

Step-by-Step Guide to Astrometric Fitting and Aperture Photometry in IRAF

Patrick Edwards, Ricky Patterson, & Mark Whittle

May 10, 2018

1 Introduction

Throughout the Summer and Fall of 2017, I worked with Prof. Mark Whittle and Dr. Ricky Patterson reducing LBT data in the optical and near-IR bands (g, r, J, and Ks). The work that was done was astrometric fitting - matching star positions on the image to accurate wcs coordinates - and aperture photometry - measuring the flux of standard stars in order to determine an ADU-to-magnitude conversion for each image.

The two processes are fairly straight forward, with photometry being the most time consuming. When the photometry was completed over the summer, many of the fields already had astrometry completed. However with the release of the more accurate Gaia catalog, it was in our best interest to do the astrometry again, with better positions on the stars. While I completed the photometry first, and then went back and updated the astrometry, with new images it is necessary to do the astrometry first. Therefore I will describe the process of astrometric fitting, and then aperture photometry.

The tasks involved with astrometry are as follows:

1. Obtain a star catalog of the field from the Gaia dr1
2. In the package IMAGES.IMCOORDS, run **starfind** on the image
3. Run **ccxymatch** on the image
4. Run **ccmap** on the image
5. Load the package *mscred*, and run **mscimage** on the image

The tasks involved with photometry are:

1. Generate a file consisting of the standard stars in the field.
2. In the package NOAO.DIGIPHOT.APPHOT, run **phot** on image
3. Generate files necessary for running **fitparams**
4. Run **fitparams**

2 Preparing Images

Before any work can be performed on the images, it is first necessary to make sure that they are trimmed down to a workable size. The images in the J and Ks bands are already trimmed down to square images, and thus require no further work. The g and r images, however, require some trimming to get them down to a good size. Both the g and r band images, after being scamped and swarped, have very ill-defined edges, from being stacked on one another. It is therefore necessary to use the **imcopy** task to trim these down to a more proper size. After this is done, then astrometry can be performed.

3 Process of Astrometric Fitting

The purpose of astrometry is to match up known stars in the field with their proper wcs coordinates. Before any astrometry is done, there is a good chance that any catalogs loaded on the image will not match up with the positions of the stars. Therefore it is necessary to match the exact image centers of the stars with their coordinates.

3.1 Obtaining a Catalog

The catalogs were obtained from the Gaia catalogs. One just needs to do a quick Google search for "gaia dr1," or go to the site: <https://gea.esac.esa.int/archive/> and from there can access the search screen. I conducted searched in a 5 arcminute radius around the object, and chose to return the data columns: ra, dec, and phot_g_mean_mag. The results of the search must be saved into the file format of "VOTable (plain)", as opposed to the default option of "VOTable". This should be then saved to the directory in which you are performing the astrometry.

This catalog should be loaded into DS9 on the field, and edited so that it does not contain stars which are outside of the images, and does not contain stars which are saturated. It can then be output as a .tsv file, so it is in the file format needed for later tasks. Because the catalog needs to be edited based on the size of the image and the saturation levels, it may be necessary to make a separate .tsv catalog for each separate image.

It is very important that the catalog contains stars from all around the image, and not just from one half or section, in order to give the astrometry fitting tasks enough data points to get a good fit.

3.2 Starfind

There should now be a directory with the .vot file of the gaia catalog, and the images of the field being worked on (g, r, J, K). The next step is to run **starfind** on an image. This task finds the pixel coordinates of the center of stars in the image and outputs them to a text file. This task is fairly simple, however it normally has to be run multiple times before it has gathered all of the data necessary.

3.2.1 Prepping for starfind

Firstly, you need to open up the image which is being worked on in a DS9 window, and proceed to upload the Gaia catalog. This gives allows us to see which stars we definitely need to record with the **starfind** task. In most cases, while the circles in DS9 which are meant to mark the stars are not in the right position, the offset is small enough that one can easily recognize the pattern.

