A Student's Guide to CCD Photometry Data Processing using IRAF

with Dr. Fernández
at the University of Central Florida

written by
© Robert Grisetti
Version of July 3 2006

Table of Contents

1. Introduction

This manual is meant to provide a set of step-by-step instructions and practical information for the photometric data reduction and analysis process done for Dr. Yanga Fernández in the University of Central Florida Astronomy Department. It is specifically tailored to Dr. Fernández's preferred methodology and typical data processing requirements, and has been kept concise and practical, exploring the process only to the extent that is necessary to perform it. This is meant to be an aid to doing the process, not to be a completely rigid method, and there are parts of it that are flexible or open ended as to how exactly it is done.

There are many other more broad, extensive, and theoretical works on the subjects of photometry, CCD data reduction and analysis, IRAF, and the others touched upon here. Thus, the purpose of this is not to provide an understanding of why the process is done or how it works, but rather to be an instructional reference for understanding how it is done in this specific context.

This context will be the first information to be explored by looking at how to use Linux and its programs as necessary for IRAF, as well as basic maneuvering in IRAF itself. From there we will proceed through the data reduction process, and then through the data analysis process, ending with a look at further analysis and the final data archiving process.

2. Orientation

This section will provide an orientation for the reader of the system surrounding the data reduction and analysis process. This includes very basic instruction and information about using the Linux OS, the IRAF program, and various other intermediary or useful programs. For more extensive and detailed information about these than given here, see the sources on the reference list, especially source number 1 on the reference list.

2.1. Using Linux

Some of the computers in the Astronomy Lab use a version of Linux called Fedora as the Operating System. Two of these (as of the time when this was written) have *xgterm* on them,

which will be necessary for some of the applications we will use in reducing and analyzing the data. Very basic instruction will be given on navigating and using the Linux OS below.

2.1.1. Using the Graphic User Interface

Fedora has a fairly intuitive interface. There is a tool panel, a task panel, and a desktop. Various things may be added or taken off from the tool panel and task panel, including an Applications menu, an Actions menu, a date and time display, a workspace switcher, and a system monitor, among other things. The Applications menu gives access to various programs, including *gedit* and the *Firefox* web browser. The Actions menu gives access to Run and Search applications, as well to log off and shut down commands. The workspace switcher allows the user to switch between different same-user desktops called workspaces, and to move windows from one workspace to another. The System Monitor, if added, will allow the user to view the processor usage to see how hard the computer is working and if there may be a problem. This and other features may be added by selecting "Add to Panel" after right clicking on one of the panels, and more tool panels may be added by selecting "New Panel" from the same menu.

Fedora's GUI uses windows to display folders and applications. Using these windows is very similar to using those in any Windows OS. They can be very helpful in keeping the data and other files you will be working with organized. Every user on the system has a home folder, which appears on their desktop, and provides a place to put their own files and folders. Items on the desktop are stored in the 'Desktop' folder in the home directory.

There are several programs available in Linux that are very useful for the process in the rest of this manual. Among these are *gedit* and *Terminal*, which are described below.

2.1.2. The *gedit* text editor

The default text editor on the Linux machines in the Astronomy Lab is called *gedit*. It may be accessed by opening a text document, or by clicking on the "Text Editor" shortcut in the Applications/Accessories menu on the toolbar. It is useful for this process since it has very limited formatting. It has a tab feature for viewing more than one file in a single window, and many other features, including find and replace, which may be very useful for you.

2.1.3. The *Terminal*

The *Terminal* in Linux is essentially a command prompt that allows users to interact with the system without going through the GUI. The *Terminal* uses Unix commands, some of which will be reviewed below for those unfamiliar with Unix. The *Terminal* will be used in this photometry process almost exclusively only to access other programs we will be using, namely *xgterm*, the specialized terminal through which IRAF is run, as well as *ds9*, the display program we use with

IRAF.

To start the *Terminal*, right-click on the desktop and select "Open Terminal". To open either of the programs mentioned above, simply type in the name of the program, with an '&' on the end. For example, entering 'xgterm&' will yield:

```
[username@computername ~]$ xgterm&
[1] 2430
[username@computername ~]$
```

Or something like it; the job number (2430) will be different each time. The '&' brings the prompt back up so further commands may be entered.

From the *Terminal*, you have access to the file directory on the computer, and may view it and manipulate it. The starting directory is your profile's home folder, which should be called the same thing as your username, and is the first part of the prompt. This also true for *xgterm*. The following are some commands for moving around in the file directory:

cd	changes directory to home directory
cd [subdirectory]	changes directory to that entered within current directory
cd ..	changes directory to that immediately above
back	changes directory to that immediately previous

Besides moving around, the *Terminal*'s Unix commands allow you to interact with the file directory in other ways, including the following:

ls	lists contents of directory
dir	lists contents of directory
mkdir	makes a new directory
rmdir	removes a directory
pwd	shows current directory pathway

Other useful commands include those that deal with jobs such as the following:

jobs	lists jobs currently running
kill [job number]	kills the job of the specified number
CTRL–C	interrupts processes
exit	exits the *Terminal*

These are useful when there are problems with programs running. The first command will let you see the programs currently running from the *Terminal* and their assigned numbers, and the second command will let you manually shut down the program if it causing trouble or not responding. The 'kill' command is not a good way to normally shut down a program, however. 'CTRL-C' is another way to deal with problematic jobs. It stops any current processes running from the *Terminal*. This command is also one that is not normally good to use.

During the data reduction and analysis process covered in this manual, the *Terminal* is principally used only for opening the *xgterm* and *ds9* programs, and so further knowledge about the *Terminal* and how to use it is not necessary.

2.1.4. The *xgterm* terminal

The *xgterm* application is a special terminal that IRAF requires for the use of its visual and interactive components. Therefore, IRAF should be run in *xgterm* and instead of any other type of terminal. The directory and other commands are mostly the same as those in the regular Terminal. The principal use of *xgterm* in this manual will be running IRAF, which is displayed in this special terminal. It is IRAF that we are really interested in and that most of the work covered in this manual will take place, so *xgterm* need not be further reviewed.

2.1.5. The *ds9* viewer

The *ds9* program is an image viewer used by IRAF to display and allow for interaction with images. It allows for the examination and photometric measuring of images, and will be used much in the process outlined in this manual. A brief overview of the most important functions of the viewer will be discussed here.

The program is capable of holding multiple images for display at once. It does this by using frames. Each frame contains a separate image. The number of frames used is unlimited, though the more images held in frames, the slower the computer becomes, and the higher the likelihood of problems occurring. Above the image display is a row of command buttons and above that a row of command menu tabs. Switching between the menu tabs will display different sets of command buttons. Above these is displayed various information about the frame and the image in it, and next to that there are two thumbnail displays – one of the entire image and a box showing what part of it is being displayed, and the other of a zoomed in portion of the image centered on where the cursor is.

The edit tab and pull-down menu gives you access to switching between various types of cursors, among other things. To navigate between frames, you may use the commands in either the frame tab or pull-down menu. Frames may be displayed singly, or tiled. Another useful feature of the frames is the ability to blink between them. This comes in handy when trying to locate an object that moves significantly relative to the stellar background, such as a comet or an asteroid. The zoom commands control the zoom level. The center zoom command centers the image view on the origin; to center an image somewhere else, bring the cursor over it and click the center mouse button. You may also do this by using the pan cursor type in the edit menu to center the image. One other important command is located in the color menu. The reset command there will revert the displayed image contrast back to its default if it is accidentally or otherwise altered. Changing the contrast and brightness is done by holding the right mouse button down and moving the mouse along the vertical and horizontal axes, respectively.

There are many other commands and functions in *ds9*, though they are not essential or important to this process, and so will not be reviewed here.

2.2. IRAF

IRAF is short for Image Reduction and Analysis Facility. It is a program that will allow for, among many other things, the reduction and analysis of CCD photometric data, which is what you will be working with. IRAF will be the core program involved in our photometry process.

On any one computer with IRAF installed, there can be many different users. Each user has separate login and parameter files. To use IRAF, you must first create these. This is done by entering the command 'mkiraf' into *xgterm* while in your home directory. The 'mkiraf' command will create a folder called 'uparm' that will store your parameter files and a default 'login.cl' file that will allow you to login to IRAF and to set user-specific preferences for using IRAF.

