CHARACTERIZATION AND ANALYSIS OF THE SITE 1K CCD AS USED IN THE TENAGRA II .81M TELESCOPE

by

Cody R. Short

A senior thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Bachelor of Science

Department of Physics and Astronomy

Brigham Young University

April 2007

BRIGHAM YOUNG UNIVERSITY

DEPARTMENT APPROVAL

of a senior thesis submitted by

Cody R. Short

This thesis has been reviewed by the research advisor, research coordinator, and department chair and has been found to be satisfactory.

Date Michael D. Joner, Advisor

Date Eric G. Hintz, Research Coordinator

Date Scott D. Sommerfeldt, Department Chair

ABSTRACT

CHARACTERIZATION AND ANALYSIS OF THE SITE 1K CCD AS USED IN THE TENAGRA II .81M TELESCOPE

Cody R. Short

Department of Physics and Astronomy

Senior Thesis

The Astronomy Group of Brigham Young University has been actively involved in acquiring astronomical data from the Tenagra Remote Observatory located in the Sonoran Desert of southern Arizona. From September 17, 2006 through January 31, 2007 this data has undergone a holistic review to characterize the performance of a single CCD employed in the Tenagra II telescope's imaging system.

Several of the CCD's main characteristics have been established. The CCD gain and readnoise were found to be approximately 4 e^-/ADU and 29 e^- respectively. The CCD's response linearity has been confirmed and the dark current examined. The CCD has been examined for bad pixels and some results are included. Temperature stability of the CCD has also been evaluated and found to be consistent within ±1%.

The results of this review, presented herein, will be employed by faculty and students to lead to the best possible calibration procedures for the raw data acquired from the Tenagra II telescope. The results will also be used to aid in the process of calibrating and evaluating CCD detectors that are used in future investigations.

ACKNOWLEDGMENTS

Professor Michael D. Joner has been an invaluable resource for ideas and general knowledge. Without his guidance and expertise this project would never have happened. Dr. Eric Hintz's assistance with questions and logistics has brought this thesis to reality. Craig Swenson and Jake Albretsen's work within the Tenagra group obviated hours of extra work on the part of the author and is appreciated. IRAF scripts written by SummerDale Beckstrand and Eran Ofek also helped to cut out a lot of tedious work in this project. These contributions are also appreciated.

Lisa Joner and Mike Fogarty have provided valuable assistance with editing this document, and I am grateful to them.

I would like to acknowledge my colleagues Roo Phillips and Paul Iverson, who have been friends throughout my astronomy experience at BYU. Just knowing there were people to turn to for assistance when I was stuck with an assignment or for other reasons has been reassuring.

I acknowledge the stars that stirred my soul and brought me to this point. I acknowledge the Creator of those stars for everything that I am.

This thesis and my extensive efforts in this project are dedicated to my wife Lori and our son Jackson, who should arrive in time for graduate school. Without Lori's love and support and the knowledge that my work will in some way benefit our family, this thesis would have been exponentially more difficult to complete.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Background

In this chapter the operating principles of CCDs are discussed. The motivations for characterizing the CCD are established, and the particular CCD of study is introduced. This leads into Chapter 2 where the procedures for this study are outlined. Chapter 3 includes the analysis and results of the data and the 4th and final chapter presents the conclusions of the study and recommendations for further study. Appendices containing applicable code and some specific parameters are included after the main body.

1.1 Principles of CCDs

A charge-coupled device, or CCD, is effectively a matrix of light sensitive pixels or wells which can fill up with light. The light incident on the active sensing area of a CCD is perceived as a stream of individual light particles called photons. Each photon, upon striking a pixel of the CCD, is translated into an equivalent amount of charge, or ADUs (analog to digital units). Charge fills up the well associated with the pixel where the photon struck the CCD.

A light source which strikes a certain part of the CCD will fill up that section's pixel wells to a degree corresponding to the effective brightness of the source. After the pixel wells are filled up to their relative levels and the CCD has been removed from exposure to the light source, the count level of charge in the wells is read out by the imaging system operating the CCD. This imaging system is usually a digital camera connected to a computer. By comparing the amount of charge in each CCD well, the relative brightness of objects on the frame can be determined.

