Chapter 1

Do it yourself

Here we show how photometry is done through a practical example that goes from the beginning to the end. We imagine using MIDAS for our reductions. The logical procdure would be the same if using IRAF, but of course the commands would be different.

The data used come from a calibration run at the Danish 1.54m telescope, using DFOSC.

1.1 Overview of our data

After uploading the tape corresponding to a given night, we will have a directory full FITS images, in this case with names `dfsc????.fits` (or `dfsc????.mt`). In order to start getting an idea of what we have, it is good to extract some useful parameters from the image headers, and write them in a table. However, MIDAS works with images in a format called BDF, so we first must convert from one format to the other. We create a list of the original files, and a list of the new files.

```
ls *.fits > in.cat
ls *.fits | sed s/fits/bdf/ > out.cat
```

Then we start MIDAS with `inmidas`. The conversion is now done with

```
Midas 001> indisk/fits in.cat out.cat
```

After the conversion, FITS files can be deleted. Now we can have a first idea of what the images are, by creating a MIDAS catalog

```
Midas 002> crea/icat src
Midas 003> read/icat src
```

For example, a few lines of the catalog could be

```
dfsc0329.bdf bias  HIB
.................................
dfsc0338.bdf bias  HIB

dfsc0344.bdf ff dome v ESO#451* 20.0 56.90 2148 2102 -120.20 1730 -105.0 FREE Free HIB
..................................
dfsc0380.bdf ff dome b ESO#450* 10.0 56.90 2148 2102 -120.0 1730 -105.0 FREE Free HIB

dfsc0446.bdf pg1633 v ESO#451* 15.0 40.80 2148 2102 -119.80 1730 91.3 FREE Free HIB
dfsc0447.bdf pg1633 b ESO#450* 20.0 40.90 2148 2102 -120.00 1730 91.3 FREE Free HIB
dfsc0448.bdf pg1633 u ESO#632  60.0 41.10 2148 2102 -119.70 1730 91.3 FREE Free HIB
dfsc0449.bdf pg1633 r ESO#452* 15.0 41.30 2148 2102 -120.00 1730 91.3 FREE Free HIB
dfsc0450.bdf pg1633 i ESO#425* 15.0 41.50 2148 2102 -119.90 1730 91.3 FREE Free HIB
dfsc0451.bdf pg1633 v ESO#451* 15.0 41.70 2148 2102 -119.90 1730 91.3 FREE Free HIB
.....................................
dfsc0508.bdf marka i ESO#425* 15.0 35.30 2148 2102 -120.00 1730 91.3 FREE Free HIB

dfsc0621.bdf SKYFF/U ESO#632* 3.0 29.20 2148 2102 -119.90 0 89.6 FREE Free HIB
dfsc0622.bdf SKYFF/U ESO#632* 6.0 29.10 2148 2102 -120.10 0 89.6 FREE Free HIB
...................................
dfsc0773.bdf ff/i ESO#425*  10.0 28.60 2148 2102 -119.80 1730 -105.0 FREE Free HIB
dfsc0774.bdf ff/i ESO#425* 180.0 28.60 2148 2102 -119.70 1730 -105.0 FREE Free HIB
```

Table 1.1: An excerpt from the 'home-made' catalog

```
dfsc0379.bdf ff dome b                                  ^2 2148,2102
dfsc0380.bdf ff dome b                                  ^2 2148,2102
dfsc0446.bdf pg1633 v                                   ^2 2148,2102
dfsc0447.bdf pg1633 b                                   ^2 2148,2102
```

Now this is some useful information, but we just have the identifier of the frame, and the dimension in pixels. The problem is that (a) the identifier could be wrong (at 4am in the morning many things can happen!), and (b) we need more parameters to calibrate our data. Basically, we want to be sure of the filter, the exposure time, and the airmass. This information is contained in the image header, and each piece of information is called *descriptor* in MIDAS. So for example

```
Midas 004> read/descr dfsc0379.bdf zdist
frame:  dfsc0379 (data = UI2)
ZDIST: 56.900000000000
```

will tell us the zenith distance of the object. Of course we want to extract the information for all the images, and save it to a file. Each astronomer can have his own way to do this. Here we do it with a bit of MIDAS, and a bit of Unix. We take advantage of the MIDAS command `execute/catalog`, which allows to repeat an operation on every image of a catalog, and then we will do a `print/descr`, instead of a `read/descr`.

This will print the value of a list of descriptors into a file, and the file must be previously assigned with the `assign/print file` command. The format of the output is not very useful, so Unix is used to produce a final practical table. All these operations are performed with two MIDAS procedures, `all_head.prg`, and `image_head.prg`, which are listed in Appendix A. We do

```
Midas 005> @@ all_head
```

and we obtain `all_head.dat`. A fraction of this file is shown in Table 1.1 (the table has been edited a bit compared to the real one made by the procedure, in order to make it fit the page). Now we see more useful data in the table. From left to right, the name of the frame, the identifier, the filter number, the exposure time, the zenith distance, the frame x and y pixels, the CCD temperature, the camera focus, the rotator angle, the status of the other filter and grism wheels, and the amplifier that is used.

As a first check, we verify that the identifier of the image lists correctly the broadband filter used, and the verification is fine. Indeed, we find that there is this correspondence between filter number and name: 451 = v, 425 = i, 452 = r, 532 = u, 450 = b, and 632 = u.

Now we can use this file to create other catalogs to be used in the further processing. Use any text editor to open `all_head.dat`, and then save portions of it, comprising just bias, flat field and scientific images.

1.2 Master BIAS

The first 10 images are BIAS frames. The BIAS level must be subtracted from all images, before any other processing. It is recommendable to monitor the BIAS level during the observing run, in the evening, morning, and perhaps even during the night. In this case, the bias frames were taken in the afternoon. If the BIAS level is constant across the CCD surface, then it is just fine to subtract this constant value from all the frames. If there are systematic features, then one must use a real image. However, any processing of the images introduces some error, so it is better to obtain a BIAS frame that has the least possible pixel-to-pixel scatter. That is why one takes several BIAS frames, and then combines them into a mean image.

Thus, the first thing to do is to inspect every BIAS frames to decide if we need to create the mean frame. And, in such case, if some of the images must be rejected because they have problems. We can use the procedure `dispall.prg` to do the job. It loads every image contain in a catalog (in this case, `bias.cat`), and then gets a cursor on the image. The right button will exit so the next image is loaded. The left button will do a statistics in a small area. We then see that the image `dfsc0329.bdf` has an artificially high level in the lower right corner. The other images look very similar, and the BIAS level looks constant across the CCD. So we can use a constant in the further image processing. The similarity of the images can be checked in a more quantitative way, by

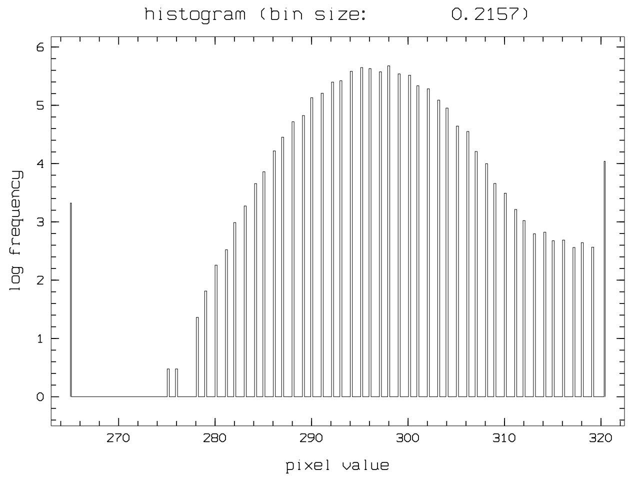

Figure 1.1: plot of the histogram of the pixel value distribution for a single `bias` frame

dividing each of them by one reference frame.

Just to show how to proceed in the alternate case, we create the mean BIAS image with MIDAS

```
Midas 006> average/image BIAS = bias-good.cat ?  ?  average
```

where the catalog `bias-good.cat` has been created by editing `bias.cat`. In order to find what BIAS value is to be used, we perform a statistics of one of the BIAS images, e.g.

```
Midas 007> stat/ima dfsc0338 plot=p excess=265,320
```

This gives us a mean value of 296.6 ADU, and a standard deviation of $\sigma_1 = 4.14$ ADU. The plot of the histogram is shown in Fig. 1.1. A similar operation yields the graph shown in Fig. 1.2, and in this case we obtain a mean value of 296.6 adu, and a standard deviation of $\sigma_9 = 1.49$ adu. Comparing the two dispersions, we then find, as expected, $\sigma_1/\sigma_9 = \sqrt{9}$. The noise in the average frame is $\sqrt{9}$ times lower than in the single frame.

Since we used amplifier B in high gain mode, the conversion factor is $0.77\,e^-/\mathrm{ADU}$, so the noise for one BIAS image corresponds to $\simeq 3.2\,e^-$. This is indeed close to the noise measured in recent CCD tests[1].

Now we proceed to subtract the BIAS to each image

```
Midas 008> exec/cat compute/ima no-bias.cat = no-bias.cat - 297
```

where `no-bias.cat` has been created from `src.cat`, by removing the BIAS frames.

[1]See The DFOSC CCD page `http://www.ls.eso.org/lasilla/Telescopes/2p2T/CCDs/02/`

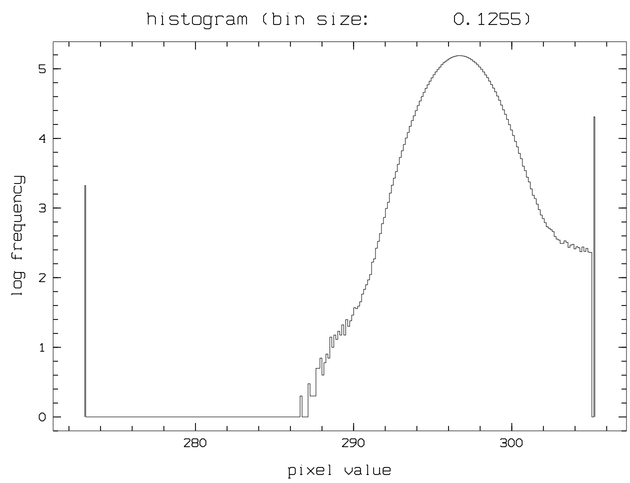

Figure 1.2: plot of the histogram of the pixel value distribution for a mean of 9 `bias` frames

A useful thing to remember is that MIDAS stores in a descriptor all the operations performed on the images. So

```
Midas 009> read/descr dfsc0621 history
frame:  dfsc0621 (data = UI2)
HISTORY: Converted from:  dfsc0621.fits
COMPUTE/IMAG dfsc0621.bdf = dfsc0621.bdf - 297
```

tells us that 297 has been subtracted from image `dfsc0621`.