In the **starfind** task, the only parameters which need to be adjusted are:

- *image*: name of the input .fits file
- *output*: name of the desired output file. The convention I used was "####G.obj.gaia", where the "####" represents the number of the field (e.g. 0342), and the "G" is whatever image is being run
- *hwhmpsf*: half-width half-max of the psf in the image in pixels, as can be found using **imexamine** radial plot function
- *threshold*: the lower threshold in ADU for finding a star. This is determined using **imexamine**, by viewing the radial profiles of the faintest Gaia stars, and determining at which ADU values they peak. Note that this is not sum of the ADUs in a star, but rather the ADU values in the pixels at which they peak.

After running **starfind**, there should now be a file containing the pixel coordinates of all the stars which match the parameters set.

3.2.2 tvmark

The next step is to run **tvmark**. This task overlays the results from **starfind** onto the image in DS9. First, you must use the *display* command to display the image being worked on into a new frame in DS9 (usually frame 2, as frame 1 has the original image and the Gaia catalog uploaded). You can then use the task **tvmark** to overlay the results. It is easiest to run this task straight from the command line in the IRAF terminal. As an example, I will show the commands for this using the field W0342 in the g band:

```
imcoords> display W0342G.fits
frame to be written into (1:16): 2
imcoords> tvmark
Default frame number for display (:16): 2
Input coordinate list: 0342G.obj.gaia
imcoords>
```

If successful this should display the field W0342 in a new frame in DS9, and have red circles over some of the stars. The circles indicate which stars were found by **starfind**. Now in most cases, **starfind** will either find too few or too many stars in the field. What we want is have the lowest number of stars found by **starfind** that covers all of the stars in the Gaia catalog. To get to this point, it is likely that **starfind** will have to run multiple times, with varying values for *threshold*.

When this is completed, we are then prepared to move onto the final few steps.

3.3 ccxymatch and ccmmap

The next step is to run the task **ccxymatch**. This task takes as input the .tsv Gaia catalog and the output of **starfind**, and will match the points between the two. The parameters necessary to run this task are:

- *input*: the output file from **starfind** - e.g. 0342G.obj.gaia
- *referenc*: the Gaia .tsv file - e.g. W0342G.tsv
- *output*: the desired output file - e.g. 0342G.ccxy.gaia
- *tolerance*: set to 1
- *ptoleran*: set to 3
- *xmag*: the X axis scale in arcseconds / pix
- *ymag*: the Y axis scale in arcseconds / pix
- *lngcolu*: the ra column number in the Gaia file (normally 1)
- *latcolu*: the dec column number in the Gaia file (normally 2)
- *xcolumn*: the x pixel column number in the **starfind** output (normally 1)
- *ycolumn*: the y pixel column number in the **starfind** output (normally 2)
- *lngunit*: set to "degrees"
- *latunit*: set to "degrees"

Once these values are set, simply run the task, and it should be output that "### reference coordinates matched". If this number is zero, then this task was run incorrectly. The most likely situation is that there are far too many pixels coordinates from **starfind** compared to the number of coordinates in the catalog. If these numbers are comparable, though, then there should be no issues.

The output of **ccxymatch** is a file which contains a line for each one of the stars which were

matched, and has the ra/dec and x/y coordinates for each matched object.