There are several of these preferences in the 'login.cl' file that should be changed to improve the process and to be able to follow the steps in the rest of this manual. The first change is in the image directory setting. IRAF creates a default image directory for each user. Every new image it outputs, IRAF will store in that default directory. This is a result of line 8 that appears as follows in the 'login.cl' file:

```
set     imdir           = "/iraf/imdirs/grisetti/"
```

A single default image directory is not in our interests however, since we will be doing many different things with many different types of images. Therefore, change the line to read as the following:

```
set     imdir           = "."
```

This will alter your user settings to output images into whatever directory you are located in, allowing for a more flexible environment.

Another change in user settings to make concerns the default image size. Initially this is set as shown in this line in the 'login.cl' file:

```
#set     stdimage          = imt800
```

This is too small for most images we will be working with, however. Therefore, change the line, by removing the '#' and replacing the number, to look like this:

```
set     stdimage          = imt2048
```

Now that your user settings are correct, go ahead and log on.

To log on to IRAF, type 'cl' into the *xgterm* prompt, while in your home directory, which is, again, the one it starts in. This will bring you to the main screen and menu of IRAF. Tasks in IRAF belong to packages, which are organized into a hierarchical directory. To execute a task, you must load the package it is in. To load a package, you must load the package it belongs to on the level above it. Once a package is loaded however, you may run any task or load any package

in it anytime during the session and from anywhere in the task directory. The two letters in the prompt indicate the first two letters of the package most recently loaded. To see a list of the tasks or packages that may be run or loaded from the current package location, enter '?' into the prompt. To see a list of all packages and tasks currently loaded, enter '??' into the prompt. To see a description of a specific task or package, enter 'help' and the name of the task or package. The 'help' command will be useful for you if there is any task this manual does not clarify enough for you about how to use it, why to use it, or how it works.

An important thing to note is that IRAF is case sensitive. When entering commands, task names, file names, or parameters, it is important to keep the case of the letters correct to avoid problems or confusion. Also of note is that when using IRAF in *xgterm*, the 'Backspace' key does not work; the 'Delete' key should be used instead, and works the same as 'Backspace' while in IRAF.

IRAF tasks often use external files as input and create new external files as output. Because of this, IRAF gives a user a location in the file directory by using the terminal it is running in. This allows the user to refer to files without entering their entire path every time. At the same time however, it makes being located in the correct directory essential to using the correct files. To move around in the file directory in IRAF, simply use the same commands for this as in the Terminal (including the 'back' command). Also, to escape tasks as a last resort, or to end tasks that are being problematic, the 'CTRL–C' command works in IRAF as well. There is another command in IRAF that is helpful after using 'CTRL–C' or ending problematic tasks – the *flprcache* task. Entering 'flpr' flushes the process cache for the program, essentially terminating any tasks that have finished but have not exited. These then become idle, though may be problematic for continued operation. Terminating them with *flprcache* may therefore be helpful when IRAF is not running correctly.

2.2.1. Parameters

IRAF tasks have variables that affect what they do called parameters. Each task has a set of embedded parameters, and some have separate parameter files. For the tasks to function the way we want them to, their parameters must be set correctly.

There are two types of parameters – required and hidden. Required parameters are necessary for the task to run, and usually include input and output file names and paths. Required parameters appear at the prompt when running the task to be set before the task is actually executed. Even if they were already set, they will still be prompted, with the set parameter shown as the default in parenthesis. Hidden parameters refer to unnecessary parameters that are not prompted when the task is run. When viewing a task's parameters, hidden parameters are distinguished by a set of parenthesis that surround each parameter. This difference is demonstrated below:

Required parameter

```
calib  =              Output calibrated standard indices file
```

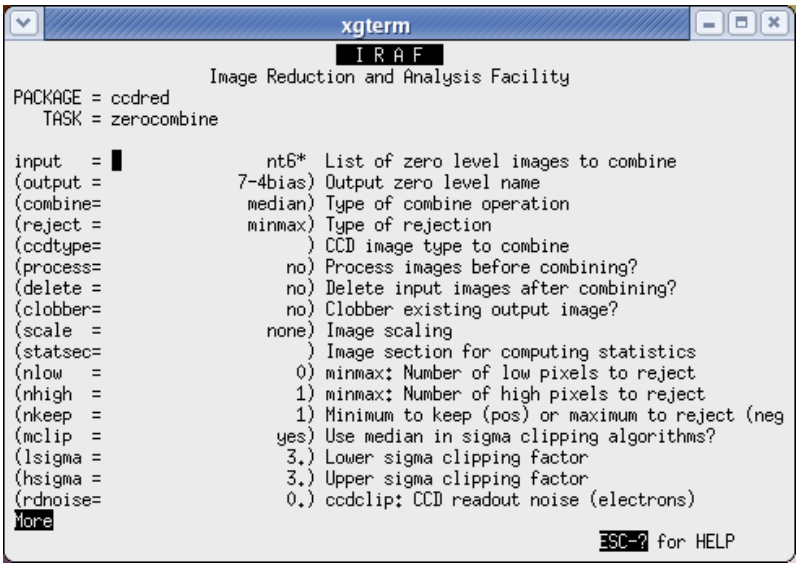
Hidden parameter

(catalog=) List of standard catalog files

To edit a task's parameters, the *eparam* task is used, entered and abbreviated as shown.

>epar [task name] (e.g. >epar zerocombine)

This will display the parameters of the task, and allow the user to move between them and edit



them. An example of a parameter display in *eparam* is shown below:

For this parameter list, there is only one required parameter – the rest are hidden. While editing a parameters file, you can move up and down using the up and down arrow keys, and edit a parameter simply by starting to type. If you want to change the parameter only slightly, you can copy and paste by selecting the parameter setting and pressing the middle mouse button. This will paste the selected text back into the parameter as if you entered it, and you may edit it from there. Some parameter files, like the one shown above, may have more lines than may be shown in your *xgterm* window. When this is the case, it is continued on another 'page' so to speak, that may be accessed by moving down the list past the last parameter listed.

Inside the *eparam* task, as in some other IRAF tasks, there is something called colon command mode. It is called this because it is entered by typing a colon. Once entered, there are a variety of different commands the user may enter, depending on the task. The *eparam* task requires the use of the colon command ':q' to save the changes and quit. The command ':q!' quits without saving changes. For more information on the various commands in colon command mode, refer to the help file for the *eparam* task. Other tasks have colon command modes as well, and the commands for these can be found in the help file as well.

2.2.2. Basic image tasks

Another commonly used task in this process is *display*. This task opens an image into a frame in the *ds9* viewer to be seen, examined, and measured. This is done by entering the following:

> disp [filename] [frame number] (e.g. > disp n2047r.fits 3)

If there are multiple files with the same name and different extensions, such as an '*.imh' format file and a '*.fits' format file of the same image, make sure to include the extension in the filename. Otherwise, it is not needed. Refer back to section 2.1.5. above for more details about using the *ds9* program itself.

The IRAF tasks *imstatistics* and *imexamine* are also very useful for photometry data processing. They provide different ways of manually examining an image, and are helpful in confirming an image's usefulness, among other things. The *imstatistics* task displays several key statistical values of an image, such as the number of pixels, the minimum, maximum, and mean pixel values, and others. To use *imstatistics*, use the following command:

> imstat [filename] (e.g. > imstat nt3.002.fits)

Also, you may display statistical information with the task about more than one image at once. To do this, you must write a list of images instead of entering a single image.

When writing a list, you may either type each file in separating them by commas (no space), or you may use a portion of the file names and the asterisk. For example:

nt3.002.fits,nt3.003.fits,nt3.004.fits,...

or

nt3*

The second list will include all files beginning with 'nt3' so it will be important to make sure that there are no files that would be included in your list that you don't want there when using the asterisk list. When writing an asterisk list, you may use the asterisk at the beginning of the filename, at the end of the filename, at one or more places in the middle of the filename, or any combination of the above. Remember to be mindful about how IRAF will interpret the list you enter.

The *imexamine* task provides for a further more in depth examination and analysis of the image than *imstatistics*. Using plots and graphs, going beyond a few statistics, it allows the user to look at the image in various visual ways, and thus to get a greater, more detailed and revealing understanding of it. The task will be very useful at certain points in the process outlined in this manual, and may be helpful to you outside of its instructed use.