1.2.1 An improved BIAS frame

The BIAS frame that has been obtained so far is a good first approximation. However, there is a detail that one might consider. The bias pattern is mostly the same for different images, but there can be a few pixels that differ greatly (e.g. those affected by cosmic rays). In this case, if we take an *average* of the frames, the average value for these pixels will be affected by that particular single image. A better way would be to take a *median* of the frames, which will remove the outliers. However, the pixel-to-pixel scatter of the median is larger than that of the mean. A possible way out is then the following: if we have N bias frames, we can create a new set of frames by taking the *median* of N indipendent *triplets* of the original images. In this way, the new frames will be free of outliers. Then, we can take the *average* of the N new BIAS frames, and we will have a master BIAS that is both free from peculiar pixels, and has a low *rms* scatter.

From statistics, with four images, the indipendent triplets that one can form is four, with five images the number is ten, and so on. So we need at least 4 frames to make this procedure work. A SUPERMONGO macro is listed in Appendix C.1, which reads a

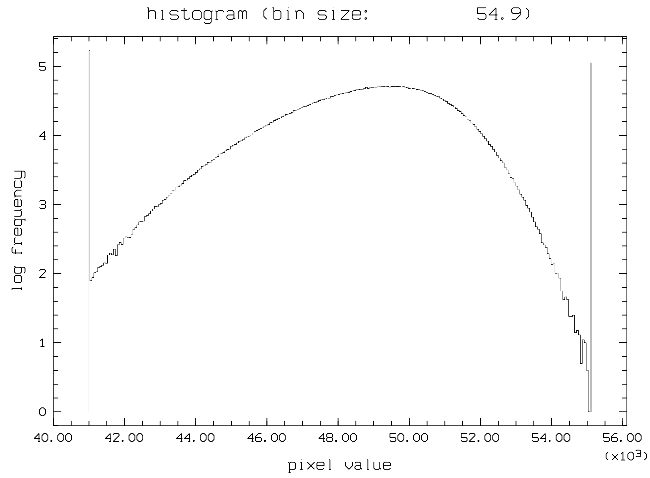

Figure 1.3: the pixel value distribution for a U sky flat field (`dfsc0623`)

catalog and creates a MIDAS procedure to combine images in such way.

1.3 Master flat-field frames

We have several flat-field images, for the five filters, and both for dome and sky. For both sky and dome flats, the master flat-field frames are obtained as *medians* of all the available images per filter. The difference with respect of BIAS frames, is that the mean level of each frame is different from the other. Before processing, we must then scale all the images to a common level. The easiest way is to 'normalize' the images, i.e. dividing each of them by its average, or median. The mean (or median) level of all the images will then be 1, so for each pixel, the median will effectively remove outliers, such as possible images of stars.

Once more, before processing, we must inspect each image, to be sure that no defects are present. Moreover, in the case of flat-fields, we must also ensure that no images are grossly saturated. A few saturated pixels in one of the images can be removed by the median filtering, but it is better in any case if saturation is not reached at all.

We start with the U filter. We use `dispall.prg` to inspect the images, and we confirm that all of them are useful (so separate catalogs for filter/type of flat field must be created). For each image, we find the median value using `stat/ima`

```
Midas 010> stat/ima dfsc0623 plot=p excess=41000,55000
frame:  dfsc0623 (data = UI2)
complete area of frame
minimum, maximum:  4.100100e+04 5.497800e+04
at pixel (418,21),(1690,93)
mean, standard_deviation:  4.885266e+04 1.866203e+03
```

Image	Median	Image	Median
Sky U		**Dome U**	
dfsc0621.bdf	4.065260e+04	dfsc0368.bdf	3.217860e+04
dfsc0622.bdf	5.234000e+04	dfsc0369.bdf	3.276065e+04
dfsc0623.bdf	4.902960e+04	dfsc0370.bdf	3.418652e+04
dfsc0624.bdf	2.580420e+04	dfsc0371.bdf	3.454275e+04
		dfsc0372.bdf	3.530123e+04
Sky B		**Dome B**	
dfsc0626.bdf	3.783546e+04	dfsc0376.bdf	2.803738e+04
dfsc0627.bdf	3.551713e+04	dfsc0377.bdf	2.810478e+04
dfsc0628.bdf	3.180806e+04	dfsc0378.bdf	2.743030e+04
		dfsc0379.bdf	2.696518e+04
		dfsc0380.bdf	2.788832e+04
Sky V		**Dome V**	
dfsc0630.bdf	2.109507e+04	dfsc0344.bdf	3.113650e+04
dfsc0631.bdf	1.972303e+04	dfsc0345.bdf	3.054523e+04
dfsc0632.bdf	1.874282e+04	dfsc0346.bdf	3.123113e+04
dfsc0633.bdf	1.711298e+04	dfsc0347.bdf	3.100312e+04
		dfsc0348.bdf	3.095292e+04
Sky R		**Dome R**	
dfsc0742.bdf		dfsc0360.bdf	3.267272e+04
dfsc0743.bdf		dfsc0361.bdf	3.252919e+04
dfsc0744.bdf		dfsc0362.bdf	3.197055e+04
		dfsc0363.bdf	3.248187e+04
		dfsc0364.bdf	3.195892e+04
Sky I		**Dome I**	
dfsc0770.bdf	1.941913e+04	dfsc0355.bdf	3.450826e+04
dfsc0771.bdf	2.720515e+04	dfsc0356.bdf	3.433630e+04
dfsc0772.bdf	2.602812e+04	dfsc0357.bdf	3.353450e+04
dfsc0773.bdf	2.332042e+04	dfsc0358.bdf	3.337445e+04
dfsc0774.bdf	2.071578e+04	dfsc0359.bdf	3.314239e+04

Table 1.2: Median values of the flat field frames that have been used to normalize the images

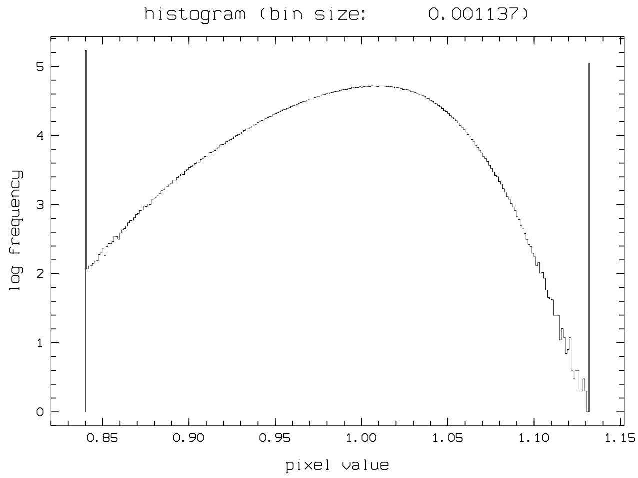

Figure 1.4: the pixel value distribution for a normalized U sky flat field (`ndfsc0623`)

```
3rd + 4th moment:  -0.486399 3.19563
total intensity:  2.0679e+11
median, 1.  mode, mode:  4.902961e+04 4.146667e+04 4.937255e+04
total no.  of bins, binsize:  258 5.490196e+01
# of pixels used = 4232931 or 93.75% of all possible pixels (= 4515096)

from 1,1 to 2148,2102 (in pixels)
```

It is important to use the `exces=xxx,yyy` option, so the statistics is computed on just the 'good' pixel values (which, in this particular case, are $\sim 94\,\%$ of the total pixels). The resulting histogram of pixel values is shown in Fig. 1.3. We then see that the median value is 49029.6 ADU. The normalized image is now created with the following command

```
compute/ima ndfsc0623.bdf = dfsc0623.bdf / 49029.6
```

And the operation is repeated for every flat field image.

If now we perform the statistics of `ndfsc0623.bdf`, the result is the following (and the corresponding histogram is shown in Fig. 1.4)

```
Midas 011> stat/ima ndfsc0623 plot=p excess=0.84,1.13
frame:  ndfsc0623 (data = R4)
complete area of frame
minimum, maximum:  8.400028e-01 1.128930e+00
at pixel (1153,27),(1573,1781)
mean, standard_deviation:  9.964029e-01 3.804192e-02
3rd + 4th moment:  -0.482879 3.18299
```

```
total intensity:  4.21743e+06
median, 1.  mode, mode:  1.000006e+00 8.508039e-01 1.006608e+00
total no.  of bins, binsize:  258 1.137255e-03
# of pixels used = 4232654 or 93.74% of all possible pixels (= 4515096)

from 1,1 to 2148,2102 (in pixels)
```

Now the master U flat is obtained as the median of the four normalized flat field images

```
Midas 076> avera/ima SKYU = sky-nu.cat ?  ?  median
```

The `sky-nu.cat` has been obtained from `sky-u.cat`, by adding a leading 'n' to the file names.

In the same manner, the sky flats for the other filters are created. However, we have a problem since two of three R flats are saturated. We will then use the dome flat fields, since we show below that the differences are not so great. A minor point is that the first two I flats have identifier *test* in their header. After inspection, we see that the images are fine, so we must assume that the observer forgot to set the correct header in this case.

