

Using flamstead, mapccd and starpin.

Once you have a summed, deep image, there are two ways to proceed. If you are only interested in objects with a relatively high signal-to-noise in each frame you only need to do the following.

- Run mapccd, putting the summed image in first, to find approximate offsets between the frames.
- Run flamstead on the summed image, and an appropriate short exposure to get the final list of stars.
- Run opphot (with inputs psf.pos, stars.pos, offsets.pos), letting it centroid to find the star positions.

But, if you want to pick up low signal-to-noise objects in each frame, and then sum them to get a good signal-to-noise in the final catalogue things are a little more complicated.

- Run mapccd, putting the summed image in first, to find approximate offsets between the frames.
- Run flamstead on the summed image, to find stars for starpin. Since you only need a relatively small number, a high sigma can be used.
- Run starpin, to find accurate offsets between the images.
- Run flamstead on the summed image, and an appropriate short exposure to get the final list of stars.
- Run opphot (with inputs psf.pos, stars.pos, offsets.pin) with the star positions fixed.

I guess if we think about it hard we could use the catalogue from mapccd to find the stars that starpin will fit, although no removal of non-stellar objects will have been done on it. But it would remove one stage.

The combination of positions from different images in cluster is done by cluster creating its own six co-efficient transformation, so choosing the easy option from above does not degrade your final astrometry.

Running Opphot

You will need to decide the values of χ^2_ν and skewness of the sky fit you are willing to accept. After running cluster, you can use the .sky files to make a plot using the sky macro, and then decide on a cut before re-running opphot.

Running Cluster

Create a log file

The first thing to do is create a log file. This tells cluster various things it needs to know about each frame to manage to process it. The file is called something like headers.log Working in the simplest way (option 2), it takes this information from the header of the ARK file, and from the .opt file. If the items are not in the header, make sure you edit the resulting .log file, and correct the exposure time, airmass and filter (the date and time are not used). You will also be prompted for whether or not the image was taken under photometric conditions.

Aperture correction

Now you should select stars which could be used for aperture corrections. Two groups of stars are selected. The first group cover a central box of the CCD (which can be used to create a median stacked profile), and the second set are scattered over the entire CCD. If you think the PSF is a function of position you can restrict the central box, since the fraction of the X and Y dimensions to be used is prompted for. The program then reads the .opt file, and then selects the 80 brightest, unsaturated stars in the region. In addition it will search for a file with the same name as the .opt file but with a .bpl extension. If any of the bad pixels in this fall within a star's aperture, it will also be rejected from the list. It writes a results file for each image, with the extension .mdin, which should be a copy of the .opt file for the stars it has chosen. If you gave the X and Y fractions as less than 1, the code assumes you want a PSF which varies with position, to calculate this it selects the 80 brightest, unsaturated, un-bad-pixelled stars over the entire CCD. These stars are written to a file called .varin.

You can now run the aperture correction option. First you are prompted for the standard star aperture size, and then for the order of the polynomials which you want to use to represent the varying aperture corrections (use 1,1 for unvarying PSF). This stage normally prints lots of information to the screen, as aperture correction is probably the most critical stage of the data reduction and so must be watched carefully. Once its all over, look at the .mdlog, and if you have a varying PSF the .varlog files.

Trouble Shooting The most useful diagnostics in the .mdlog file are the list of corrections using different stars in the center, and the FWsHM of the stars used. The aperture correction should not change much, whichever star is placed at the center, although the list is in order of increasing brightness, so those nearer the top of the list may show a larger scatter due to noise. The bottom few should be consistent, and their scatter gives you some idea of the true error in the correction. Several stars are normally rejected because their FWHM is discrepant. Check the mean for the final list is close to that for the PSF star, and that the scatter is at most a few percent.

