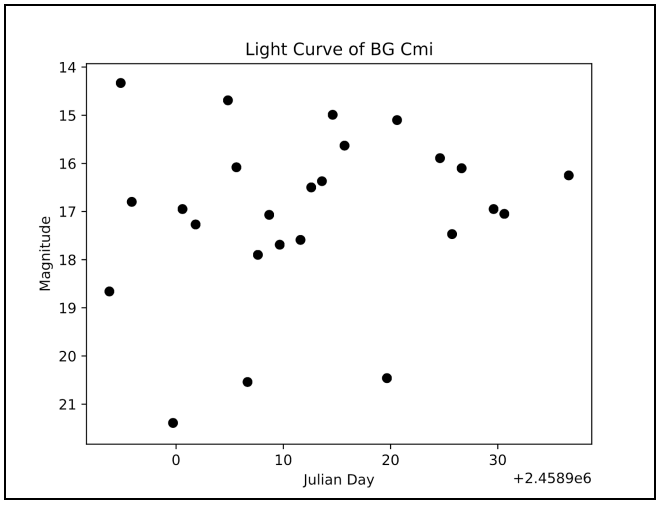

Fig. 1: Light curve of BG CMi demonstrating the irregularities in light curves caused by other stars being in the calculation radius of the target star.

In Figure 2 there is a light curve of the star DW Cnc. In the image the light curve looks mostly normal, however it is filled with unusually large error bars. This is likely caused by airmass interference, as the images of DW Cnc were taken very close to the horizon. Another possibility that could cause high errors is not using enough reference stars, adding more reference stars would allow for inaccuracies to be smoothed out.

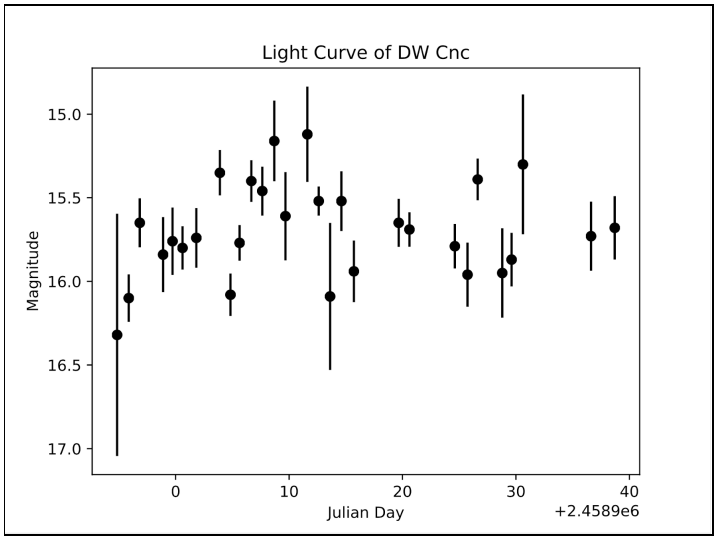

Fig. 2: Light curve of DW Cnc demonstrating a light curve with higher error than is expected caused by either high airmass or low reference star counts.

Figure 3 is a light curve of V2306 Cyg. While only 4 items were plotted on the graph, the program used roughly 20-30 images to create this image. Most of those images were discarded for magnitude calculation problems. The problem with this image is that the image was in a

crowded field, making it difficult to find and use reference stars not bordered by another reference star. Most of the images were discarded and those left are highly inaccurate representations (the images in question could not perceive stars dimmer than 16th magnitude, so the calculation of a 20th magnitude star is not possible). The solution to this could be to pick out reference stars by hand, carefully, where no other stars are present and to manually set a smaller radius to avoid catching other stars in it.

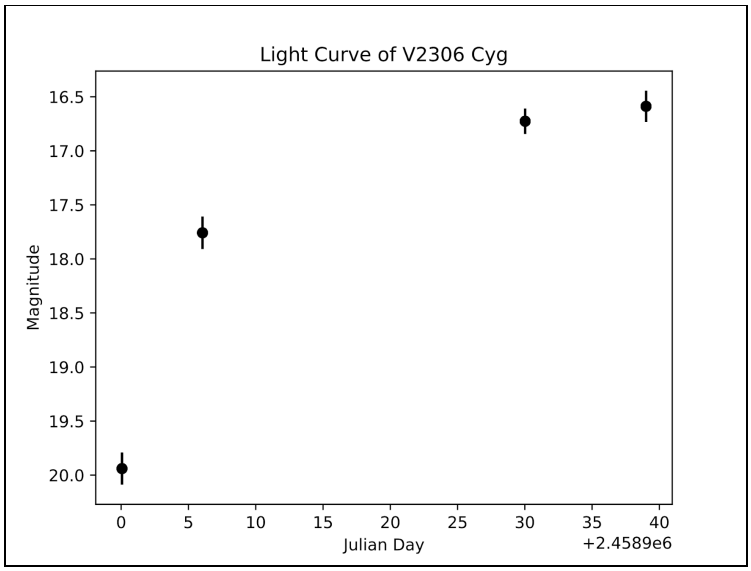

Fig. 3: Light curve of V2306 Cyg demonstrating crowded field photometry problems that may occur.

Input and Output Formatting

There are two kinds of output and input .csv files used in Photometry+, files for Photometry object results and files for Star objects. The formatting for each is as follows:

Photometry Results File:

File Name,JD,Magnitude,Error,
filename.fts,2458894.8388310187,15.65,0.379867,
...
X,Reduced Chi Square: 1.20,

Stars File:

ID,Right Ascension (Decimal Degrees),Declination (Decimal Degrees),Radius (pixels),Photons,Magnitude,Calculated Target Magnitude,Error,
Ref1,119.704825,16.117481,25,43099.0,16.22,0.0,523.0,
...

Glossary

Term	Meaning
Differential Photometry	Photometry is the process of measuring the light coming from a celestial object, differential photometry determines an objective magnitude by comparing that light to stars of known magnitude in the vicinity.
Calibration files	Images that come from a telescope are not perfect, background radiation and noise need to be removed from it. This is done by subtracting calibration files, named dark frames and bias frames, and dividing by a normalized flat field.
Charge coupled device (CCD)	A CCD is an array of pixels accepting light from where it is pointed at, counting the amount of times it is hit by a photon.
Color Magnitude Diagram (CMD)	A diagram plotting stars by their brightness and color that can be used to determine the age of a globular cluster.
Counts	The amount of counts caught in a CCD pixel, this is representative of but not exactly equal to the photons hitting that pixel from that part of the sky.
.fits file	Images from a telescope that are attached to WCS data are of the type .fits or .fts.
Full-width half-maximum (FWHM)	In a Gaussian distribution, the full-width half-maximum represents the distance between the two sides of the curve at half of the maximum value of the Gaussian. Stars images are close enough to a Gaussian curve that this is used as an effective radius for imaged stars.
Light Curve	Light curves are plots of the brightness of a star over time which is used to see the way variable stars move, dim, brighten, and change over time.
Magnitude	Magnitudes are the measure of brightness of a star and work on an inverse scale, where larger numbers indicate dimmer stars.

Objects	Objects are custom data types created in a program to store certain information. Then new instances of the object can be created like any other variable.
χ^2_r (reduced chi squared)	The χ^2_r is a measurement of if a star is displaying variability or not. A number larger than three indicates the star is changing over that period of time and a number less than one indicates error in measurement.
World coordinate system (WCS)	The WCS is a set of parameters attached to a telescope image that denote what area of the sky the image is of.