All the normalization operations are gathered in the procedure `norm-all-flats.prg`, and the averaged flat field images are obtained with `mk-all-flats.prg`. Be sure to remove from the procedure the images that have already been processed in the previous steps (if you did it).

We are now able to divide every science image by its appropriate flat field, using simple procedures all similar to `proc-u.prg`. After this, the images are ready for the measurements. All the flat fielding operations are contained in `flat-all.prg`.

1.4 Instrumental photometry

We can split the photometry in two steps. Finding the centers of the standard stars, and then making the real measurement. The procedure `cg.prg` will load an image with "good" cuts, and then execute a `center/gauss` with output to a table with a name like `myframe_cen.tbl`, where '`myframe`' is the real image name. We will then point and click to each standard star, in a predefined order that must be always the same. For example, we show in Fig. 1.5 the MARK A group of photometric standard stars, both as it appears in the Landolt ([1992]) catalog, and in the real frame. We can just click on the stars following the '1', '2', etc., order already marked in the Landolt finding chart (GIF images have been copied to disk). If we want to repeat this measurement for each MARK A frame, we just do

```
exec/cat @@ cg marka.cat
```

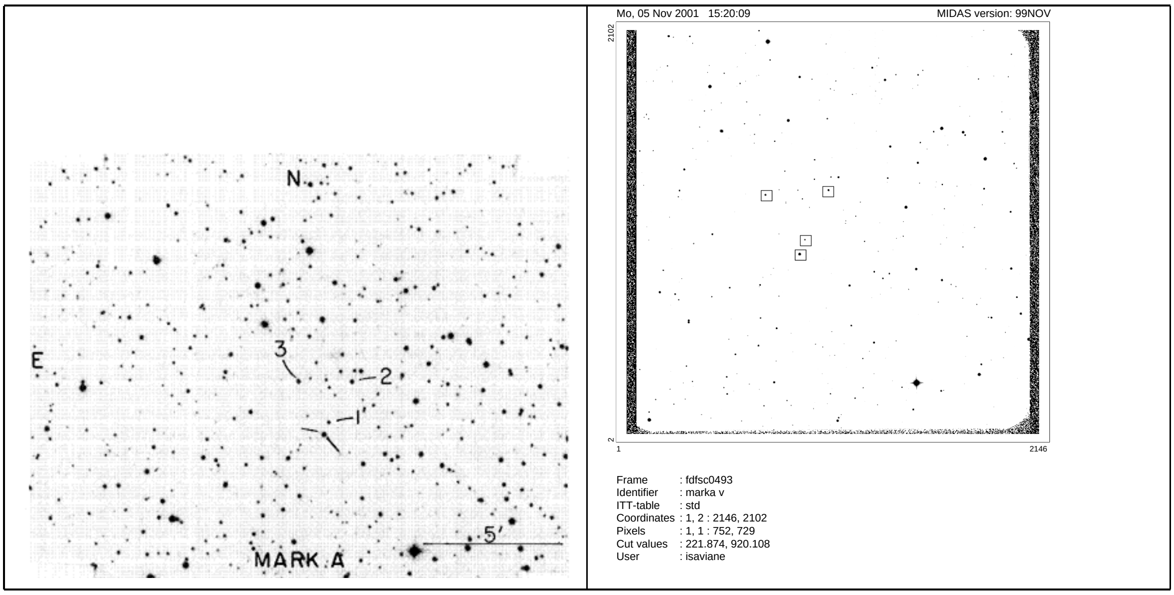

Figure 1.5: The MARK A group of photometric standard stars. The left panel shows the map provided by Landolt ([1992]), while the right panel is a reproduction of the MIDAS display, showing the same field observed with DFOSC

where `marka.cat` must have been created previously. The fields PG1633 and PG1657 have been also observed, and note that star 'D' of Landolt in the PG1657 field has no measurements in the catalog, so it should not be used. The tables with the standard star centers are then used by another procedure, `photo.prg`. This procedure creates a new file called `myframe_aperture.dat`, which contains the total magnitudes. For each star, several values of the magnitude are computed by summing the counts in concentric apertures of increasing radius. This is done to ensure that all the flux has been considered in the measurement. We then plot the total magnitude vs. radius, and we check that after some step, the magnitude stabilizes at a certain value, i.e. that the *growth curve* approaches a constant value. Even more important, we must be sure that the *color* of the standard star stabilizes after a given radius, since it is the color that is directly connected to the spectral type of the star.

The aperture photometry of all the standard star fields can be obtained with `phot_all.prg`, provided that the star centers have been measured for all fields, with the procedure outlined above.

All the steps that have been completed up to now are summarized in the MIDAS procedure `everything.prg`.

1.4.1 More discussion on the growth curve

As discussed by King ([1971]), the wings of a star can extend for arcminutes. This means that within our relatively small apertures, the total flux of the standard is still increasing

#name	LanMag	rad	mag	filt	texp	airmass	LanCol
#							
pg1633	14.397	22.000	-13.176	V	15.000	1.321	-0.212
pg1633	15.256	22.000	-12.327	V	15.000	1.321	1.015
pg1633	12.969	22.000	-14.623	V	15.000	1.321	1.090
pg1633	13.229	22.000	-14.360	V	15.000	1.321	1.138
pg1633	13.691	22.000	-13.891	V	15.000	1.321	0.650
pg1633				V			
RMS = 0.005052692723							

Table 1.3: An example of `myframe_photo.dat` file

(magnitude decreasing). However, experience shows that after some radius the fraction of flux that is missed does not depend on the seeing (King [1971]). In order to find this radius, one can plot the difference in magnitude between a fixed very large aperture and an increasing smaller one. The good radius is then the one that gives the same magnitude difference for every star, for every field, and for every filter. Adopting this radius for every star means that there will be a constant difference between the total magnitude and our instrumental magnitude, but this difference will be incorporated into the zero point of the calibration equations. In his investigation based on photographic plates, King finds that $11''$ is a good choice for the aperture, and we will assume that our aperture of $\sim 9''$ is good as well.

1.5 Calibration equations

1.5.1 Growth curves and preliminary checks

The next step is to compute a reliable instrumental magnitude for every standard star. We use SUPERMONGO to obtain the growth curve for every star in every field, and we will compute the reference aperture magnitude as the median of the last three points in the curve itself (assuming that it has reached the asymptotic behaviour). This is shown in Figures 1.6 to 1.9, where the last three points are marked in red, and the assumed magnitude is highlighted with a larger symbol on the last point. The figures have been obtained with the `pgcurve` macro (see Sect. C.1). The macro reads the `myframe_aperture.dat` table produced in the last step, and write a `myframe_photo.dat`, with all the essential data to compute the calibration equations. An example of such file is shown in Table 1.3, where the columns are the standard field name, the Landolt ([1992]) magnitude in the same filter, the largest radius in pixel of the growth curve (in this case $0.4\,''/px \times 22\,px = 8.8''$), the total instrumental magnitude, the broadband filter, the exposure time, the airmass, and the Landolt color of the standard star. The color is used in obtaining the calibration equations, and for each filter the pre-defined color is hard-coded in the macro (see C.1). The last line gives the r.m.s. difference between instrumental and standard magnitude values, and it can be used to have a quick