After successfully running this **ccxymatch**, we are now ready to pump this through into the actual mapping task, **ccmap**. This task takes the output of **ccxymatch** and performs a best fit. The parameters involved in **ccmap** are:

- *input*: the output file from **ccxymatch** - e.g. 0342G.ccxy.gaia
- *database*: the desired output database name - e.g. 0342G.ccmap.db
- *solutions*: set to "alpha"
- *images*: the input image on which the fit is being performed - e.g. W0342G.fits
- *xcolumn/ycolumn/lngcolumn/latcolumn*: the column numbers for the x, y, ra, and dec coordinates. Should be set to 3/4/1/2, respectively
- *lngunits*: set to "degrees"
- *latunits*: set to "degrees"
- *projection*: sky projection geometry - set to "tnx"
- *update*: MUST be set to "yes". This makes sure that the task updates the wcs coordinates in the header file of the image.
- *interac*: set to "yes". Allows an interactive fit in an irafterm

In addition, all the fitting order parameters (e.g. *xxorder*) should be set initially to "2". We can then run **ccmap**, and an IRAF term will pop up with a map of where the mapping stars are on the image. By using the keys *x*, *y*, *r*, *s* in the irafterm, you are able to see the plots of residual v. pixel position. If the order of the fit is correct, each residual plot should look like a random distribution around the center, and the standard deviations should be low. However in many cases there will be a discernible pattern in the residual plots (a parabola, sine curve, etc.), in which case you must change the order of the fit to something higher. This is done by typing the task ":order 3", then replotting the fit by hitting *f*. If the fit order is still too low, this must be done once more. There should not be a fit of a higher order than 4.

After this is completed, the images should now have updated wcs coordinates, and are ready for aperture photometry.

4 Process of Photometry

4.1 List of Standard Stars - DS9

The first step provided in the listing in Section 1 is to make a list of standard stars. This task differs between the J/Ks images and the g/r images. Each task is described in brief below. The description of these tasks are done using the J/K images as examples, but the process extends to the g and r images as well.

4.1.1 J,Ks Images

First open up the chosen image in DS9. For our purposes there are two images, one field taken in the J-band, and one in the K-band. Either one of these images is fine as they will have the same stars and coordinate systems. Overlay the 2MASS infrared catalog to this field, and the stars in the catalog should line up on the center of the stars in the field. Note: not all stars will be catalogued.

4.1.2 g,r Images

For these images we use a catalog from the SDSS survey. To obtain a catalog in, simply navigate to the site <http://skyserver.sdss.org/dr14/en/tools/search/rect.aspx> and fill out either the radial or rectangular search form. Then save the output from the search as a .tsv or .vot file in the proper directory.

The next step is to remove all the stars which are undesirable to us. This includes stars which are too near the edges on the field (which will not do us very good in the later sections), and stars which have undefined magnitude errors. If the stars have an undefined error in only one of our two bands, they are kept. The way which we found to get rid of these bad stars is by saving the catalog as a .tsv file, and then manually picking and deleting the stars from this file. *Note: It is much easier to identify the stars on the edge if the catalog list is first sorted by either RA or dec.* Additionally, an new column, titled *NUM* was added to the .tsv file, simply numbering each of the chosen objects. The stars which have a usable J and K magnitude are the only ones which were used.

In order to run the **phot** task, you need to have a file containing the coordinates of the chosen stars, given in the form: x y. We do this by running the **ccfind** task, located in the IMAGE.IMCOORDS. package. Using the catalog downloaded from DS9, and specifying the columns in which the RA and dec are located in this file, one can run this task and have the pixel coordinates appended to the end of the file. Using some crafty rectangle-kill commands in emacs, you can delete all the information before the pixel coordinates, and save new files with specifically the xy positions of each object. *Ctrl-SPACE sets the mark at the upper left of the area you want to kill, move marker with keys to bottom right of rectangle, then Ctrl-x (then let Ctrl go) r k finishes the command.*

4.2 Prepping for phot

The **phot** task requires a lot of specific parameters, which must be manually entered. These commands lie in the parameter file for **phot**, as well as in psets of **phot**: *datapars*, *centerpars*, *fitskypars*, *photpars*. These parameters must be found either in the header for the image file, or by examining the image using the command **imexamine**.