To use *imexamine*, the image to be examined must be open in the *ds9* program. When it is simply enter in the following command:

> imexam [filename] (e.g. > imexam n2068r.fits)

When you enter the command, the *ds9* viewer will automatically switch to the frame with the

referenced image in it. In *ds9*, the cursor should now be something like a blinking annulus. Use the annulus, or the small zoomed-in box in the upper right, to align the center of the cursor with the center of the object. There are several ways that this task allows you to examine the object. Once the cursor is centered on the object, press the key that corresponds with the examination you want to perform. To see a full list of *imexamine*'s capabilities, view the help file for the task in IRAF. The type of examination that is most important to this procedure is the radial plot of an object's pixel values. To command the task to do this, press 'r' when the cursor is centered on the object. The curve itself is fitted to the plot of pixel values as related to radius, and both the curve and the points are shown.

3. Data Reduction

3.1. Introduction

Now that you are well oriented with IRAF's general features, we will go through the CCD photometric data reduction process. This is a procedure essentially meant for cleaning the images that will be used for scientific analysis. The result of the reduction process will be a set of corrected images that will be used for analysis, which will be cleared of noise from many various sources, external to the object(s) and our interest. Before starting though, there are several other noteworthy points to be explained.

Firstly, it is good to have well organized data. During the reduction process, before the corrected science data set is acquired, you will be working more data than at the start. The tasks will sometimes leave you with several versions of the same image. It is good to have a consistent and intuitive naming system to avoid confusion, and to split the images into folders for each step of the process, at least until you are finished. In the end you will have a smaller set of data to analyze, but before then you will be working with many times the number of images that you began with.

Second, keep to using IRAF tasks in single filter format, even if you will be working with multiple filters. Photometric observations may be made in different particular regions of the electromagnetic spectrum using different optical filters. When reducing and analyzing data, it is vitally important to keep images using filters of one type separate from images of another. Bias frames do not use filters, and so will be used for correcting images of all filters. For every filter used in the object images, each will have its own corresponding flats and standard stars. In many of its tasks, IRAF accommodates multiple filters, allowing them to be put through the various tasks at the same time. For this manual however, tasks will not be done with multiple filters. It is

simpler and less risky to go through the process for the separate filters one at a time. This way there is less confusion, and there is a smaller chance of mixing the data from different filters, which would make the results erroneous and useless.

To run the data sets of different filters through one at a time, simply treat them as distinct data sets, to be passed through the reduction and analysis processes separately.
When this is the case, you do not need to specify in IRAF the filter identifications at all, though you may if you want to. Otherwise, simply use 'INDEF' for them all. Make sure to keep track of the filters, even though IRAF doesn't need to.

For a more thorough or abstract treatment of the reduction process, see the sources on the reference list, especially number 6.

3.2. *Ccdred* - Data Reduction

Out of the many packages and tasks available in IRAF we will primarily be using only a few. The first of these covered will be that for data reduction, the first step in the processing of photometric data.

Photometric data reduction for CCD images in IRAF centers around the use of a handful of tasks located in the *noao.imred.ccdred* package. Both the importance and operation of these tasks will be explained below.

Before this however, I will note that while IRAF will use the *.fits image filetype, the CCD Reduction tasks output images of a different format. These output images are split into two files each – a header file (*.imh) and a pixel file (*.pix). These should not be separated, and neither may be moved from the folder it was created in. If you do move it, the image will simply no longer be read correctly by the display program, even if you move the file back to its original folder. While you may go along with IRAF and leave the images in this format as you continue, you may instead find it useful to convert them into the more versatile *.fits format after every step. To find information about doing this, go to the section about the *wfits* command.

3.2.1. *Zerocombine* - Making a master bias

The first step in CCD photometry data reduction is to combine the bias images into one image called a master bias. This will essentially be an average of all of the bias frames, and will be used to remove additive effects that would make the photometric results erroneous.

The IRAF task used to create the master bias image is called *zerocombine*. To properly use this task, as well as any IRAF task, you must edit the parameters using the *eparam*. Therefore, in the

command prompt of the *noao.imred.ccdred* menu, enter 'epar zerocombine' to open the parameters of the task. There are many parameters, but in the *zerocombine* task, only 7 should be edited. (Note: IRAF uses the term 'zero level frame' instead of 'bias frame' to refer to the same thing. Hence the name *zerocombine*.)

The first of these is the "input" parameter. This will be the list of bias images in your data set. You may write a list in one of the ways described above, in Section 2.2.2. Make sure you are only inputting the desired bias images. This is one reason why organization is so important.

The next important parameter in *zerocombine* is "output". This will be the name of your master bias image. There are no requirements for what the master bias must be named, except for the Linux and IRAF naming requirements, though it should be something that distinguishes the file as the master bias and makes sense to you, and possibly other who may have to understand the work you are doing. Also, if you will be reducing multiple data sets and will be creating multiple master bias images, it may be helpful to use the same naming conventions for each data set. This applies for other task output files in the process as well. An example of a name for a master bias file would be 'n3mbias'.

The next parameter to set is "combine". This refers to the method of combining the images, and should be set to 'median'.

Also important is the "reject" parameter. Z*erocombine* processes the images one pixel at a time and will reject the pixels that it deems to be outliers depending on how this parameter is set. This should be 'minmax' so that at each pixel combination it will not use the highest pixels and the lowest pixels. When using the 'minmax' rejection method, the "nlow" and "nhigh" parameters must be specified. These dictate how many low pixels are rejected and how many high pixels are rejected, respectively.

To determine what numbers to put for this, as well as to make sure the bias images are good images, look at them in the *ds9* display program. The master bias frame should be an average of all of the bias frames. As such, the relative contrast of pixels should be very low, and the images should therefore be a fairly uniform grey. There may be pixels on the frames however, that are of a significantly higher value than the surrounding static. These should generally be rejected in making a master bias. The “reject” parameter is what does this. The high outliers on the frames should go towards determining the "nhigh" value. Low outliers, which are of a lower value than the surrounding static, contribute towards determining the "nlow" value, though these are very rare. A zero amount should be used if there are no high or low value outlying pixels, respectively. Typically, there will be no low outliers, so set "nlow" to zero. Also, while there usually are high outliers, there are typically not enough of them to justify setting "nhigh" higher than 1. Even if it appears unnecessary to use “nhigh” also, there are usually plenty of frames to make up for a missing pixel in each calculation, so set "nhigh" as 1 to be safe. If you think either of these two settings would be wrong for your data set, discuss the matter with Dr. Fernández to determine what should be done.

The last parameter to check is "ccdtype". It should be left blank. To make a parameter blank simply type in two double quotes like so

```
(ccdtype= ""          )
```

Then press 'Enter' and it will be blank. When all of the parameters are set, terminate the *eparam* task by typing ':q'. This will bring back the 'cc>' prompt. At this point, you will want to make sure of which bias images are good to be used for the master bias. To do this, use the *imstat* tool and the *ds9* program to examine them. With *imstat*, the important thing to notice will be the total count number for each image. All of the bias images you use in *zercombine* should have very close totals. Any far outliers should not be used to make the master bias. Since the frames should all have the same number of pixels, the mean may also be used to compare the images. Displaying the images in the *ds9* program will help you to make sure that the reject numbers are correct.

Once you are finished with these examinations, you may execute the *zerocombine* task. Remember to make sure *xgterm* is looking in the correct directory.

The output file will appear, and you may continue the next step of unbiasing the flat frames.

3.2.2. *Ccdproc* - Removing bias from the flats

The next step in data reduction is the removal of instrument generated error in the flat frames using the master bias. For this we will use the "ccdproc" task.

Again, the parameters must be set correctly. You will again want to set the list of images that the task will operate on, in the "images" parameter. Also, the "output" should set here as well, however "ccdproc" is a correction task, so you will have an output of the same number of images as your input. Therefore, another list is needed. An explicit list may be created in "output" (e.g. nt3.016,nt3.017,nt3.018,...), however an asterisk list will not work as well for output, since the asterisk will only result in IRAF choosing that portion of the name, which likely will make connecting the output images with their corresponding input images. You can again save yourself work though, by creating a text file list and using copy and paste commands in it. To use the text file list as your output, simply enter '@[textfilename]' (e.g. '@flatfinallist') into the parameter. Be sure here again that the text file is in the folder where *xgterm* is looking.