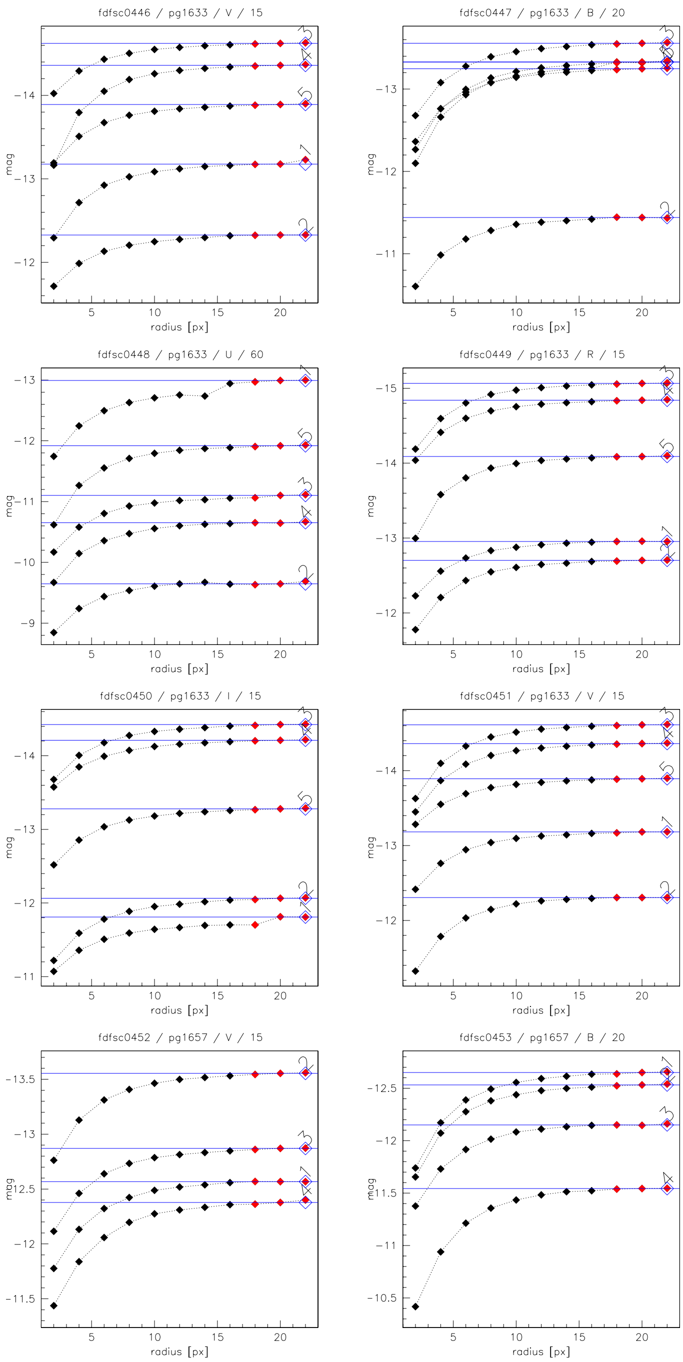

Figure 1.6: Growth curves for the first eight standard star fields

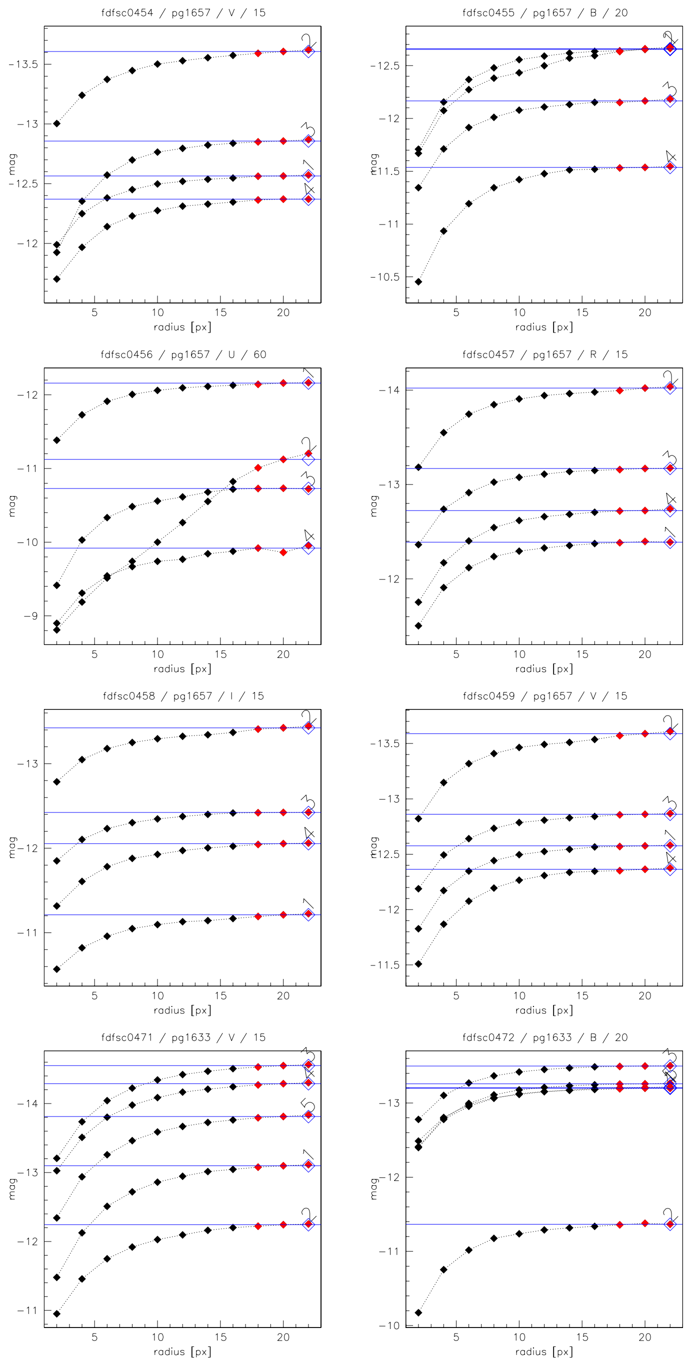

Figure 1.7: Growth curves for the second eight standard star fields

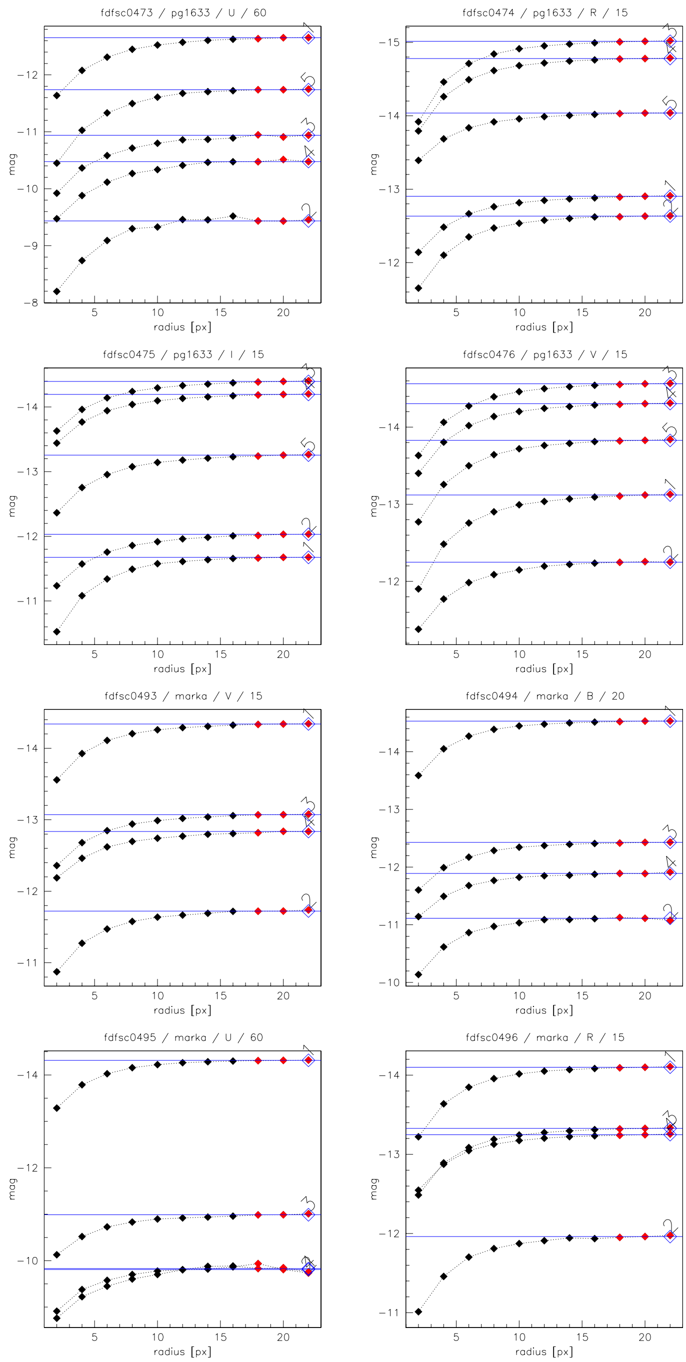

Figure 1.8: Growth curves for the third eight standard star fields

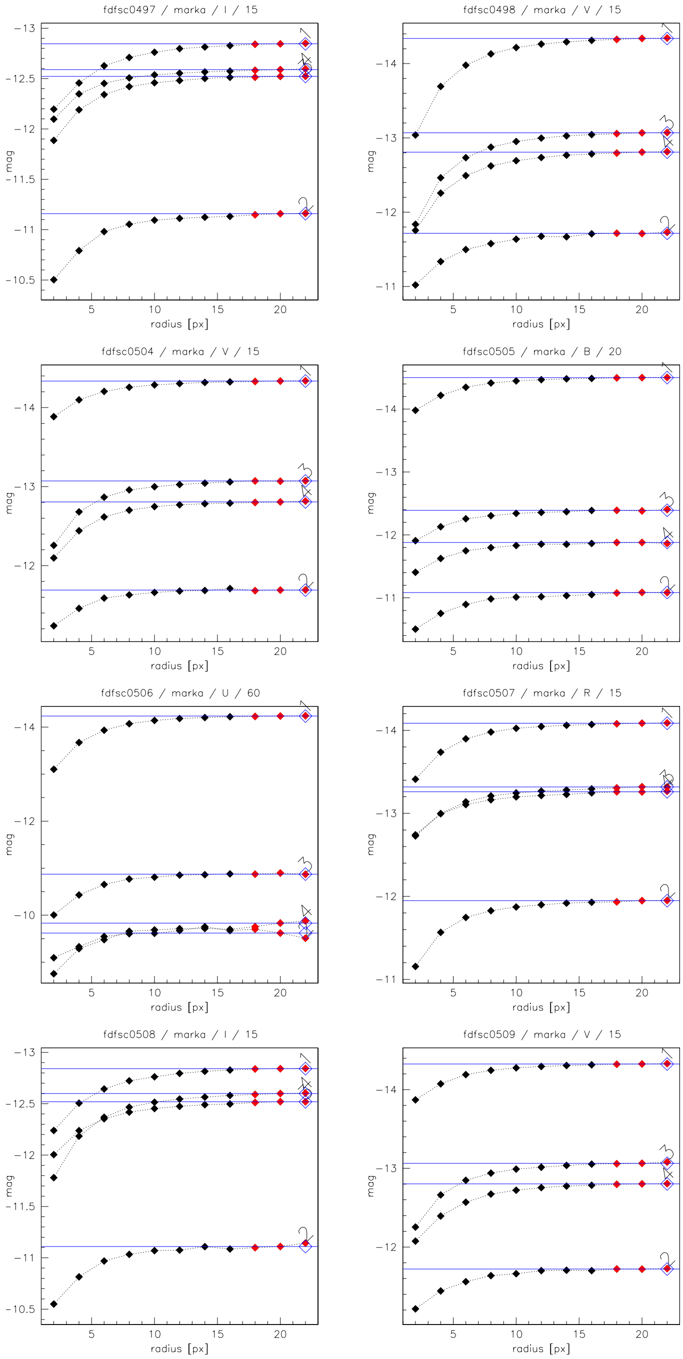

Figure 1.9: Growth curves for the fourth eight standard star fields

idea if there are any problems in the instrumental magnitudes. Using the script `check-rms.sh`, we obtain a table sorted in increasing r.m.s. difference, and if this difference is larger than, say, 0.05 magnitudes, we might want to check the results of the photometry. With this limit, we see that problems occur for images 455 and 456. The growth curves for 455 and 456 indeed show two stars with bad plots.

The macro needs `standard_head.dat`, which is obtained from `all_head.dat` with the filter name in upper case, and the ESO filter ID removed. Moreover, the standard magnitudes and colors are taken from files `marka-landolt.dat`, `pg1633-landolt.dat`, and `pg1657-landolt.dat`.