4.2.1 datapars

A few details which need to be fixed in this pset:

- *scale*: Defined the base units of your image (for us arcsec / pixel)
- *fwhmpsf*: FWHM of the PSF in scale units (arcsecs are the scale units, as set above)
- *sigma*: Stdev of background counts; found using **imexamine** with the **m** tool
- *datamin*: make sure to set a low enough datamin for each image. This and sigma are image dependent

In addition, there are the parameters which are pulled from the image header, and those must be updated to make sure that they are accurate.

4.2.2 centerpars

In this pset, just make sure that the centering algorithm is on "centroid", and that the cbox is set to 5.

4.2.3 fitskypars

This pset deals with the background sky values, and thus sets the background sky annulus. In this section, you must just check that the parameter *annulus* is set to a reasonable value (for 0342 it is 4. arcsec), and the parameter *dannulus*, which is the width of the sky annulus, is set to a reasonable value.

4.2.4 photpars

In this section, the only two parameters needing to be changed are *apertures* and *zmag*. By default, *zmag* should be set to 25. This is fine. The apertures are chosen to get a wide variety of mag values. The ideal aperture is $3.5 * FWHM$, so choose a few around this value. In the 0342 case, they were chosen as [1.0,3.5] in steps of 0.5, which did the job fine.

4.3 Let's Run phot

This is a fairly easy step. After fixing the parameters, simply enter the name of the image, the coordinate file associated with the image, and turn interactive mode to "no". *Note: when entering the image, you only need to specify its name, i.e. W0342J or W0342K, and not say it is a .fits image.* Then run **phot**, and you should be gifted with a file, titled "W####[J/K].mag.1", containing all the information from the task. This is the name of the object, details of the observation (UT, airmass, etc.), the information on each aperture, and the magnitudes and errors. *Note #2: Pay attention to how the magnitudes are calculated by the **phot** task. This is important in understanding the mag values.*

After running **phot**, we now have all that we need to go on to the next step of photometry, which is running the task **fitparams**...

4.4 Prep for fitparams

The **fitparams** task resides in NOAO.DIGIPHOT.PHOTCAL, and requires a few files before it can be run. As input, we need to have these files completed:

- A file containing the 2MASS magnitudes and errors
- A file containing all the calculated magnitudes from **phot**, and other information (####standobs)
- A file containing the configuration of these two files, and the equation they will be fit to (####.cfg)

You will notice that the 2MASS catalog we have already has the J- and K-band magnitudes and errors that we want. Therefore we can simply use this as our standard catalog. However before this file is usable, it must have object names in the first column (obj-#), and it must be separated by spaces, not tabs. A copy of this file can be made, and the tabs be converted to spaces by selecting all and using the command "Esc-m tabify". We now have a usable catalog file.

This file also needs a configuration file. Using the task **mkcatalog** would have formatted one automatically, but since we did not do this, we must make our own. For reference, you can simply look in the W0342/ directory at the f0342standcat.dat file to see what it should be like.

We now need to use **mnobsfile** to generate the observation file. This is a fairly simple task, but can get confusing if the wrong files are used as parameters. The necessary parameters are:

- photfiles = "*mag*" ——— gets the data from the output of **phot**
- idfilters = "J,K" ——— tells which filter names there are
- imsets = "0342standstars" — the image naming file which needs to be made
- observations = output ——— set this to whatever you want the task to output to
- obscolumns = "2 3 4 5" ——— defines the columns in the mag files which it will grab info from
- aperture = "#" ——— defines which aperture in the mag files you actually want to use. This is the aperture at which it seems that the magnitudes are beginning to level off.

The only file which needs to be created for this task is "####standstars". We do this by simply writing a new file which simply says "obj : W####J W####K". Simple as that. It is necessary that the beginning for standstars is the same as the naming convention in the 2MASS catalog (i.e. standstars must be obj, and name in 2MASS must be obj-#). **Make sure that they are in the same order that they are listed in this task's idfilter parameter.** Then simply running **mknobsfile** will generate the observation file and the configuration file necessary.