If you have a spatially varying PSF you'll get a .apcors file, which you can plot out using graph to give you the aperture correction as a function of X and Y or flux using graph (there are macros varx and vary and varf which will plot this file). After the fitting is done. the .varlog file has the residuals in a similar format (also plottable using varx and vary), but also has the RMS of the fit in the header. Finally, if you have a variable PSF, comparing the values of the profile correction at the centre of the field given by the median stacking method (at the tail of the .mdlog file) with that for the fitting method (at the head of the .varlog file) can be revealing, as they should agree to within a few hundreths.

The photometric co-efficients

The next file that you'll need is one with the photometric co-efficients in, called "coeffs". The file should begin with a line that can be used for comments. The next line should contain the number of filters you are using, which must be the same as the number of colours you are creating. The line after that is the names of the colours you want to create, the best example being

V B-V V-I U-B

Thereafter, each line in the file should begin with the field number the co-efficients apply for, followed by (in this example)

$$K_V \ \Psi_V \ Z_V \ K_{B-V} \ \Psi_{B-V} \ Z_{B-V} \ K_{V-I} \ \Psi_{V-I} \ Z_{V-I} \ K_{U-B} \ \Psi_{U-B} \ Z_{U-B}.$$

The equations being used are

$$B - V = \Psi_{B-V}(b - v) + Z_{B-V} - K_{B-V}X,$$

$$V - I = \Psi_{V-I}(v - i) + Z_{V-I} - K_{V-I}X,$$

$$U - B = \Psi_{U-B}(u - b) + Z_{U-B} - K_{U-B}X,$$

$$V - v = \Psi_V(B - V) + Z_V - K_VX.$$

Note that you are not free to make up any colours you like, you must make up your colours by substituting different filter names for those above. For example, if you are doing *RIZ* photometry, you could substitute *I* for *V*, *R* for *B* and *Z* for *I*, and nothing for *U*. Then your header would read

3
I R-I I-Z

and the corresponding co-efficient lines are

$$K_I \ \Psi_I \ Z_I \ K_{R-I} \ \Psi_{R-I} \ Z_{R-I} \ K_{I-Z} \ \Psi_{I-R} \ Z_{I-Z}.$$

There is one other allowed combination of filters, which is the following.

$$V \quad V - I \quad V - B \quad R - I$$

The equations used are then slightly different

$$V - B = \Psi_{B-V}(b - v) + Z_{B-V} - K_{B-V}X,$$

$$V - I = \Psi_{V-I}(v - i) + Z_{V-I} - K_{V-I}X,$$

$$R - I = \Psi_{R-I}(r - i) + Z_{R-I} - K_{R-I}X,$$

$$V - v = \Psi_V(V - I) + Z_V - K_VX.$$

These definitions mean that *K* should always greater than zero, and Ψ should be around 1 except for Ψ_V in the *UBVI* system or Ψ_I in the *RIZ* which should be close to zero. If the co-efficients have come out of Tim's calib program, note that you must add one to Ψ for the equations for the colours, but not for the magnitudes (*e.g.* for Ψ_{B-V} but not in Ψ_V).

The photometry option

Now you can run the photometry option (number 7). The first diagnostic you should look at here is in photometry.log. Here it will tell you the "transparency correction", which is actually the final correction which must be applied to make the magnitudes in the frames agree. It should be small. If it isn't, the next place to look is the .trans files, which give magnitude in one frame against that in the other. These files can be plotted using qdp.

The final output from this section is a file called photometry.cat. From this point on all the files with the stars in share the same format. There are three lines of header, followed by one line per star. The first thing on the first line of header should be the number of colours in the file. The second line of header should be the names of those colours. Each star line has first the field number, then the star

number, next the RA and declination (set to zero at this point as they haven't been calculated), the X and Y position on the CCD. There then follows a group of three numbers giving the magnitude, error and flag for the star, followed by three more groups for each of the colours.

The flag is a two digit number, the first number being the flag for the blue filter involved in the colour, the second the red filter. There are two flags for the first colour (*e.g.* V) since you need to know a colour (*e.g.* $B - V$) to calculate it.