"Ccdtype" should here also be left blank. All of the y/n questions should be left 'no' except for "trim" and "zerocor" which should both be set as 'yes'.

The only remaining parameter to set is "zero", which should be the path and filename of the master bias.

With all of the parameters properly set, you may go ahead and execute the task. The result will be another set of flat frames, de-biased and ready for combining into a master flat.

3.2.3. *Flatcombine* - Making a master flat

Once the flat frame images have been corrected with the master bias, they must then be combined to make a master flat. This is done in IRAF with the use of the "flatcombine" task.

The parameters of "flatcombine" are very similar to those of *zerocombine*. The "input" list must be entered, and the output master flat must be named in “output”. "Combine" must be set to 'median', and "reject" must be set to 'minmax' again with "nlow" set to zero and "nhigh" set to one, unless otherwise is thought appropriate. "Ccdtype" should again be left blank. This time however, the "scale" parameter requires attention. For making a master flat, this parameter must always be set to 'mode'. Also, "process," "subsets," "delete," and "clobber" should all be set to 'no' if they are not already.

With the parameters set, you should examine these images in a similar fashion as the bias images. Instead of looking for relative outliers however, the flats should be compared to an actual range of values. The total count for a flat should be somewhere in the range of 10,000-25,000. If a flat's total count is outside of this range, be very wary about using it. Again, use the ds9 program to check the images for irregularities or further considerations that should be taken with them.

With the examinations complete, execute the task. The result is the master flat image, which will be used to divide out any multiplicative errors from the science frames. To verify the quality of the master flat, try using it to flatten a few of the individual flat frames. The results should be simple, uniform images. If one of the results is not uniform, then either the individual flat is bad, or the master flat is.

Before continuing to correct the science frames, the master flat needs to be altered. There will be a strip on one of its edges that is composed of pixels at zero. Since the master flat is divided into the science frames to correct for multiplicative effects, then this region must not stay as it is. If it does, the data correction in the next step would not work, since for a part of the frame, it would be attempting to divide by zero. This strip must be altered therefore, to give every pixel a nonzero value. There is a way to do this in IRAF and in the future it will be included in this process, though for now it will be simpler to give the master flat to Dr. Fernández to be fixed. There is a different program that will do this, though it has an entirely different environment, so he will do it for you. Let him know you have gotten to this point, so that he can trim your flat for you.

3.2.4. *Ccdproc* - Correcting the raw data

The final step in data reduction makes use again of the IRAF task "ccdproc" to correct the raw science frames, using both the master bias and the master flat acquired in previous steps.

"Images" and "output" must have file lists entered in again. The other parameters remain the same as in the first use of "ccdproc" with the exception of "flatcor" and "flat," which are to be set as 'yes' and the file name of the master flat, respectively. With these parameters set, execute the final task. The output will be the final data.

3.3. Wfits - Converting the images

One minor step remains, however. Even if you have not been converting your other images into the *.fits format and don't want to move files around, these final images, as well as the master bias and master flat, should be converted to allow for their transfer to storage eventual transfer to storage.

The task you will use to do this is *wfits*. The syntax for using *wfits* is as follows

```
> wfits [filename1.imh] [filename2.fits]
```

The use of 'filename1' and 'filename2' is meant simply to emphasize that the name of the converted file should be different than the original when using *wfits* (e.g. 'nt3.046.imh' might convert to 'n3046.fits'), though it need not be.

Now that you have the final, converted science frames, the process of data reduction is complete, and the data is ready for the analysis process.

4. Data Processing and Analysis

4.1. Introduction

With the reduced data, analysis may finally be performed. The ultimate goal of most photometry data analysis is to acquire the magnitudes of the objects in the images. The object magnitudes are essentially the results of photometry, and may be used in many various ways.

Before we begin it may be helpful to divide the final images into object type. Every data set will have at least two types - the targets of the photometry, and standard stars. The standard stars are

stars that have very well known absolute magnitudes, and so are used in translating the instrumental magnitudes of the target objects into the absolute magnitude system. There may also be several different groups of objects within a single data set (e.g. some from comet Encke, and some from the Trojan asteroids). It may be advantageous to divide the images into separate folders along the lines of these groups.

To acquire the absolute object magnitudes, we will go through several steps. The first of these is the acquiring of the instrumental magnitudes of the standard stars and the objects using an IRAF task called *phot*. Next, the standard stars are used to determine the values of the coefficients in the magnitude equation, since both absolute and instrumental magnitudes are known for them. With these coefficients, the instrumental magnitudes of the objects are then compiled and used to find the absolute magnitudes.

For more information or a more thorough or abstract explanation, see the sources on the reference list, especially numbers 2, 3, 4, 5, 7, and 8.

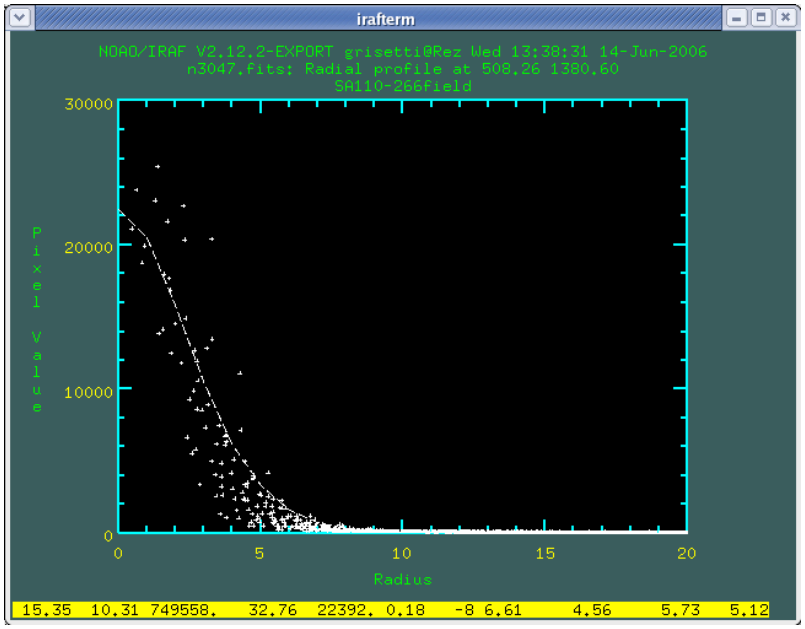
4.2. *Phot* - Finding the instrumental magnitudes

The IRAF task *phot* is one that you will be using a lot of. It is a software package that is used to acquire instrumental magnitudes of objects in digital images. It is found in the *noao.digiphot.apphot* package. It uses a handful of other tasks, each with separate parameter files, and all located in the same *apphot* package. The parameter files of interest to us are *centerpars@*, *datapars@*, *fitskypars@*, and *photpars@*. (Note: When referencing the parameter file in the prompt, the @ may be left out.)

4.2.1. Editing the parameters

Centerpars@ has only 1 parameter that should be checked and adjusted if not right - "cbox." This should be set to 5, if it is not.

Datapars@ includes many parameters that need to be set. "Fwhmpsf" will vary. This parameter refers to the full width half maximum (FWHM) of the point spread function (PSF) for the object. The PSF essentially refers to the distribution of pixel values that belong to the object. This is what is represented in the radial profile feature of *imexamine*. To find the FWHM for the parameter, use *imexamine* on a few well focused images of standard stars. Note the peak of the profile and divide it in two. Find the radial value of the curve at that half maximum, and multiply it by two to find the diameter, or the full width of the curve. This procedure is further explained below. "Ccdread" should be blank. "Gain" should be labelled 'GAIN', "readnoise" should be set to 6, "epadu" to 'INDEF', "exposure" should be 'EXPTIME', "airmass" AIRMASS, "filter" should be blank. "Obstime" should be set to 'UT' and "itime" to 'INDEF'.