The last step before running **fitparams** is generating the final configuration file. This contains the configuration information for the standard catalog, the obs file, and also information on the fit transformations. In an editor, make a file containing the transformation information, the format of which can be found in the IRAF documentation, or by looking in the photometry/jk/W0342/ directory. Then run the task **mkconfig** with these configuration files as input, and the output will be the final configuration file for **fitparams**.

4.4.1 Specifics for Configuration File

The configuration file essentially just tells **fitparams** which columns in the input files hold the listed values. There catalog and observations portions are obvious enough, but the transformation section can be used to our advantage by telling **fitparams** exactly what we want to plot, which variables to keep constant or to be fit, among other things. We want to add portions to our transformation which tell the task what to plot (usually a "left half of EQ" vs "right half"), as well as examining if there is a notable color term.

- **Defining plots:** We can choose what we want to see in the plot for each fit. This is done by using the command:
`plot(JFIT) = Jmag+j2,mJ & plot(KFIT) = Kmag+k2,mK`
Put these beneath the description of each respective fit (i.e. line after "JFIT ... ").
- **Finding Color Terms:** The color term tells if there is a strong dependence on the magnitude of a star depending on which band it is viewed in. Use the commands:
`set JK = Jmag - Kmag`
`set jk = mJ - mK`
at the beginning of the transformation section, and then add the term " + JK * j3" to the end of the JFIT equation, and a k-version to the end of the KFIT. You can again use the plotting controls above to view these and see if it actually matters. *Note: it does not appear as though the color term will be a large factor in any of these calculations, and thus the j3 and k3 parameters can be set = 0.0. However it is good practice to check this and keep the color term in the fit equations.* After these things are done, you should be ready to run **fitparams**.

4.5 Running fitparams

The input parameters for **fitparams** are the observation file, the catalog file, the configuration file, and the name of the file which the computed data will be written to. In addition, glance over the other parameters, all of which are fairly self explanatory. If you run the task once and the fit did not converge, you may consider changing the maxiter parameter to a higher number, so long as it is not ridiculously high. once run, an interactive irafterm will open up. Using the colon command **:vshow** you can see the details of the fit, such as the reduced χ^2 value, the fit values and errors, and other useful information can be used. *Save this file.* You can view the fit and see if there are any stars which are drastically off of the fit line. These will typically be the fainter stars, which you can go and comment out of the observations file, assuming there are enough brighter stars to have an accurate fit.

5 Re-binning Images

After these images have been fit astrometrically and photometry has been performed, these need to be re-binned to each have the same pixel size. The original images have pixel scales of about 0.12 "/pixel and 0.22 "/pixel for the J/K and g/r images, respectively. This is done in the task **mscimage**. We want each of the images to be re-scaled to the size of 0.1 "/pixel. This way, we

can stack the images to make proper RGB images of each field, in order to look at the colors of the galaxies. To do this, first load the package MSCRED. The parameters for this task are:

- *input*: input image
- *output*: desired output image name
- *wscsource*: set to "parameters"
- *ra*: ra of the galaxy
- *dec*: dec of the galaxy
- *scale*: new pixel scale to be re-binned to – set to 0.1
- *fluxcon*: MUST BE SET TO "yes". This conserves the ADU/area between the images. In essence a circular aperture with a 2" radius in the un-binned and re-binned image will give the same number of ADU in the SUM
- *interac*: set to "yes". Allows for interactive fitting
- *fitgeom*: set to "general"
- *??order*: the order of each fit (xx, xy, yy, yx), each should be initially set to "2"

Checking the order of the fit is the exact same process as in the **ccmap** process. Each of the residual plots should show a randomly distributed graph, with no discernible pattern. When the fit is completed to that point, each image should now be astrometrically fit, have had photometry performed to find the zeropoint offset of each field, and have been re-binned to a pixel scale of 0.1"/pix. From here, one now has the images completely reduced to the point where data can be gathered on each individual frame.