During the process of combining files, cluster will examine the χ^2 which it obtains, assuming the stars are constant. These are given in files called chisq.X where X is the band name. It may also flag stars which have a high χ^2 , but details will be given in photometry.log.

Selecting the local astrometric standards

Having got the photometry out, the next stage is the astrometry. You'll need a centers.dat, which looks like

```

ASTR
Comment 2
Comment 3
1 05 40 11 -02 32 09 22.75 11.38

```

The first line is the astrometric distortion for the telescope, as used by ASTROM. See the ASTROM documentation for details, but ASTR works well for most single CCD instruments, and GENE 220 should be used for the INT prime focus. The second two lines are comments, followed by one line per field. The first number is the field number, and then the RA and Dec of the field center. If you have a small, negative declination then the degrees may be zero, but even so you must say whether they are "-00" or "+00". The last two numbers are the full width and height of the CCD in arcminutes; if these are too generous the program will simply run a little slow, too small and you'll lose calibration stars. For cameras with many CCDs, make sure the RA and Dec are the center of telescope field, not the CCD, and that the width and height are set to the size covered by the CCDs.

Using this you can now select stars from the USNO catalogue to use as reference stars using option 8. You'll be prompted for which CD to put in the drive, or if you have the USNO-A2 on disk, which directory its in. This will create a cluster format file called f???.ast which has the stars to be used for the astrometric calibration. Of course if you want, you can skip option 8, and create your own .ast file from somewhere else (*e.g.* the 2mass catalogue). If cluster fails to find a file called f???.ast (or the old f???.stars) it will prompt you for the file name.

Supercosmos data can be used for the astrometry. Download the appropriate section of the catalogue from the SSS page in ASCII format. If you set the epoch to correspond to that for your data, then the proper motions will be used, but this is probably a bad idea as most of the proper motions are bigger than the associated errors. If your field has bright stars in it, you may wish to accept other stars close to the bright one. To do this you should state in the expert parameters section that you are willing to accept flags of up to 2047, thus accepting data with the 10th bit and lower set. The program supercos will then translate this ascii file into a cluster catalogue, throwing out blended stars.

Doing the astrometry

For the astrometry itself, you'll need to go to skyview and get a FITS image of your bit of sky. An "saoimage -fits filename" will display the image, and you should have a wagram window up at the same time with a CCD image of your field in it. The CCD image to choose is the one which photometry.log tells you "Positions measured in frame of image ...". Hitting the "p" key on the SAOimage window will print out the co-ordinates. Choose three bright stars and write them into a file called refstars.cat

This should be a cluster catalogue with the RAs and Decs taken from the astrometric catalogue, and the X and Y positions from the CCD. The field numbers and star ids can be anything you like, as can the colours (but make sure you place the correct number of them as the first thing on the first line of the file).

You will also need a file called `astrometry.info`, which should have as its first line the astrometric distortion as required by `astrom` (e.g. "ASTR"), and as its second line the tangent point (normally the centre of the CCD). This can be given either as pixel co-ordinates or in RA and Dec. If the latter, even if the declination is 00 degrees, the sign should be placed before it (not on the minutes or seconds). If there is more than 1 CCD, then there should be a line in the file for each CCD, giving first the CCD number, and then a 6 co-efficient translation to take each CCD's X and Y into a unified frame. See appendix for how to create this. Finally, by placing `~` in front of either the tangent point or the distortion will make cluster treat it as a free parameter.

This procedure will produce a file call `astrometry.cat`, which is in the same format as `photometry.cat`. In addition there are two diagnostic files; `astrometry.log`, which has the final rms of the fit and `astrometry.dat` which has the positions and residuals for each star in the fit; it can be plotted using the graph function `ast`.

Creating the final catalogues

Option 10 will copy the results from each field into a file in the directory above called `xxxx_raw.cat` (`xxxx` is your choice).

You now have to choose option 11, which gives you a new menu. The aim of this menu is first to examine the overlap regions between images and move the zero points to get the smallest possible dispersion between them. The second aim is to remove duplicates.