Above is an example of a radial profile of a standard star. To find the FWHM, we first estimate the curves maximum to be approximately at a pixel value of 22500. From here, we find its half maximum to be approximately 11250. Then, tracing that pixel value over to the function and down, we find the radius for that pixel value on the curve to be about 2.9. Now to find the full width half maximum, we multiply the radius, which is the half width of the curve, by two and get a FWHM of about 5.8. To estimate the final value to set the parameter at, do this for a sample of good size and spread of the other standard stars you are using. Precision is not terribly important here, though be sure to round up in your final estimation. A larger than actual value here is much better than a smaller than actual value.

Fitskypars@ has 2 parameters to be set - "annulus" and "dannulus". These control the radius and width of an annulus centered on the object used to calculate the sky value. All of the pixels in the annulus will be averaged to find this value, except for outliers from bright objects. The default should not necessarily be used, and the size of each varies depending on the images. To determine the size of the annulus, two limiting factors must be considered. If the annulus is too small, there may be some interference from the object. On the other hand, if it is too large and becomes a significant portion of the image, it will start to include large scale gradients and such into the sky value, which should be a local factor. Also, be careful with crowded fields. If there are many stars within the annulus region, they may be too many to be averaged out, and could affect the sky value. The centroid method that this algorithm uses is generally good at averaging out stars, though it is not perfect. Crowded fields in your data images may be something you will want to make note of to Dr. Fernández.

Photpars@ has 2 more parameters to be set, "apertures" and "zmag," which also should not

necessarily be left as default. "Zmag" should be set to 0. The “apertures” parameter essentially specifies how wide a region the *phot* task will count pixel values as part of the target object. Determining what this parameter should be set to is done by examining the radial profile of the target objects, using *imexamine*, and trying to spot where it approaches zero. While it never will really reach zero, there is a point where background noise surpasses the light from the object in prominence. In setting the aperture, there should be as little light from the object let out as possible, and as little light from noise let in as possible. Examine the background noise level near the standard stars and objects as well, or if any are in crowded fields. This parameter may be set to multiple values, and *phot* will measure the object using each of the apertures specified. It will likely be good to pick several aperture values to set the parameter to, so that error in this determination will not result in using the *phot* task on every image again, and because you may end up using this one list to measure objects of many various shapes, sizes and intensities and in many various surroundings. The particular aperture out of all those you list now that will be used in the final analysis is chosen later, and it is much easier to repeat the process from there, and to differentiate for separate object sets from there. It would therefore be good to list a wide spread of aperture values in this parameter, listing them with commas and without spaces (e.g. '5,7,12,16,20'). Dr. Fernández should again be consulted here though, for further explanation and guidance, as it is an important part of the process.

Phot has its own embedded parameters as well. None of these need to be edited. "Image" is the name of the image to be used, but that may be specified when the task is executed. "Output" has a default as 'image.mag.?', where image is the name of the file operated on, and the '?' is the number of files have been created for that image with *phot*. This may be changed, though it is simplest just to use this default, and this manual will assume the default is used.

4.2.2. Location, examination, and measurement

With the parameters set, we will continue to acquire the magnitudes. This will of course, require finding the objects in the images. This is done in various ways, depending on the type of object. For stars, charts must be used. Standard stars for example, may be found using the charts in Landolt's paper. Using the standard star name, the specific chart may be found, and from there, the pattern of stars in the chart may be compared with the pattern of stars in the image to find the particular star(s) of interest from the labels on the chart. Other objects, such as comets or asteroids, must be located using other methods. The simplest way of finding them is to display multiple images of the same object taken at different times in the *ds9* viewer. Find the star pattern in each, pick a star, and center each image on it to move out of the reference frame of the CCDs and into the reference frame of the stellar backdrop. Using the blink command, cycle through the frames and look for the object that moves through the rest of the field and has a nonzero velocity relative to the star pattern. That should be it, though check to make sure there aren't multiple moving objects and you have found the right one. If it is helpful, left clicking on the object will create a circle around it to mark it.

Once the objects in the images are located, we may move on to examining them. It is important

to be careful that the objects being used are good before measuring them and using the data. Sometimes images may be distorted or unusable, and CCDs may become saturated, yielding inaccurate values of object magnitude. Therefore, the images and objects in them should be examined to assure that the date you will be acquiring from them will be reliable. To examine an image, first simply look at it in the *ds9* viewer application. Is the image quality poor? Are the objects streaked or doubled? Is the object you will be measuring too close to another object or interfered in some other way? If any of these are the case, consider not using that frame. Use your best judgment in deciding whether to include it or not, and if you do, be sure to log anything at all that might end up making the data from that image erroneous.

Next, the objects should be examined to make sure that their pixel values are within a certain acceptable range, and that their radial pixel curves look good. Use the *imexamine* task to do this. A good curve will be something like a bell-shaped curve cut in half at the maximum. An object is generally too bright to be used if it has a significant number of pixels over 25000 counts. If there are a few over, it is possibly still good, though be sure to log that if you do use it. Objects can also be too dim to be used, if the radial curve is so low to the axis that it is practically flat. There is flexibility here as well, and if there is any sort of curvature in the fitted function, it should probably be used. It is good to err on the side of including data, so long that it flaws are logged, since it can always be taken out later. Because of this, it is helpful to keep a detailed log of the image and object examinations. It will be helpful to easily look back on them in the log, as opposed to looking back at every image one by one, once the data has been analyzed and if there are any outliers. Removing them if present may be a good way to get more precise and accurate results.

Once the images have been examined, the *phot* task may be performed to acquire the magnitudes. To execute the *phot* task, simply enter the command

> phot [filename] (e.g. > phot n2068r.fits)

Again, be sure the correct frame is displayed in *ds9*, and align the center of the cursor, which should again be like a blinking annulus now, with the center of the object. Press 'c' 's' 'o' one at a time, in that order. More than one object per frame may be measured, and should be if there are multiple objects to be measured in a frame; it is best to have only one magnitude file used for each image. Nothing should be done in between, simply re-center the cursor on the next object and repeat 'c' 's' 'o'. When measuring multiple objects in one image, be sure to take note of the order in which the objects were measured. It will be important to know this later when compiling the data and corresponding it with the object names. When you have finished all objects in a frame, press 'q' twice to quit *phot*. There will be a new magnitude file created, located in the same directory as the image and with the name 'image.mag.?' (e.g. 'n2068r.fits.mag.1').

You may either do all the exams at once then all the measurements at once, or you may do them both one image at a time, whichever you prefer.

Repeat the process for all of the science frames to create a good magnitude file for as many images as possible. Once you are finished, you may wish to organize the images and magnitude

files into folders corresponding to the different standard star and object groups, since this will likely be easier when continuing to the next step of the analysis process.

4.3. Standard Star Analysis

The last phase of the data processing is done within the IRAF *photcal* package located at *noao.digiphot.photcal*. This phase includes the use of the IRAF tasks *mknobsfile*, *mkconfig*, and *fitparams* to find the coefficients of the magnitude equation, and using the IRAF task *invertfit* to find th absolute magnitudes of the target objects.

To fit the coefficients of the linear magnitude equation, we need multiple data points. These points correspond to the absolute and instrumental magnitudes of the standard stars. To input the many data points that typically come from the set of standard stars used, as well as eventually, the object sets, it is more efficient and easier to work with the data in lists and tables. Because of this, a great deal of this next and last phase of the process involves creating lists and tables, and properly formating them so IRAF may work with them.

4.3.1. Text editor - Making the standard star catalog file

The catalog file will consist of the absolute magnitudes of the standard stars. These will be used in fitting the coefficients of the magnitude equation for use with the target objects, and one should be made for each distinct set of standard stars on different nights and filters, as with all of the other files in this section. The file will be made from another file - 'onlandolt.dat'. (There is a program task in IRAF - *mkcatalog* - that may be used for this, but it is simpler to just edit 'onlandolt.dat'.) This file is taken from the paper "UBVRI Photometric Standard Stars in the Magnitude Range 11.5 < V <16.0 Around the Celestial Equator" by Arlo U. Landolt, that catalogs the standard stars. The file must be edited because it has many stars in it, and we only need those used in the data set. To do this, simply delete all of the lines from the original 'onlandolt.dat' that are for standard stars not used in the data set. The remaining table should have only those standard stars that are used in the data set, and should look something like this:

```
PG1323-086  13:25:39  -08:49:18  13.481 -0.140 -0.681  ...
PG1323-086A 13:25:49  -08:50:24  13.591  0.393 -0.019  ...
PG1323-086C 13:25:50  -08:48:39  14.003  0.707  0.245  ...
PG1323-086B 13:25:50  -08:51:55  13.406  0.761  0.265  ...
PG1633+099  16:35:24  +09:47:45  14.397 -0.192 -0.974  ...
PG1633+099A 16:35:26  +09:47:48  15.256  0.873  0.320  ...
   .        .        .      .     .     .
   .        .        .      .     .     .
   .        .        .      .     .     .
```

Be sure you keep this edited version of 'onlandolt.dat' distinguished from the original version.