1.5.2 The calibration equations

A calibration curve is a relation that gives the magnitude of a star in a standard photometric system, as a function of the instrumental magnitude of the same star. As it was discussed in the first part of this course, the instrumental magnitude of a star is affected by the airmass at which it is observed, and by its color. Moreover, different standard star fields contain stars of different apparent brightness, so it is normally necessary to use different exposure times to ensure a good, non-saturated signal for each standard. The instrumental magnitudes must then be corrected for these three effects before comparing them to the standard magnitudes. It is easy to convince oneself that the so-called normalized magnitudes must be expressed as

$$\overline{m} = m_{\text{ins}} + 2.5 \times \log t_{\text{exp}} - K_m \times X$$

where $\overline{m}$ is the normalized magnitude, $m_{\text{ins}} = -2.5 \times \log F$ is the instrumental magnitude as measured by MIDAS (and F is the total counts in ADU), t_{exp} is the exposure time, K_m is the absorption coefficient at the effective wavelength of the filter for magnitude m, and X is the airmass. If the calibration curves can be approximated by a linear relation, then they can be written as

$$m_{\text{std}} = \overline{m} + a_m \times c_{\text{std}} + b_m$$

where m_{std} is the standard magnitude, $\overline{m}$ is the normalized instrumental magnitude obtained above, and a_m, b_m are the coefficients of the linear relation with the standard color c_{std}. The calibration equations are obtained with the SUPERMONGO macro `airmass`. The macro first searches the best absorption coefficient by minimizing the r.m.s. scatter around the linear relations, for a range of plausible K_m. Then it will find a_m and b_m with a least square method. The final calibration equations are shown in Fig. 1.10, and the corresponding parameters are listed in Table 1.4. The results for filter U have been obtained after removing the measurements from image number 456 from the calculations, and star PG 1633 from image number 448. The latter star has not been used in the I measurements either. The whole image 455 has been excluded from measure-

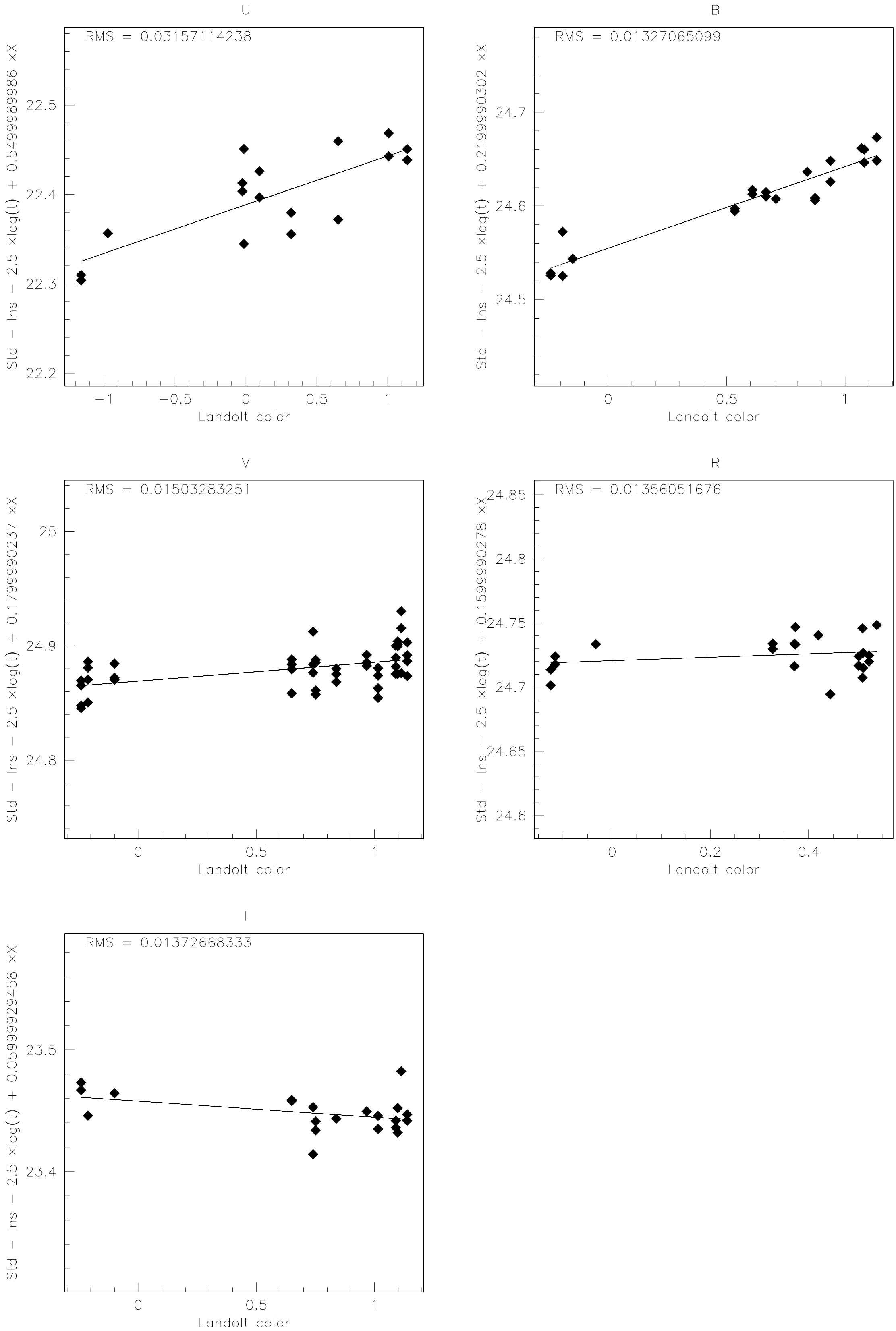

Figure 1.10: The calibration curves for the filters U, B, V, R, and I

Filter	Color	a_m	b_m	K_m	r.m.s.
U	$U-B$	0.054 ± 0.011	22.389 ± 0.008	0.55	0.032
B	$B-V$	0.087 ± 0.006	24.555 ± 0.005	0.22	0.013
V	$V-I$	0.017 ± 0.004	24.869 ± 0.004	0.18	0.015
R	$R-I$	0.013 ± 0.013	24.721 ± 0.005	0.16	0.014
I	$V-I$	-0.013 ± 0.007	23.458 ± 0.006	0.06	0.014

Table 1.4: Parameters of the calibration equations

ments of the B filter, while all measurements have been used for the V and R filters. Note that the extinction coefficients are compatible with typical values for La Silla[2].

1.6 A few remarks on crowded field photometry

TBD — see some Peter Stetson paper...

[2]See at Geneva Observatory `http://obswww.unige.ch/photom/extlast.html`

Appendix A

MIDAS procedures

A.1 all_head.prg

```
write/out " "
write/out " this creates a summary of your bdf files "
write/out " you should have created src.cat before starting "
write/out " "

$ rm all_head.dat
$ rm dfsc*dat*
exec/cat @@ image_head src.cat

$cat dfsc*.dat > all_head.dat

$rm *.dat1 *.dat2 *.dat3 *.id

write/out " "
write/out " Now all_head.dat will have lots of parameters for your
images "
write/out " "
```

A.2 image_head.prg

```
ass/print file {p1}.dat1
print/descr {p1} NPIX,O_TIME,CCDTEMP,ZDIST,CAMERA,ROTATOR
ass/print file {p1}.dat2
print/descr {p1} ESO.INS.OPTI-1.NAME,ESO.INS.OPTI-2.ID,ESO.INS.OPTI-
3.NAME
ass/print file {p1}.dat3
print/descr {p1} ESO.INS.OPTI-4.NAME,ESO.INS.OPTI-5.NAME,ESO.DET.MODE

$ echo `cat {p1}.dat1` | awk '{print $2}' > x1
$ echo `cat {p1}.dat1` | awk '{print $16,$20}' > x2
$ echo `cat {p1}.dat1` | awk '{print $7,$8,$18,$22,$24}' > x3
$ echo `cat {p1}.dat2` | awk '{print $9}' > x4
$ echo `cat {p1}.dat2` | awk '{print $7,$11}' > x5
$ echo `cat {p1}.dat3` | awk '{print $7,$9,$11}' > x6

ass/print file {p1}.id
print/descr {p1} IDENT
$ grep IDENT {p1}.id | awk '{print $2,$3,$4,$5,$6,$7,$8,$9,$10}'
> x0

$ echo `cat x1 x0 x4 x2 x3 x5 x6` > {p1}.dat
```

A.3 dispall.prg

```
crea/displ
load/lut heat
load/itt log

define/parameter p1 ?  I " give catalog name:  "

$ cat {p1} | grep -v "Image catalog" > x.cat
!  stat of full image
!  $cat x.cat | awk '{print "stat/ima "$1" ; load/ima " $1" cuts=f,ihap
scale=-3; stat/ima " $1" cursor "}' > ld.prg
!  stat of cursor
$cat x.cat | awk '{print "load/ima " $1" cuts=f,ihap scale=-3; stat/ima
" $1" cursor "}' > ld.prg

@@ ld.prg
```