CCDs are popular detectors possessing relatively high quantum efficiencies (QEs). Quantum efficiency is a measure of how much light is detected relative to how

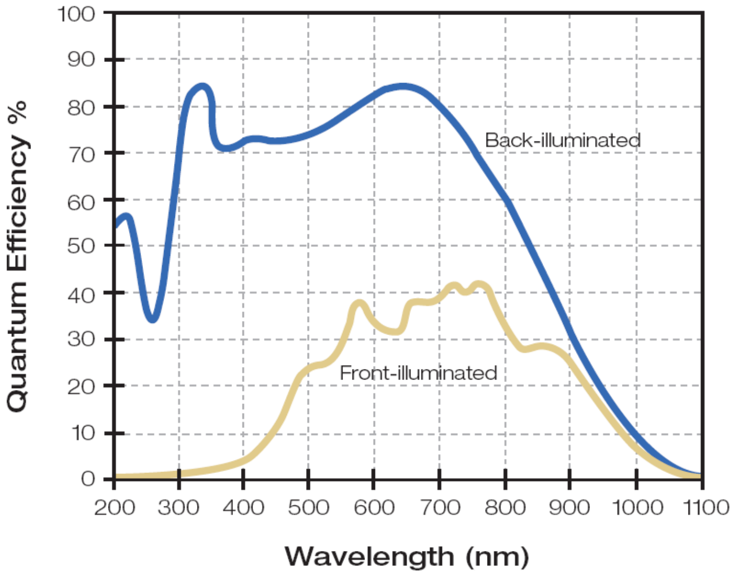


Figure 1.1: Quantum Efficiencies Characteristic of the SITe 1K CCD (SITe 2003).

much light actually strikes the detector. If a detector has a QE of 50%, this means that it only registers 50% of the light that actually hits it. A publication by Scientific Imaging Technologies shows that some CCDs have QEs that can range from 0% to 90%, depending on the wavelength of incident light and the design of the CCD (SITe 2003). A graph of quantum efficiencies taken from the that publication can be found in Figure 1.1.

Another positive feature of CCDs is their generally good resolution. Resolution is a measure of how well a detector can distinguish between adjacent objects. For example, two stars extremely distant from each other could appear to be right on top of each other in a two-dimensional image. This is because they lie along the same line of sight from the detector. How close these two stars can be to each other, side by side, and still remain distinguishable in the image is a measure of how good the detector's resolution is. The reason why CCDs generally have good resolution is because they are made up of many individual pixels operating together. As the number of pixels on the CCD increases and the size of the pixels decreases, the resolution of the CCD increases. Typical CCDs can have anywhere from a quarter of a million pixels to upwards of several million pixels on their active imaging area.

Inherent to CCDs are some effects that must be considered when using a CCD based detector. First, the process of gathering the light at the CCD creates instrumental noise that affects the quality of the information. A more significant form of instrumental noise is a result of the CCD readout process. As the information contained in a pixel well is transferred to the imaging system, it has to go through a serial transfer process. For example, a pixel on the far side of the CCD from the readout system cannot transfer its information until the pixel in front of it is emptied. This far pixel then transfers its information into the recently vacated adjacent pixel. This process continues until it has traversed the whole length of the CCD. This serial transfer process is similar to the transfer of water by a fire bucket brigade. Professor Joner pointed out that this charge transfer process has an associated efficiency, known as the charge transfer efficiency (CTE). The CTE for a CCD is as high as .999999 for some detectors (Joner 2007). The noise introduced in the signal during this process is called readout noise, or simply readnoise.

Another effect evident in CCD imaging is dark current. This effect is a result of ambient thermal energy being interpreted by the CCD as incident signal. To correct for this effect, most CCDs are operated at very low temperatures. Even at low temperatures, however, when the CCD is exposed to a null field (e.g., it takes a picture of the camera shutter) for any length of time, some dark current will register on the CCD. When an actual image is taken with the shutter open, the dark current is still there underlying the image.