4.3.2. Text editor - Making the list of standard star images

This list will be composed of the image file names to be used in the data set (remember to only include those that have good magnitude files), and their corresponding object names. If there are multiple objects in an image, these must be differentiated after the observation file has been made. The number of lines in the image list should match the number of images with good magnitude files. An example of an image list is the following:

```
PG1323-086 : n2030r.fits
PG1323-086 : n2031r.fits
PG1323-086 : n2032r.fits
PG1323-086 : n2034r.fits
PG1633+099 : n2058r.fits
PG1633+099 : n2059r.fits
PG1633+099 : n2060r.fits
   .          .
   .          .
   .          .
```

Although there are multiple objects in these images, those in each image are all in one magnitude file. They will appear separately on the observation file though, and will be distinguished then. This is why it was important to keep track of the order the magnitudes were taken in, because they will appear in that same order, though all with the name given to correspond with the image.

4.3.3. *Mknobsfile* - Making the standard star observation file

The observation file is a table of information concerning the measurements of each standard star in each image. This information includes the instrumental magnitudes, and will be used in fitting the coefficients of the magnitude equation.

There are several parameters to set in *mknobsfile*. "Photfiles" requires the list of magnitude files, acquired from *phot*, that will be concatenated into the observation file. For this it may be helpful to simply enter "*.mag.*" if the default name was used for creating the magnitude files. If you do this though, remember to check that there are no files with ".mag." somewhere in the name that you don't want in the observation file. "Idfilters" refers to the filter band(s) of the images. Since this manual is limiting the fitting procedure to put each filter through one at a time, this parameter can simply be left as 'INDEF'. "Imsets" requires the name of the list of images made with the text editor in the previous step. The "observations" parameter requires the name of the observations file that will be made. "Aperture" requires the specification of the aperture to be used in extracting the magnitudes from the magnitude files, as mentioned earlier in the section about the *photpars@* parameter file. Choose an aperture out of the list you estimate to be suitable to the standard stars, looking at the various radial profiles of the stars, how close other bright objects are, and how high the background noise level is in most of them. The "aperture" parameter should be set to be the number corresponding to the chosen aperture (e.g. '1' for the first in the list, '2' for the second, etc.).

Once these parameters are set, execute the task, and review the created file to check for errors,

and to rename the objects if there were multiples in any individual image. Be sure they are renamed according to the order the magnitudes were taken in *phot*. The order will be the same, with the first measured being the top line of the group, and the last measured being the bottom line of the group. Also, be sure to adjust the column margins so the columns are straight for being read back into IRAF for the *fitparams* task step. Here are some examples:

Before renaming and adjusting column margins

```
# FIELD       FILTER        OTIME AIRMASS  XCENTER   ...

PG1323-086-1    INDEF      6:12:54.6  1.175  1495.110  ...
PG1323-086-2    INDEF      6:12:54.6  1.175   798.621  ...
PG1323-086-3    INDEF      6:12:54.6  1.175   733.458  ...
PG1323-086-4    INDEF      6:12:54.6  1.175   771.602  ...
PG1323-086-1    INDEF      6:15:16.8  1.179  1496.957  ...
PG1323-086-2    INDEF      6:15:16.8  1.179   800.904  ...
PG1323-086-3    INDEF      6:15:16.8  1.179   735.595  ...
PG1323-086-4    INDEF      6:15:16.8  1.179   773.716  ...
PG1323-086-1    INDEF      6:17:15.2  1.182  1518.223  ...
PG1323-086-2    INDEF      6:17:15.2  1.182   821.899  ...
   .          .          .      .      .
   .          .          .      .      .
   .          .          .      .      .
```

After renaming and adjusting

```
# FIELD       FILTER        OTIME AIRMASS  XCENTER  ...

PG1323-086     INDEF      6:12:54.6  1.175  1495.110  ...
PG1323-086A    INDEF      6:12:54.6  1.175   798.621  ...
PG1323-086B    INDEF      6:12:54.6  1.175   733.458  ...
PG1323-086C    INDEF      6:12:54.6  1.175   771.602  ...
PG1323-086     INDEF      6:15:16.8  1.179  1496.957  ...
PG1323-086A    INDEF      6:15:16.8  1.179   800.904  ...
PG1323-086B    INDEF      6:15:16.8  1.179   735.595  ...
PG1323-086C    INDEF      6:15:16.8  1.179   773.716  ...
PG1323-086     INDEF      6:17:15.2  1.182  1518.223  ...
PG1323-086A    INDEF      6:17:15.2  1.182   821.899  ...
   .          .          .      .      .
   .          .          .      .      .
   .          .          .      .      .
```

In addition to the observation file, a second file will be created. This will contain the format description for the observation file, which allows IRAF to read in the observation file and manipulate the data in other tasks.

4.3.4. Mkconfig - Making the standard star configuration file

The configuration file is basically a formatting file. It is used by IRAF to understand the other input in the fitting procedure. It is basically a combination of the format files from some of the lists made above. Using the *mkconfig* task, IRAF will concatenate the two format files for the catalog file and the observation file ('fonlandolt.dat' and e.g. 'fn2.standstars.obs.dat'), together with the file 'tonlandolt.dat'.

The parameters of the *mkconfig* task should be set as follows. "Config" should contain the name of the output configuration file (e.g. 'n2.standstars.cfg'). "Catalog" should be set to use the

'fonlandolt.dat' file. "Observations" should be the name and path of the observation format file (e.g. 'fn2.standstars.obs.dat'). The "transform" parameter should refer to the 'tonlandolt.dat' file.

With the parameters set, execute the task. The output file still needs editing to be complete though, since it will include the magnitude equations for all filters, using inconsistent variables. IRAF will open an editor in *xgterm* for this to be done, though it may be simpler to use a exit and use a text editor instead. To complete the configuration file, remove all magnitude equations except the one that corresponds to the filter used for this set of standard stars. Once this is done, the variables 'm_' and 'X_', where '_' is the filter, must be replaced with 'mINDEF' and 'XINDEF', since 'INDEF' is what the filter has been named in IRAF. If you did not set the "idfilter" parameter of the *mknobsfile* task as 'INDEF', then you should use whatever you set for that parameter in renaming the variables. (For example, if the R filter is being used, delete the equations corresponding to other filters, and replace the variables 'mR' and 'XR' with 'mINDEF' and 'XINDEF', respectively.) Below are examples of what the configuration file should look like.

Before editing

```
#  Declare the new Landolt UBVRI standards catalog variables

catalog

V  4                    # the V magnitude
BV          5                   # the (B-V) color
UB          6           # the (U-B) color
VR          7                   # the (V-R) color
RI 8                    # the (R-I) color
VI9                     # the (V-I) color

error(V)  12            # the V magnitude error
error(BV) 13            # the (B-V) color error
error(UB) 14            # the (U-B) color error
error(VR) 15            # the (V-R) color error
error(RI) 16            # the (R-I) color error
error(VI) 17            # the (V-I) color error
# Declare the observations file variables

observations

TINDEF      3           # time of observation in filter INDEF
XINDEF      4           # airmass in filter INDEF
xINDEF      5           # x coordinate in filter INDEF
yINDEF      6           # y coordinate in filter INDEF
mINDEF      7           # instrumental magnitude in filter INDEF
error(mINDEF) 8             # magnitude error in filter INDEF

# Sample transformation section for the new Landolt UBVRI system

transformation

fit   u1=0.0, u2=0.65, u3=0.000
const u4=0.0
UFIT : mU = (UB + BV + V)  + u1 + u2 * XU + u3 * UB + u4 * UB * XU

fit   b1=0.0, b2=0.35, b3=0.000
const b4=0.0
BFIT : mB = (BV + V)  + b1 + b2 * XB + b3 * BV + b4 * BV * XB

fit   v1=0.0, v2=0.17, v3=0.000
const v4=0.0
VFIT : mV = V + v1 + v2 * XV + v3 * BV + v4 * BV * XV

fit   r1=0.0, r2=0.08, r3=0.000
```

```
const r4=0.0
RFIT : mR = (V - VR)  + r1 + r2 * XR + r3 * VR + r4 * VR * XR

fit   i1=0.0, i2=0.03, i3=0.000
const i4=0.0
IFIT : mI = (V - VR)  + r1 + r2 * XI + r3 * VI + r4 * VI * XI
```