A.4 norm-all-flats.prg

```
!  --- normalize U sky flats
!
compute/ima ndfsc0621.bdf = dfsc0621.bdf / 40652.6
compute/ima ndfsc0622.bdf = dfsc0622.bdf / 52340.0
compute/ima ndfsc0623.bdf = dfsc0623.bdf / 49029.6
compute/ima ndfsc0624.bdf = dfsc0624.bdf / 25804.2

!  --- normalize B sky flats
!
compute/ima ndfsc0626.bdf = dfsc0626.bdf / 3.783546e+04
compute/ima ndfsc0627.bdf = dfsc0627.bdf / 3.551713e+04
compute/ima ndfsc0628.bdf = dfsc0628.bdf / 3.180806e+04

!  --- normalize V sky flats
!
compute/ima ndfsc0630.bdf = dfsc0630.bdf / 2.109507e+04
compute/ima ndfsc0631.bdf = dfsc0631.bdf / 1.972303e+04
compute/ima ndfsc0632.bdf = dfsc0632.bdf / 1.874282e+04
compute/ima ndfsc0633.bdf = dfsc0633.bdf / 1.711298e+04

!  --- normalize I sky flats
!
```

```
compute/ima ndfsc0770.bdf = dfsc0770.bdf / 1.941913e+04
compute/ima ndfsc0771.bdf = dfsc0771.bdf / 2.720515e+04
compute/ima ndfsc0772.bdf = dfsc0772.bdf / 2.602812e+04
compute/ima ndfsc0773.bdf = dfsc0773.bdf / 2.332042e+04
compute/ima ndfsc0774.bdf = dfsc0774.bdf / 2.071578e+04

!  --- normalize U dome flats
!
compute/ima ndfsc0368.bdf = dfsc0368.bdf / 3.217860e+04
compute/ima ndfsc0369.bdf = dfsc0369.bdf / 3.276065e+04
compute/ima ndfsc0370.bdf = dfsc0370.bdf / 3.418652e+04
compute/ima ndfsc0371.bdf = dfsc0371.bdf / 3.454275e+04
compute/ima ndfsc0372.bdf = dfsc0372.bdf / 3.530123e+04

!  --- normalize B dome flats
!
compute/ima ndfsc0376.bdf = dfsc0376.bdf / 2.803738e+04
compute/ima ndfsc0377.bdf = dfsc0377.bdf / 2.810478e+04
compute/ima ndfsc0378.bdf = dfsc0378.bdf / 2.743030e+04
compute/ima ndfsc0379.bdf = dfsc0379.bdf / 2.696518e+04
compute/ima ndfsc0380.bdf = dfsc0380.bdf / 2.788832e+04

!  --- normalize V dome flats
!
compute/ima ndfsc0344.bdf = dfsc0344.bdf / 3.113650e+04
compute/ima ndfsc0345.bdf = dfsc0345.bdf / 3.054523e+04
compute/ima ndfsc0346.bdf = dfsc0346.bdf / 3.123113e+04
compute/ima ndfsc0347.bdf = dfsc0347.bdf / 3.100312e+04
compute/ima ndfsc0348.bdf = dfsc0348.bdf / 3.095292e+04

!  --- normalize R dome flats
!
compute/ima ndfsc0360.bdf = dfsc0360.bdf / 3.267272e+04
compute/ima ndfsc0361.bdf = dfsc0361.bdf / 3.252919e+04
compute/ima ndfsc0362.bdf = dfsc0362.bdf / 3.197055e+04
compute/ima ndfsc0363.bdf = dfsc0363.bdf / 3.248187e+04
compute/ima ndfsc0364.bdf = dfsc0364.bdf / 3.195892e+04

!  --- normalize I dome flats
!
compute/ima ndfsc0355.bdf = dfsc0355.bdf / 3.450826e+04
compute/ima ndfsc0356.bdf = dfsc0356.bdf / 3.433630e+04
compute/ima ndfsc0357.bdf = dfsc0357.bdf / 3.353450e+04
```

```
compute/ima ndfsc0358.bdf = dfsc0358.bdf / 3.337445e+04
compute/ima ndfsc0359.bdf = dfsc0359.bdf / 3.314239e+04
```

A.5 mk-all-flats.prg

```
avera/ima SKYU = sky-nu.cat ?  ?  median
avera/ima SKYB = sky-nb.cat ?  ?  median
avera/ima SKYV = sky-nv.cat ?  ?  median
avera/ima SKYI = sky-ni.cat ?  ?  median
avera/ima DOMEU = dome-nu.cat ?  ?  median
avera/ima DOMEB = dome-nb.cat ?  ?  median
avera/ima DOMEV = dome-nv.cat ?  ?  median
avera/ima DOMER = dome-nr.cat ?  ?  median
avera/ima DOMEI = dome-ni.cat ?  ?  median
```

A.6 proc-u.prg

```
compute/ima fdfsc0495.bdf = dfsc0495.bdf / SKYU
compute/ima fdfsc0506.bdf = dfsc0506.bdf / SKYU
compute/ima fdfsc0456.bdf = dfsc0456.bdf / SKYU
compute/ima fdfsc0448.bdf = dfsc0448.bdf / SKYU
compute/ima fdfsc0473.bdf = dfsc0473.bdf / SKYU
```

A.7 flat-all.prg

```
compute/ima fdfsc0494.bdf = dfsc0494.bdf / SKYB
compute/ima fdfsc0505.bdf = dfsc0505.bdf / SKYB
compute/ima fdfsc0453.bdf = dfsc0453.bdf / SKYB
compute/ima fdfsc0455.bdf = dfsc0455.bdf / SKYB
compute/ima fdfsc0447.bdf = dfsc0447.bdf / SKYB
compute/ima fdfsc0472.bdf = dfsc0472.bdf / SKYB
compute/ima fdfsc0497.bdf = dfsc0497.bdf / SKYI
compute/ima fdfsc0508.bdf = dfsc0508.bdf / SKYI
compute/ima fdfsc0458.bdf = dfsc0458.bdf / SKYI
compute/ima fdfsc0450.bdf = dfsc0450.bdf / SKYI
compute/ima fdfsc0475.bdf = dfsc0475.bdf / SKYI
compute/ima fdfsc0495.bdf = dfsc0495.bdf / SKYU
compute/ima fdfsc0506.bdf = dfsc0506.bdf / SKYU
compute/ima fdfsc0456.bdf = dfsc0456.bdf / SKYU
compute/ima fdfsc0448.bdf = dfsc0448.bdf / SKYU
```

```
compute/ima fdfsc0473.bdf = dfsc0473.bdf / SKYU
compute/ima fdfsc0498.bdf = dfsc0498.bdf / SKYV
compute/ima fdfsc0493.bdf = dfsc0493.bdf / SKYV
compute/ima fdfsc0504.bdf = dfsc0504.bdf / SKYV
compute/ima fdfsc0509.bdf = dfsc0509.bdf / SKYV
compute/ima fdfsc0452.bdf = dfsc0452.bdf / SKYV
compute/ima fdfsc0454.bdf = dfsc0454.bdf / SKYV
compute/ima fdfsc0459.bdf = dfsc0459.bdf / SKYV
compute/ima fdfsc0446.bdf = dfsc0446.bdf / SKYV
compute/ima fdfsc0451.bdf = dfsc0451.bdf / SKYV
compute/ima fdfsc0471.bdf = dfsc0471.bdf / SKYV
compute/ima fdfsc0476.bdf = dfsc0476.bdf / SKYV
compute/ima fdfsc0470.bdf = dfsc0470.bdf / SKYV
!
!  use dome flat field for the R filter
!
compute/ima fdfsc0496.bdf = dfsc0496.bdf / DOMER
compute/ima fdfsc0507.bdf = dfsc0507.bdf / DOMER
compute/ima fdfsc0457.bdf = dfsc0457.bdf / DOMER
compute/ima fdfsc0449.bdf = dfsc0449.bdf / DOMER
compute/ima fdfsc0474.bdf = dfsc0474.bdf / DOMER
```

A.8 cg.prg

```
!  needs debug, otherwise produces xxx.bdf_cen !!
!
@a autocuts {p1}
load/ima {p1} scale=1
center/gauss cursor {p1}_cen
```

A.9 photo.prg

```
!  makes aperture photometry for a number of radii
!
!  the centers will be taken from the table xxx_cen, if xxx is the
!  name of the image
!
!  xxx_cen.tbl must be created with center/gauss
!
!  use @@ phot_all.prg to photometer all images
!
```

```
define/parameter p1 ?  I " Give image name [without extension]:  "


define/local ap/i/1/1

$ rm {p1}_aperture.dat
do ap = 2 24 2

  write/out ap = {ap}
  magnitude/circle {p1},{p1}_cen {p1}_cen{ap} @{ap},@2,@2
  @@ print_ptable {p1}_cen{ap} {p1}_cen{ap}.dat
  $ paste {p1}_aperture.dat {p1}_cen{ap}.dat > x
  $ mv x {p1}_aperture.dat
enddo
```

A.10 print_ptable.prg

```
ass/print file x1
print/tab {p1} :XCEN,:YCEN,:MAGNITUDE n
ass/print file x2
print/tab {p1} :MAG_SIGMA n
!!  ,:SKY,:SKY_SIGMA,:MAGPIX,:FLUX n
$ paste x1 x2 > {p2}
```