Finally, an effect related to the pixel wells themselves is somewhat reminiscent of water tension. When you empty a glass of water you can never get the last few drops to come out because they stick to the glass. A CCD's pixel wells experience a similar effect referred to as signal bias. Essentially the bias on the signal is that last little bit of charge that can never really be emptied out of the well because it effectively sticks inside. According to Professor Joner, this can also be viewed as the charge left on the CCD to allow it to operate and properly transfer the signal (Joner 2007).

1.2 CCDs in Astronomical Work

Astronomy has been conducted in many ways for thousands of years. In its earliest form, people relied on only their eyes to see what the night sky had to offer. The human eye possesses a tremendous ability to perceive. However, this ability is overshadowed by the immensity of what exists to be seen. Many forms of astronomical detectors have been developed in response to the eye's limitations. In part, this is why the CCD has been developed. In fact, according to Janesick and Blouke, pioneering CCD developers, "Astronomers were among the first to recognize the extraordinary imaging capabilities of the CCD." (Janesick & Blouke 1987)

In recent years the CCD has become one of the primary tools of astronomy. Used in conjunction with telescopes and science grade cameras, CCDs have vastly increased the astronomer's ability to resolve adjacent objects, quantitatively measure the characteristics of those objects and understand their significance.

Due to the nature of the CCD's operation of converting light into a discrete digital signal, astronomers can look at how this signal changes for a given source over different lengths of time and establish how the light source is changing. In the case of supernovae and variable stars, the way these changes in brightness or other characteristics occur can lead to remarkable insights. For example, knowing that a certain type of supernova has a characteristic way in which its brightness changes can lead to an extrapolation of the distance to it. Because these supernovae occur often in other galaxies we can determine the distance to their host galaxies. These logical steps, based in a fundamental way on how the brightness of some distant star changes, have led to estimates for the size of the Universe.

Other astronomical applications of CCDs include creating detailed pictures of the structure of nebulae or other objects. By employing different types of filters which

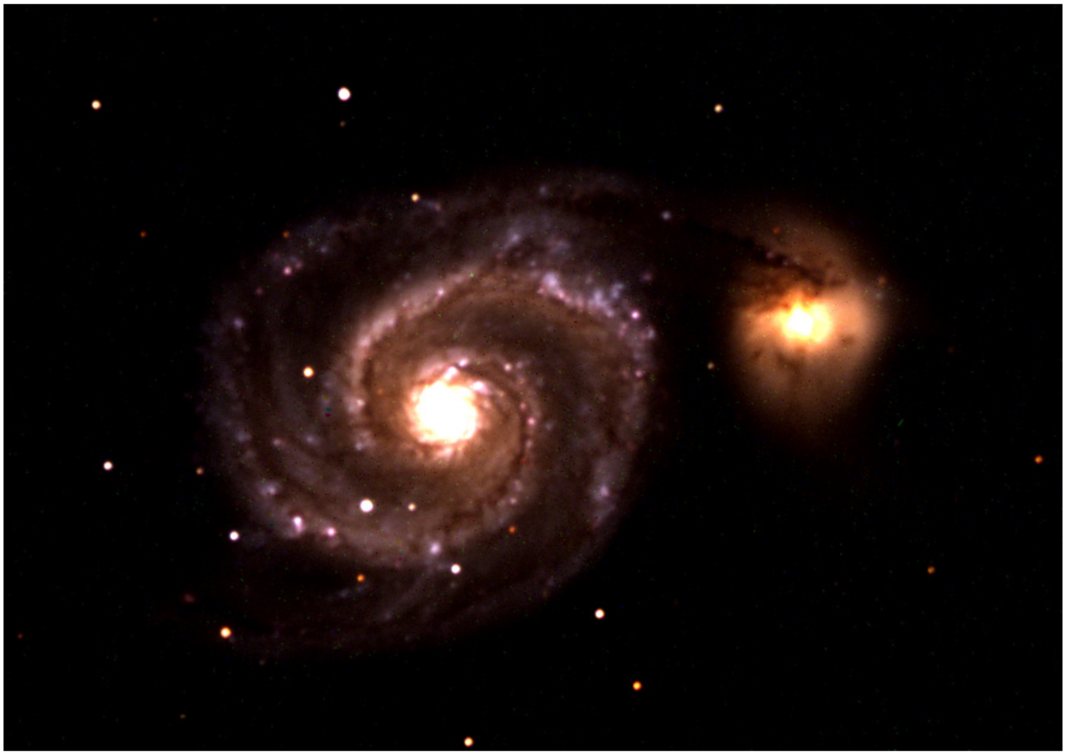


Figure 1.2: M51 Three-Color Image from 300-Second B, V and R Exposures (One of Each).