Tranformation section after editing (using R-filter and the filter ID 'INDEF')

```
   .
   .
   .
# Sample transformation section for the new Landolt UBVRI system

transformation

fit   r1=0.0, r2=0.08, r3=0.000
const r4=0.0
RFIT : mINDEF = (V - VR)  + r1 + r2 * XINDEF + r3 * VR + r4 * VR * XINDEF
```

If your sample size of standard stars is very small, or if there turns out to be a great deal of uncertainty in the fitting process, alter the magnitude equation further by correcting the airmass variable – 'XINDEF' – in the term with the second coefficient, the third term of the right side of the equation. This is done by subtracting the average airmass of the images in the observation file from the airmass variable to bring that term in the equation much closer to zero, essentially shifting the intercept of that variable, and resulting in higher precision. The airmass is the fourth column in the observation file, and may be averaged from all of the elements of that column. Below is an example of the corrected airmass variable:

```
   .
   .
   .
transformation

fit   r1=0.0, r2=0.08, r3=0.000
const r4=0.0
RFIT : mINDEF = (V - VR) + r1 + r2 * (XINDEF - 1.125) + r3 * VR + r4 * VR * XINDEF
```

With the configuration file complete, we may continue now to fit the coefficients of the magnitude equation.

4.3.5. Fitparams - Fitting the coefficients to the standard stars

To determine a magnitude equation for the targets, we must fit the coefficients of the equation to the absolute and instrumental magnitudes of the standard stars. The IRAF task *fitparams* is used for this. Again, this process should be done for each set of standard stars that have a different filter or are on a different night.

The paths and names of the observation, catalog and configuration files should be put into the "observations", "catalogs", and "config" parameters respectively. The "parameters" parameter should have the output file name and path in it. These, again, may also be entered in upon execution of the command.

With all of the parameters set right, execute the *fitparams* task. *Phot* uses *inlfit* - the interactive

nonlinear least squares fitting package – to fit a function and to display the distribution and fit information. A plot window will appear, graphing data points of magnitudes and residuals from a fitted average. The closer to a zero residual value, the more accurate the point is. The cursor will be shown as a red cross, usable for seeing the coordinates of its location on the axes. Above the plot is displayed information about the fit, including the RMS, the equation used, and whether the solution converged or not. Below are some of the basic commands for working in *inlfit*:

c – gives object identity of and information about the data point nearest to the cursor
d – deletes the nearest data point and puts an 'X' at its location; re-fitting will not include deleted points
f – refits the data
q – quits the interactive plot
r – redraws the current plot
u – undeletes the closest deleted data point
w – sets the graph window ranges for both axes
? - displays commands for setting the window
? - displays commands for *inlfit*

There are many other commands, including some in a colon command mode, but many of them are not very useful or important to what we are doing.

Some points may have relatively high residuals, meaning that they are farther in error from the fit. It may be good to remove some of these, to get a better fit. It is difficult to determine when points should be removed, however. It wouldn't be good to remove too many, since every removal shrinks the sample size and brings the fitting closer to simply making numbers up. Leaving bad data points in though, is not good either. Unfortunately, there is no fixed objective way to decide what points to remove while maintaining a balance between these two considerations. You must decide subjectively what to reject and what not to, by weighing several different factors. The first of these is the root mean square (RMS) of the fitting. This is displayed above the plot. The RMS is a good measure of how precise a fitting is. Another important factor to look at and use in the decision is the distribution of the data points – how close they are to each other and to zero residue, how many outliers there are, if any, and how far away they are, etc. Finally, it is good to refer to your magnitude log from the *imexamine* and *phot* tasks portion of the process. With this you can compare any questionable characteristics of images or objects with outliers that might be removed. The 'c' command mentioned above will identify the data points for you so you may use the log. It is a good idea to use the 'c' command to identify all the data points to get a better idea of what the distribution is.

The ideal RMS is about 0.01 or 0.02. If this is the value of the RMS for your fit, do not bother with removing data points. The value will not drop much below this. A larger RMS, say approaching 0.1, or even higher, there may be trouble that could be fixed by removing a few points. An RMS of 'INDEF' indicates that there was no convergent solution found. Again, one way to shrink this uncertainty in the fitting is to correct the intercept of the airmass term. Refer back to the end of section 4.3.4. for more details on this process. As a last resort to achieve a convergent solution, you may increase the tolerance of the fitting by changing the “tolerance” parameter in *fitparams* (e.g. from 3E-5 to 7E-5). Again, bad data points may be removed to

lower the RMS and to seek a more accurate solution. Determining what data points are bad is a difficult task though, and is further explained below.

The distribution of data points is crucial to the rejecting decision, yet again there is no objective rule. There are only helpful things to consider and look at in deciding. There may be a wide spread or a narrow spread. There may be outliers, or no outliers. Outliers may be in a pattern, or they may not. There may be a large number of data points, there may be a small number of data points. All of these characteristics of a distribution should be considered when looking at it.

As said above, if the fit has a close to ideal RMS, you should not bother to reject any of the points, even if there is a wide spread, or other aspects of the distribution look wrong. At the same time, if there are a small number outliers that are in a distribution of much higher number and that are seen to have been questionable in the log, it may be good to remove them, even with an already low RMS.

If the RMS is higher, and there is a large number of narrowly arranged points with only a few outliers, it seems obvious that they should be rejected. A look at the log, again, may help confirm or deny this.

For a small number of points, a wide distribution or the presence of outliers might have to go unchanged, even with a high RMS, to avoid cutting the sample size too much.

If the distribution is wide, but there are no obvious outliers, there may again be nothing to be done, even with a high RMS. The log may point to a few objects that show up on the fringes, though you still should be careful when removing them.

If there are several outliers that are from the same object, it may indicate something that is wrong with the object, not necessarily the images themselves. In this case, it may be good to remove them, especially if all of the data points from that object are outliers.

(example distributions and fittings, will add images later)

If you suspect that removing some points may be good, there are several ways to check. From inside *inlfit*, you can delete the points with 'd', and redo the fit with the new smaller data set by pressing 'f'. This will let you know how the new fit will look without permanently altering the data set. Once you are satisfied that removing the points is good, exit *inlfit* and remove the points from the catalog and observation files. Then reset the parameters if necessary (if you made copies of the lists and changed them or if you are keeping the output file from the previous fitting, etc.), and execute the *fitparams* task again. Check again to make sure the fit is good while not having too little data left and/or not improving enough to warrant the removals. Another good thing to check is the coefficients themselves. Try running the fit and saving the parameters file, then viewing it. It would likely be good to show Dr. Fernández the parameters file and review it together, as well as anything notable about the fitting process. When you are satisfied with the fit, save the final parameters file, and move on to the object analysis portion.

4.4. Object Analysis

Now that we have acquired the fitted coefficients for the magnitude equation, we can use that equation to finally find the absolute magnitudes of the objects we are studying by plugging the instrumental magnitudes of the objects into it. Again, we do this with lists, to make working with large numbers of images simple. The end result will be lists of absolute magnitudes, one magnitude corresponding with each image processed, and one list for each distinct object set.

4.4.1. Text editor - Making the object image lists

This step is very similar to the standard star image list, except now the same thing must be created for the object sets. Also, they should look similar to the standard star list. For example:

```
02-EC24 : n2047r.fits
02-EC24 : n2062r.fits
02-DB13 : n2048r.fits
02-CW207 : n2049r.fits
02-CW207 : n2064r.fits
01-CG21 : n2050r.fits
01-CG21 : n2065r.fits
01-DW34 : n2052r.fits
  .       .
  .       .
  .       .
```

A list should be made for each distinct set of object images.