First find the stars in the overlap regions which can be used to calculate the differences between the zero points. To do this use option 12, with a relatively small (say 1 arcsecond) correlation radius. It creates a file called `xxxx.repeats` which has the overlap stars in. Option 13 creates a file called `colldata.dat` (via two temporary files call `data.dat` and `sortdata.dat`) which has the magnitude differences between all the overlap stars in a selected colour, with those that contribute a χ^2 of 4 or more removed. From this it creates a set of mean magnitudes for each overlap called `coll.means` etc. The sense in which this mean is given is that it is the first field minus the second field given on that line. Option 16 which calculates the shifts that must be applied to each field (`collfitcons.dat`), to minimise the differences. The format is simply lines on which is a field number and a correction to be subtracted off the data. These are applied to `xxxx_raw.cat` by option 25, which results in `xxxx_norm.cat`.

Finally option 26 sorts the catalogue by RA and then averages the results for stars which appear in more than one field to give `xxxx.cat`. The problem is that it also averages all stars (even if they appear in the same field) within the correlation radius you give. This ensures that no star appears twice, but set the radius too large and stars will disappear. So the search radius should be large compared with your astrometric accuracy (say 3 times it), but not much bigger than the fwhm seeing disc, otherwise you start losing real objects. The field numbers and co-ordinates in `xxxx.cat` where several stars magnitudes and colours have been averaged, refer to the frame from which the co-ordinates were taken. There are two diagnostics to make sure you get the radius used correct. The best is `remove.log`, which gives you a plot of magnitude difference against separation for each pair. If you do a run with a radius of two or three times the seeing the plot normally divides into two regions. One with small radius and small differences are the real pairings, whilst bigger separations and bigger differences are spurious pairings. Choose a radius somewhere between these groups. The other diagnostic is `marged.cat`, a catalogue of all the objects which result from mergings.

Data flagging in catalogues

The flags are divided into string and weak. A strong flag will always be written, overwriting any other flags, whilst a weak flag will only be written if the measurement is unflagged so far. Thus the order the flags are applied is important. If possible flags are nominated to be strong, as this makes them simple to implement. A weak flag is only used if there might be reason to trust the resulting photometry. The ill-determined sky criterion is very cautious, so if inspection of the image suggests the sky is not strongly structured, such data can be useful. Similarly patching bad pixels can work very well, at which point images on bad pixels may be worthwhile. Finally, data with negative counts are normally good data, but the magnitudes are encoded in a particular way. Cluster will always ignore flagged data, with the exception of negative counts, when combining several measurements of the same star. The list of flags, in the order they are applied, is as follows.

Flag Number	Flag Name	Program	Algorithm	Type
1	Non-stellar	flamsteed	Star shape estimator.	Strong
2	Too close to edge	opphot	Position fitting.	Strong
3	Sky fit failed	opphot	Sky determination.	Strong
5	Ill-determined sky	opphot	Sky determination.	Weak
4	Saturated	opphot	Flux measurement.	Strong
2	Too close to edge	opphot	Flux measurement.	Strong
8	Bad pixel	opphot	Bad pixel flagging.	Strong
7	No input data	cluster	Data input.	Strong
9	Negative counts	cluster	Data input.	Weak
6	Variable	cluster	Combining images.	Strong

Cluster converts the counts measured by opphot into magnitudes, and this could lead to problems when dealing with faint objects, especially ones which have negative counts in some images. Stars with negative counts are flagged as such, and the magnitude replaced by the flux error converted into a magnitude. Thus magnitudes for objects flagged 9 should be treated as bright limits. Cluster still keeps track of the counts, so that when several measurements of the same object are averaged, they give the correct answer. It also means that colours which are quoted for these objects are limits. For example a flag of 90 in the BV colour, means that the B magnitude is a lower limit, so B-V is a lower limit.