allow only certain types of light to strike the CCD, composite images can be made that lead to better understanding of the composition of different types of astronomical objects. For example, a filter which allows light emissions from a certain element in a gaseous nebula while not allowing others can help astronomers get a better picture of how that particular element plays a role on the cosmic stage. Beautiful three-color images are made through this process by combining light from different exposures taken in different filters. The different exposures from the different filters are assigned a color value. When these exposures are combined, a full-color effect like that seen in Figures 1.2 and 1.3 results. These figures were created by the author from Tenagra data, using separate exposures secured through standard red, green and blue (R, V and B) filters.

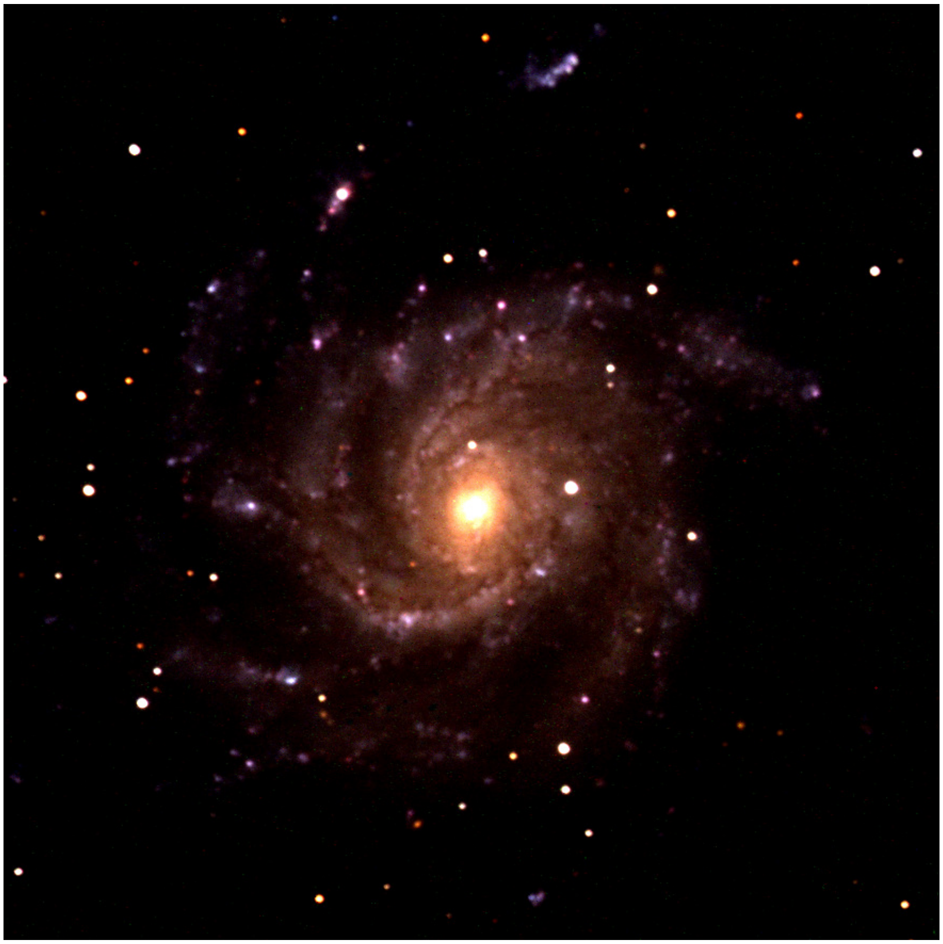


Figure 1.3: M101 Three-Color Image from 300-Second B, V and R Exposures (One of Each).