4.4.2. *Mknobsfile* - Making the object observation files

This process is really the same as the first use of this task, with the exception that we are now working with the objects instead of the standard stars, and so we want the object image list in "imsets" (e.g. 'n2.trojans.img'), the object magnitude files in "photfiles"(*.mag.* in the directory of object images e.g. of Trojan asteroids), the name of the filter in "idfilters" (use "INDEF" again here), instead of the information for the standard stars. Using these will result in an object observation file (e.g. 'n2.trojans.obs'), which will then be used to find the absolute magnitudes of the objects. Before doing this however, the aperture parameter must be chosen again. Remember this will vary for each object set, and it will depend on the typical size, shape, and profile of the object(s) in the set of images, as well as the typical environment of background noise and proximity with other objects.

Once the parameters are completed with the object information and settings, execute the task. If

you measured multiple objects per image, be sure to correct the object names. The completed observation file should look something like this:

```
# FIELD       FILTER        OTIME AIRMASS  XCENTER  ...

02-EC24       INDEF      7:34:06.9  1.394   922.100  ...
02-EC24       INDEF      8:53:29.9  1.314   985.691  ...
02-DB13       INDEF      7:41:36.2  1.557   901.610  ...
02-CW207       INDEF      7:49:14.0  1.465   909.281  ...
02-CW207       INDEF      9:08:36.8  1.392   985.692  ...
01-CG21       INDEF      7:57:08.0  1.209   915.872  ...
01-CG21       INDEF      9:16:37.1  1.149   983.671  ...
01-DW34        INDEF      8:14:53.5  1.383   936.818  ...
   .          .          .      .      .
   .          .          .      .      .
   .          .          .      .      .
```

This should be done for each distinct object set you have. Multiple object sets may be used with the same set of standard stars. This means that for the same standard star catalog, observation, configuration, and fitted parameter files (from *fitparams*), multiple object observation files may be put into the magnitude equation through the next step - the *invertfit* process - to acquire the object absolute magnitudes. This holds so long as the object sets are all from the same night as each other and the standard stars, and that they are all of the same filter as the standard stars. For example, the images of a group of asteroids and the images of a comet may both go through inverse fitting to the magnitude equation with coefficients fitted from the same standard stars from the same night and filter, but the images of a comet in one filter may not use the same standard star parameters as the images of the same comet taken on the same night, but in a different filter.

4.4.3. *Invertfit* - Finding the absolute magnitudes

Now that the magnitude equation coefficients are known, from using *fitparams*, all that is left in finding the absolute magnitudes is to plug the instrumental magnitudes into the equation. This is done with the IRAF task *invertfit*. Again, this step should be done for each distinct object set.

The parameters in *invertfit* to be set are "observations" (the object observation file, e.g. 'n2.trojans.obs'), "config" (the standard star configuration file e.g. 'n2.ctio.cfg'), "parameters" (the *fitparams* output file, e.g. 'n2.fitparam.dat'), and "calib" (the output file e.g. 'n2standtrojans').

Before commencing with the *invertfit* task, the configuration file needs further modification. The magnitude equation includes a variable known as the color of the object. It is basically the difference between two filters, and is used in several places in the equation. In *fitparams*, the color is allowed to remain a variable. In the inverse fitting process of IRAF however, the variable prevents the task from finding a solution. It must be replaced by the actual value of the color to be considered as a constant and allow for the equation to be solved. Finding the actual value involves either looking at previous paper written about the object or using multiple filters to solve for it. Dr. Fernández should be consulted about finding that value.

Once you have the color value to put into the magnitude equation, edit the configuration file as demonstrated below:

Before modification

```
  .
  .
  .
transformation

fit   r1=0.0, r2=0.08, r3=0.000
const r4=0.0
RFIT : mINDEF = (V - VR) + r1 + r2 * XINDEF + r3 * VR + r4 * VR * XINDEF
```

After modification

```
  .
  .
  .
transformation

fit   r1=0.0, r2=0.08, r3=0.000
const r4=0.0
RFIT : mINDEF = (V – 0.045) + r1 + r2 * XINDEF + r3 * 0.045 + r4 * 0.045 * XINDEF
```

Now there will only be one unknown value in the equation when *invertfit* tries to solve it. Unfortunately, that value will be the V magnitude (for this example) instead of the R magnitude as desired. This means we will have to use the color again after the *invertfit* gives a solution to find find the actual solution. Be sure to keep track of which configuration file was used for each task.

With the parameters properly set and the configuration properly modified, execute the task. The output of *invertfit* is the final result of the data analysis process - the absolute magnitudes of all of the objects in the analyzed data set and their errors. Below is an example of the output calibrated indices file:

```
# Wed 14:14:01 19-Apr-2006
# List of observations files:
#               Trojans/n2.trojans.obs
# Config:       n2.ctio2.cfg
# Parameters:           n2.fitparam2b.dat
#
# Computed indices for program and standard objects
#
# Columns:
#   1           object id
#   2           V
#   3           error(V)


02-EC24    20.208  0.016
02-EC24    20.204  0.017
02-DB13    19.896  0.016
02-CW207   19.527  0.014
02-CW207   19.896  0.018
01-CG21    20.083  0.013
01-CG21    20.057  0.012
01-DW34    20.401  0.025
  .        .       .
  .        .       .
  .        .       .
```

If you were working with a filter other than that shown on the final list of magnitudes, simply

use the color term to find the correct magnitude. For instance, the images used in the process resulting in the above list were taken in the R filter. The equation solved for the V magnitude, but to find the R magnitude, one would just use the following equation:

$$R = (V - (V - R)) = (V - VR)$$

The errors need not be changed when transforming to the correct filter, since all that is being done is the subtraction of a constant.

Another correction must be made to the final magnitudes – the aperture correction. This takes the form of an added value to each magnitude, and is meant to correct for counts from the object that were outside of the chosen aperture. In most cases, and for our purposes here, this will be a small value and will be a single constant for the magnitudes from all the images in any given night. There is a way to do this in IRAF, but it is very tedious, and will be left out. Again, you will instead let Dr. Fernández know you have gotten here so he may use a different program to find the correction values for each night.

5. Further Analysis and Storage

To finalize the results of the photometry, the magnitudes may be used to do many different things. It may be good to place the data into a spreadsheet, including the corresponding object names, magnitudes, magnitude errors, and observation times to allow for versatility in further analysis. The coefficients determined and used for the magnitude equation are important also, and should be included as well.

Now that the data has been processed and results have been achieved, the final science images and other important files should be transferred off of the computer to storage. These other files include the master bias, the master flat, the magnitude files from *phot*, the image, catalog and observation list files, the configuration file, the fitted coefficients file, the invert fitted object magnitude file, and whatever further analysis files have been made, such as a spreadsheet. Ask Dr. Fernández about how he would like the files to be stored.

Congratulations! You are finished with the photometry data processing.

References

1. Barnes, Jeanette. "A Beginner's Guide to Using IRAF." Central Computer Services, National Optical Astronomy Observatories. Tuscon, Arizona; August 1993.

2. Davis, Lindsey. "Specifications for the Aperture Photometry Package." National Optical Astronomy Observatories. Revised October 1987.

3. Davis, Lindsey E. "A Reference Guide to the IRAF/DAOPHOT Package." IRAF Programming Group, National Optical Astronomy Observatories. Tuscon, Arizona; January 1994.

4. Davis, Lindsey Elspeth. "A User's Guide to the IRAF Apphot Package." National Optical Astronomy Observatories. Revised May 1989.

5. Landolt, Arlo U. "UBVRI Photometric Standard Stars in the Magnitude Range 11.5 < V <16.0 Around the Celestial Equator." The Astronomical Journal, Vol. 104 Num. 1. July 1994. Louisiana State University Observatory, Baton Rouge, Louisiana. Recieved 7 January 1992, revised 6 March 1992.

6. Massey, Phillip. "A User's Guide to CCD Reductions with IRAF." 15 February 1997.

7. Massey, Phillip and Davis, Lindsey E. "A User's Guide to Stellar CCD Photometry with IRAF." April 15, 1992.

8. Wells, Lisa A. "Photometry Using IRAF." February 1994.