A.11 phot_all.prg

```
$ sh mkphot_prg.sh !!  will produce pt.prg
@@ pt
```

A.12 **everything.prg**

```
!
!  this will go from raw images to calibration curves!!
!
!  (provided that the necessary image catalogs have been created)
!
!
!  Assume that src.cat contains the starting images
!
!  bias-good.cat   contains bias images without dfsc0329.bdf
!  no-bias.cat     contains all images that are _not_ bias frames
!
!  sky-u.cat       contains all U sky flat fields
!  sky-b.cat           "     "  B
!  sky-v.cat                    V
!  sky-i.cat                    I
!  dome-u.cat      contains all U dome flat fields
!  dome-b.cat                   B
!  dome-v.cat                   V
!  dome-r.cat                   R
!  dome-i.cat                   I
!  sky-nu.cat       contains all U sky normalized flat fields
!  sky-nb.cat           "     "  B
!  sky-nv.cat                    V
!  sky-ni.cat                    I
!  dome-nu.cat      contains all U dome normalized flat fields
!  dome-nb.cat                   B
!  dome-nv.cat                   V
!  dome-nr.cat                   R
!  dome-ni.cat                   I
!
!  marka.cat        catalog of images of Landolt Mark A field
!  pg1633.cat       catalog of images of Landolt PG 1633 field
!  pg1657.cat       catalog of images of Landolt PG 1657 field
!
!  standard_head.dat from all_head.dat with some editing:
!   * filter put in upper case
!   * ESO filter id removed
!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!  --- create master BIAS
```

```
!
average/image BIAS = bias-good.cat ?  ?  average

!  --- subtact BIAS constant from all images
!
exec/cat compute/ima no-bias.cat = no-bias.cat - 297

!  --- normalize all sky and dome flat fields
!
@@ norm-all-flats.prg

!  --- create median sky flat fields
!      (or use mk-all-flats.prg)
!
avera/ima SKYU = sky-nu.cat ?  ?  median
avera/ima SKYB = sky-nb.cat ?  ?  median
avera/ima SKYV = sky-nv.cat ?  ?  median
avera/ima SKYI = sky-ni.cat ?  ?  median

!  --- create median dome flat fields
!
avera/ima DOMEU = dome-nu.cat ?  ?  median
avera/ima DOMEB = dome-nb.cat ?  ?  median
avera/ima DOMEV = dome-nv.cat ?  ?  median
avera/ima DOMER = dome-nr.cat ?  ?  median
avera/ima DOMEI = dome-ni.cat ?  ?  median

!  --- divide all the rest by the corresponding flat fields
!
@@ flat-all.prg

!  --- for every image, get standard star centers interactively
!      Click on every field following the same order of stars
!
!      you will get tables like   myframe_cen.tbl  when cg is
!      used standalone.  It will produce  myframe.bdf_cen if
!      used with exec/cat, so please debug!
!
exe/cat @@ cg marka.cat
exe/cat @@ cg pg1633.cat
exe/cat @@ cg pg1657.cat

!  --- now compute aperture photometry for every image, based
```

```
!      on std star centers just found
!
@@ phot_all

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!
!  --- the procedure produces myframe_aperture.dat
!      and from now on you must switch to \supermongo\
!
!      execute macros pgcurve, and then airmass, contained in
!      combine_frames.mac
!
```

Appendix B

Unix scripts

B.1 mkphot_prg.sh

```
#
ls *_cen.tbl | cut -d.  -f1 | sed s/_cen// |  awk '{print "@@ photo
"$1}' > pt.prg
```

Appendix C

SUPERMONGO macros

C.1 combine_frames.mac

```
#
# [ combine_frames.mac ]
#
# mkprg         :  will create the MIDAS procedure to combine a set of
#               :  N>=4 frames using the median and then the average
#
# gcurve        :  take table produced by photo.prg and make growth curve
# pgcurve       :  create PS files with growth curve for every frame
# airmass       :  find absorption coefficients
#

airmass         #
                define enorm 1.5
                define ebig 3
                define SORT {"sort -r"}  # for Solaris
                define SORT {"sort"}     # for Linux

                expand $enorm

                define kfilt ?  { Filter nr.:  }

                !cat *photo.dat | grep -v "\#" | grep -v RMS > x
                da x read {_1 1.s}
                print x {_1}
                !sort -u x | grep -v "\#" > std_names.list
                da std_names.list
                read {nvec 1.s}

                # print {nvec}

                define kname 0

                define stdname ( nvec[$kname] )

                !cat *photo.dat | grep $stdname | $SORT > x
                da x
                read {_5 5.s}
                print x {_5}
                !sort -u x | grep -v "\#" > std_filters.list
                da std_filters.list
                read {fvec 1.s}
```

```
# print {fvec}


define filter ( fvec[$kfilt] )

echo
echo Working with $filter
echo

# !cat *photo.dat | grep $stdname | grep $filter | sort > x
!cat *photo.dat | grep -v RMS | grep $filter | $SORT -k 2 > x

da x

read {id 1.s  mlandolt 2 rad 3 mins 4\
      ffiltname 5.s texp 6 airmass 7 clandolt 8}

erase
set y = mlandolt - mins - 2.5*lg(texp)
limits clandolt y
box
ptype 4 0

!  rm krms.dat
do K = -1,1,0.01 {

 # define K 0.3

 set y = mlandolt - mins - 2.5*lg(texp) + $K * airmass
 points clandolt y
 lsq clandolt y clandolt z rms
 connect clandolt z
 !echo $K $rms >> krms.dat
}

echo
echo krmas.dat generated ...
echo
define nul ?

da krms.dat
read {k 1 rms 2}
limits k rms
era
box
points k rms
sort {rms k}

define rmsmin ( rms[0] )
define kmin ( k[0] )
ctype red
expand $ebig
relocate $kmin $rmsmin
dot
expand $enorm
ltype 1
vline $kmin
ltype 0
ctype default

define K ( $kmin )

define nul ?

$0.gplot
```

airmass.gplot

```
#
# --- final plot
#
location 4500 31000 4500 31000

erase

 set y = mlandolt - mins - 2.5*lg(texp) + $K * airmass
 set y1 = y + 0.1
 set y2 = y - 0.1
 set tmp = y1 concat y2

 limits clandolt tmp
 era
 box 1 2 3 3
 xlabel Landolt color
 ylabel Std - Ins - 2.5 \times log(t) + $K \times X
 ptype 4 3
 angle 45
 expand $ebig
 points clandolt y
 angle 0
 expand $enorm
 lsq clandolt y clandolt z rms
 connect clandolt z

 echo
 echo a = $a+/-$sig_a / b = $b+/-$sig_b / rms = $rms
 echo
 echo Absorption coeff = $K
 echo

 # toplabel $stdname / $filter
 toplabel  $filter

 relocate (6000 30000)
 label RMS = $rms

ResetLocation

define psfile "calib"$filter".ps"
define clfile "calib"$filter".dat"

!  echo > $clfile
!  echo a = $a+/-$sig_a / b = $b+/-$sig_b / rms = $rms >> $clfile
!  echo  >> $clfile
!  echo Absorption coeff = $K  >> $clfile
!  echo  >> $clfile

set dy = abs( y - ($a*clandolt+$b) )
sort {dy id mlandolt  mins  airmass clandolt}
print {id dy mlandolt  mins  airmass clandolt}

echo
echo --> airmass.gplot to redo the last plot
echo
echo   \$psfile is $psfile
echo \$clfile is $clfile  with cal curve
echo
```

pgcurve

```
#
define rootname dfsc  # assuming images are $rootname+number

define headfile standard_head.dat # contains relevant information
```

```
                              # for every $rootname+number file

define psflag ?  { See PS? }

!ls *_aperture.dat | cut -d.  -f1 | sed s/_aperture// > x
da x
read {bdfname 1.s}
do k=0,dimen(bdfname)-1 {

  define bdf ( bdfname[$k] )
  echo $bdf now used by gcurve_
  gcurve_
}


#!ls *_gcurve.ps | awk '{print x,x,\\$1}' | sed 's/ /ghostview/' > shps.sh
#!sh shps.sh
```

gcurve_

```
# non-interactive version of gcurve, to be used after
# defining bdf
#
define nap 11  ## how many apertures
define enorm 1.5
define ebig  3.0
define ehuge 5.0

expand $enorm

# define bdf ?  { Midas image [no extension]:  }


# --- get further data for the standard star from the
#     header file
#
!echo $bdf | sed 's/dfsc/@/' | cut -d@ -f2 > x
da x read {_1 1.s} define numfile ( _1[0] )
define stdfile $rootname""$numfile
echo $stdfile
!  grep $stdfile $headfile | sed 's/\#//'| sed 's/*//' > x
da x
read {_2 2.s _3 3.s _4 4 _5 5}
define _2 ( _2[0] )
define _3 ( _3[0] )
define _4 ( _4[0] )
define _5 ( _5[0] )

# we assume such format for the aux file
#
# image_name std_name filter t_exp zenith_dist
#      1        2       3     4         5
#
set stdname = _2
set stdfilt = _3
set  stdexp = _4
set stdzdis = _5

set finalfilt = stdfilt
set finalname = stdname


# --- create filename with aperture photometry data
#
define table $bdf"_aperture.dat"

# --- create final output file name
```

```
#
define ofile $bdf"_photo.dat"

da $table

read {_1 1}
define nlines ( dimen(_1) )

set dimen(m) = $nap

# --- define PS file
#
define psfile $bdf"_gcurve.ps"
device postencap $psfile

# --- find good limits for the plot
#
GoodLims

set r = 2,22,2
limits r $mmax $mmin
erase
box 1 2 3 3
xlabel radius [px]
ylabel mag
toplabel $bdf / $_2 / $_3 / $_4 # / $_5

ptype 4 3
angle 45
expand $ebig
set dimen( finalmag) = 0
set dimen( finalrad) = 0
set dimen( finalexp) = 0
set dimen( finalair) = 0

do ln=1,$nlines {

  read row _l $ln
  set m[ 0] = _l[  3]
  set m[ 1] = _l[  9]
  set m[ 2] = _l[ 15]
  set m[ 3] = _l[ 21]
  set m[ 4] = _l[ 27]
  set m[ 5] = _l[ 33]
  set m[ 6] = _l[ 39]
  set m[ 7] = _l[ 45]
  set m[ 8] = _l[ 51]
  set m[ 9] = _l[ 57]
  set m[10] = _l[ 63]
  #set m[11] = _l[ 69]
  points r m
  ltype 1
  connect r m
  ltype 0

  label $ln

  # --- take median of last 3 points
  #
  set _v = m[8] concat m[9] concat m[10]
  set _r = r[8] concat r[9] concat r[10]
  ctype red
  points _r _v
  ctype default
  stats_med _v _vmed _sq
```

```
  ctype blue
  hline $_vmed
  ctype default

  # --- update vector of final mag value et al.
  #
  set airmass = 1/cos(stdzdis*3.141592653/180)
  set  finalmag =  finalmag concat $_vmed
  set  finalrad =  finalrad concat r[dimen(r)-1]
  set finalname = finalname concat stdname
  set finalfilt = finalfilt concat stdfilt
  set  finalexp =  finalexp concat stdexp
  set  finalair =  finalair concat airmass
}

# --- check termination points of growth curves
#
expand $ehuge
ptype 4 0
ctype blue
points finalrad finalmag
ctype default
angle 0

dev x11

# --- see growth curve
#
PS.$psflag

# --- find the right magnitude vector from the
#     Landolt catalog
#
GetStdMag

# --- get rms diff between instrumental and std mag
#
lsq finalmag LanMag finalmag tmp rms

# --- save information to file
#
print $ofile '%s%9.3f%9.3f%9.3f  %s  %9.3f%9.3f%9.3f\n'\
    {finalname LanMag finalrad finalmag\
   finalfilt finalexp finalair LanCol}

!echo RMS = $rms >> $ofile

PS.y            #
                !  ghostview $psfile &
PS.n            #
PS.             #
GetStdMag       #
                define lname ( stdname[0] )
                !ls *landolt* | grep $lname > x
                da x read {_1 1.s}
                define lfile ( _1[0] )
                echo
                echo I am using standard catalog:  $lfile
                echo

                da $lfile
                read {V 4 BV 5 UB 6 VR 7 RI 8 VI 9 errorV 12\
                      errorBV 13 errorUB 14 errorVR\
                    15 errorRI 16 errorVI 17}
```

```
                ##   print {V BV UB VR RI VI}

                set B = V + BV
                set U = B + UB
                set R = V - VR
                set I = V - VI
                define lfilt ( stdfilt[0] )

                CreaMagVec.$lfilt

CreaMagVec.U    #
                set LanMag = U
                set LanCol = UB
CreaMagVec.B    #
                set LanMag = B
                set LanCol = BV
CreaMagVec.V    #
                set LanMag = V
                set LanCol = VI
CreaMagVec.R    #
                set LanMag = R
                set LanCol = RI
CreaMagVec.I    #
                set LanMag = I
                set LanCol = VI


gcurve          #
                define nap 11  ## how many apertures
                define enorm 1.5

                expand $enorm

                define bdf ?  { Midas image [no extension]:  }

                define table $bdf"_aperture.dat"

                da $table

                read {_1 1}
                define nlines ( dimen(_1) )

                set dimen(m) = $nap

                set r = 2,22,2
                limits r -10 -15
                erase
                box 1 2 3 3
                xlabel radius [px]
                ylabel mag

                ptype 4 3
                angle 45
                do ln=1,$nlines {

                  read row _l $ln
                  set m[ 0] = _l[  3]
                  set m[ 1] = _l[  9]
                  set m[ 2] = _l[ 15]
                  set m[ 3] = _l[ 21]
                  set m[ 4] = _l[ 27]
                  set m[ 5] = _l[ 33]
                  set m[ 6] = _l[ 39]
                  set m[ 7] = _l[ 45]
                  set m[ 8] = _l[ 51]
```

```
  set m[ 9] = _l[ 57]
  set m[10] = _l[ 63]
  #set m[11] = _l[ 69]
  points r m
  ltype 1
  connect r m
  ltype 0

  # --- take median of last 3 points
  #
  set _v = m[8] concat m[9] concat m[10]
  set _r = r[8] concat r[9] concat r[10]
  ctype red
  points _r _v
  ctype default
  stats_med _v _vmed _sq
  ctype blue
  hline $_vmed
  ctype default
}
angle 0
```