Recommended Checks

The following is a minimalist set of checks, which should turn up a fair fraction of problems.

```
grep FWsHM */*.opt | sort -g -r -k8 | more
```

From the directory which contains the sub-directories for each field, a

```
grep RMS */*.varlog | sort -g -r -k7 | more
```

should give you a list of the RMS of the polynomial fit to the profile correction, starting with the worst.

```
grep flagged */photometry.log | sort -k5 -g -r | more
```

```
tail -2 */astrometry.log | grep arcsec | sort -k9 -g | more
```

Error bars in catalogues

The bright end of the error bar for objects with positive fluxes must match smoothly onto the upper limit for objects with negative fluxes. For this to be true it means that as the flux for an object

tends to zero, the result of subtracting the magnitude error from the magnitude (i.e. calculating the bright end of the error bar) must tend towards the value obtained by converting the flux error into a magnitude. This forces our definition of the magnitude error bar (E) to be the bright end of the error bar, subtracted from the magnitude;

$$E = -2.5\log_{10}(f) - (-2.5\log_{10}(f + e)) = 2.5\log_{10}\left(\frac{f + e}{f}\right).$$

Of course, in reality the error bar in magnitude space is not single sided. This definition we have chosen is, of course, the error bar in the bright direction. Rather than quote two-sided errors, we simply note that the faint side error bar can be calculated from the bright-side error bar in the following way. The difference between the bright-side error bar and faint side error bar is given by,

$$2.5\log_{10}\left(\frac{f}{f - e}\right) - 2.5\log_{10}\left(\frac{f + e}{f}\right) = -2.5\log_{10}(1 - (10^{E/2.5} - 1)^2).$$

Appendix. Creating the virtual mosaic

The virtual mosaic is the set of co-efficients which allow you to translate all the stars on the different CCDs into one co-ordinate frame. To create this you need to do the following.

- 1) Make a directory for each CCD
- 2) Make an astrometry.info file which looks like this.

GENE 220

22 54 59.4 62 36 22

1 1.0 0.0 0.0 1.0 0.0 0.0

2 1.0 0.0 0.0 1.0 0.0 0.0

3 1.0 0.0 0.0 1.0 0.0 0.0

4 1.0 0.0 0.0 1.0 0.0 0.0

- 3) Hack down your photometry.cat for each directory to have just the stars for that CCD on it.

- 4) Pick a master CCD into which all co-ordinates are going to be translated, and run where_is and option 9 in that directory. Check solution looks O.K.

- 5) In the directory of another CCD.

- 6) Run where_is.

- 7) Run option 9.

- 8) Now add to astrom.dat the following lines, and run astrom.

0.0 0.0 * 0 0

1000.0 0.0 * 1000 0

0.0 1000.0 * 0 1000

astrom.lis will now contain the RA and dec of these positions.

- 9) Put these into the astrom.dat in the master CCD directory thus,

22 52 20.7223 +62 45 42.814 J2000 * 0 0

22 53 08.5367 +62 45 46.768 J2000 * 1000 0

22 52 20.9073 +62 40 12.576 J2000 * 0 1000

and run astrom.

- 10) The end of astrom.lis will then have the positions of these points in the master CCD co-ordinates.

0 0 +140.447+6279.667 j- 22 52 20.722 +62 45 42.81

1000 0 +128.932+5279.048 j- 22 53 08.537 +62 45 46.77

0 1000 +1140.873+6268.801 j- 22 52 20.907 +62 40 12.58

- 11) Which you can feed into a piece of code like this.

```

real, dimension(6) :: coeff
read(*,*) dummy, dummy, coeff(3), coeff(6)
read(*,*) dummy, dummy, coeff(1), coeff(5)
coeff(1)=(coeff(1)-coeff(3))/1000.0
coeff(5)=(coeff(5)-coeff(6))/1000.0
read(*,*) dummy, dummy, coeff(2), coeff(4)
coeff(2)=(coeff(2)-coeff(3))/1000.0
coeff(4)=(coeff(4)-coeff(6))/1000.0
print*, coeff
end

```

To get the co-efficients that astrometry.info should have.