mkprg

```
#
# --- you must have a MIDAS catalog already prepared
#
define cat ?  { MIDAS Catalog of frames:  }
da $cat

# --- first line is skipped
#
lines 2 100000
read {f 1.s}
# print {f}

# --- randomize image sequence in the catalog
#
set casuale = random(dimen(f))
sort {casuale f}

# --- generate name of combined image, from catalog
#     name
#
!echo $cat | cut -d.  -f1  > x
da x read {_1 1.s}
define prg supercombine.prg ## ( _1[0] )
define final ( _1[0] )

define nframes ( dimen(f) )

# --- 3 loops to generate independent triplets
#
define l 0
!rm $prg
!echo $ rm tmp\*.bdf >> $prg

echo Generating Midas procedure ...
do i=0,$nframes-1 {

  echo external index = $i
  do j=$i+1,$nframes-1 {

    do k=$j+1,$nframes-1 {

      if ( $i!=$j && $i!=$k && $j!=$k) {
         define l ( $l+1 )
```

```
          define f1 ( f[$i] )
          define f2 ( f[$j] )
          define f3 ( f[$k] )
          #
          # --- writing MIDAS instruction
          #
          !echo avera/ima tmp$l = $f1,$f2,$f3 \?  \?  median >> $prg
      }
     }
   }
 }

 # --- final MIDAS instructions for the generation of
 #     combined image
 #
 !echo crea/icat tmp tmp\*.bdf >> $prg
 !echo avera/ima $final = tmp.cat \?  \?  average >> $prg

 echo
 echo Now you can use $prg to combine images
 echo It will take images from $cat
 echo and create $final.bdf
 echo
 echo Note that you can use the first $nframes lines
 echo of the procedure, since it can create
 echo $l (>=$nframes) new images
 echo
```

```
GoodLims         #
                 da $table
                 read {_3     4\
                    _9     10\
                    _15   16\
                    _21   22\
                    _27   28\
                    _33   34\
                    _39   40\
                    _45   46\
                    _51   52\
                    _57   58\
                     _63   64q}

                 set dimen (_lm) = 0
                 vecminmax      _3  _3min  _3max
                 set _lm = _lm concat $_3min concat $_3max
                 vecminmax      _9  _9min  _9max
                 set _lm = _lm concat $_9min concat $_9max
                 vecminmax     _15 _15min _15max
                 set _lm = _lm concat $_15min concat $_15max
                 vecminmax     _21 _21min _21max
                 set _lm = _lm concat $_21min concat $_21max
                 vecminmax     _27 _27min _27max
                 set _lm = _lm concat $_27min concat $_27max
                 vecminmax     _33 _33min _33max
                 set _lm = _lm concat $_33min concat $_33max
                 vecminmax     _39 _39min _39max
                 set _lm = _lm concat $_39min concat $_39max
                 vecminmax     _45 _45min _45max
                 set _lm = _lm concat $_45min concat $_45max
                 vecminmax     _51 _51min _51max
                 set _lm = _lm concat $_51min concat $_51max
                 vecminmax     _57 _57min _57max
                 set _lm = _lm concat $_57min concat $_57max
                 vecminmax     _63 _63min _63max
                 set _lm = _lm concat $_63min concat $_63max
```

```
vecminmax _lm mmin mmax

define mmax ( $mmax + 0.2 )
define mmin ( $mmin - 0.2 )
```

Appendix D

Iraf Useful Documentation

[This is taken from *Photometry in* IRAF, by Lisa Wells]
On-line help pages are available for all the tasks mentioned in this document using help or phelp. More information on topics covered in this manual may be found in the following documentation, accessed by anonymous ftp to iraf.noao.edu or 140.252.1.1:

Photometry in IRAF, by Lisa Wells, February 1994 (this manual).

(iraf/docs/photom.ps.Z)

A Beginner's Guide to Using IRAF[draft], by Jeannette Barnes, August 1993.

(pub/beguide.ps.Z)

A User's Guide to CCD Reductions with IRAF, by Philip Massey, June 1992.

(iraf/docs/ccduser2.ps.Z)

A User's Guide to Stellar CCD Photometry with IRAF, by Phillip Massey

and Lindsey E. Davis, April 1992. (iraf/docs/daophot2.ps.Z)

Specifications for the Aperture Photometry Package, by Lindsey Davis, May

1989. (iraf/docs/apspec.ps.Z)

A User's Guide to the IRAF Apphot Package, by Lindsey Davis, May 1989.

(iraf/docs/apuser.ps.Z)

A Reference Guide to the IRAF/DAOPHOT Package, by Lindsey E. Davis,

January 1994. (iraf/docs/daorefman.ps.Z)

FOCAS User's Guide, by Francisco Valdes, September 1993. (iraf/docs/focas/focasguide.ps.Z)
Faint Object Classification and Analysis System: Standard Test Image

Results, by Francisco Valdes, April 1989. (iraf/docs/focas/standards.ps.Z)

The following documentation is accessed by anonymous ftp to stsci.edu or 130.167.1.2:

STSDAS Users Guide, Nov 1991. (documents/stsdas-docs/user/UserGuide/UserGuide
12.ps) STSDAS Calibration Guide. (documents/stsdas-docs/user/CalibrationGuide/CalibGuide
12.ps) Quick Reference Cards for IRAF and STSDAS.

(documents/stsdas-docs/user/UserQuickRef.ps)

The following source may also be helpful:

Davis, L. E. & Gigoux, P. 1993, PHOTCAL: The IRAF Photometric Calibration

Package, eds., Worrall, D. M., Biemesderfer, C., and Barnes, J., ASP Conf. Series Vol. 52,
479.

Bibliography

[200x] XXX, something about distance determination

[1974] Golay, M. 1974, *Introduction to Astronomical Photometry*, D. Reidel Publishing, Dordrecht-Holland

[1962] Hardie, R. H. 1962, in STARS AND STELLAR SYSTEMS, Vol. 2, Astronomical Techniques, ed. W. A. Hiltner, University of Chicago Press, 178

[1990] Harris, W. E. 1990, PASP, 102, 949.

[1981] Harris, W. E., FitzGerald, M. P., & Reed, B. C. 1981, PASP, 93, 507

[1989] Howell, S. B. 1989, PASP, 101, 616

[1991] Howell, S. B. (ed.), 1991, Astronomical CCD Observing and Reduction Techniques, ASP Conf. Series, Vol. 23

[2001] Howell, S. B. 2000. *Handbook of CCD Astronomy*, Cambridge University Press

[1971] King, I.R. 1971, PASP, 83, 199

[1992] Landolt, A.U. 1992, AJ, 104, 340

[1979] Philip, A. G. Davis (ed.), 1979, *Problems of Calibration of Multicolor Photometric Systems*, Dudley Observatory, Schenectady, New York

[1987] Stetson, P. B. 1987, PASP, 99, 191

[1990] Stetson, P. B. 1990, PASP, 102, 932

[1988] Stetson, P. B., & Harris, W. E. 1988, AJ, 96, 909.

[1992] Wagner, R. M. 1992, ASPC, 23, 160

Index