1.3 Motivations for Characterizing the CCD

The quality of scientific research in any form is only as good as the data it is based on. The data acquired by means of a CCD suffers from several problems inherent to CCD technology. Corrections for these problems must be made before the data can be interpreted to form useful conclusions. Several fundamental characteristics unique to a CCD must be established to determine how these characteristics affect the data. Each characteristic of interest in this study and its accompanying motivation for being established is explained in detail below.

1.3.1 CCD Gain and Readnoise

CCD gain is the number of electrons that constitute a single ADU. It is different for every detector, due to the design and performance of the amplifiers used to read the CCD. This gain is important as it relates to the readnoise of the detector. Noise is usually measured in multiples of electrons. If the noise is significant, it can affect the quality of the image. Determining the significance of the readnoise depends on the gain of the detector. For example, if the gain of the CCD is 4 (e.g., 4 electrons per ADU) and the readnoise is 5 to 10 electrons, it becomes difficult to distinguish actual pixel value differences on small scales.

Due to these effects, it is important to know both the gain and the readnoise for the CCD in question. Knowing this information can help to better quantify the results of the data. In many cases the readnoise levels can be subtracted from the signal or they can simply be used to help quantify the error in the data.

1.3.2 Linearity of the CCD Response

CCDs are promoted as being extremely linear detectors. If you measure two objects with a CCD and the second object is physically twice as bright as the first, it should appear as such in your measurements. CCDs are linear within specific parameters. Their linearity stems from the fact that their basic function is to measure light as it arrives on the CCD. However, at some point their capacity to measure runs out as the individual pixel wells fill up or become saturated. It is at about this point that the CCD's response to light stops being linear. It is important to know at what point the CCD's response is no longer linear so data taken beyond these limits is not used.

1.3.3 Dark Current and Temperature Stability

As mentioned previously, dark current is a measure of extraneous signal resulting from thermal effects on the CCD. Essentially, the CCD's readings are generally higher than expected because of the dark current in the background. If the amount of dark current can be measured, it can be subtracted from the images before analyzing and drawing conclusions from the images.

Dark current is another aspect of the CCD that should behave linearly. If the CCD is exposed for several different time intervals, the dark current should be proportional to the exposure length. An exposure twice as long as another should have twice as much signal resulting from dark current. The linearity of the dark current is important because once it is established it becomes a simple matter to scale the dark current to match your exposure length. If an exposure is taken for a given length of time, and the dark current level for that length of time is unknown, an appropriately scaled, known dark value for another length of time can be substituted. This is why it is important to determine the linearity of the dark current.

Since the dark current is a function of temperature, it is important to cool the CCD to a level where the dark current is manageable. This can be from 0°C to ~ -200°C (the lower value being approximately the case for liquid nitrogen cooled systems). Once an operating temperature is set for the CCD, it is essential that this temperature be maintained constantly while data is being gathered. Fluctuations in temperature will cause varying levels of dark current and can result in greater errors in the data, as well as difficulty in interpreting the data. Understanding how the temperature fluctuates on the CCD can help researchers understand and correct for inconsistencies in their data.

1.3.4 Bad Pixel Analysis

Each CCD is unique. Quirks and peculiarities of one CCD will not exist in another CCD in the same way. As a CCD is composed of millions of pixels, there exists a great statistical probability that a number of those pixels will respond differently to light. Some may not respond at all. The process of acquiring light and transferring it to the imaging system can also bring out pixel to pixel errors in a CCD. A number of different types of "bad" pixels may exist in a given CCD. Pixels that respond too much to light may exist as a result of a misshapen electron well. Pixels that give little or no response may exist for the same reason. Whatever the reason for their existence, bad pixels introduce errors into the data acquired by a CCD. If these bad pixels can be isolated and adjusted for, their effect on the data can be reduced or eliminated. Therefore, it is important to determine what pixels are bad for a given CCD.

All of these various CCD characteristics play an important part in one way or another in the data acquired by a given CCD. Because any one of them can adversely affect the results of countless hours of study leading to inaccurate conclusions, it is essential that each of them be understood in the most comprehensive way possible. This is the underlying motivation for the ensuing analysis of the Tenagra II SITe 1K CCD.

1.4 Introducing the SITe 1K CCD of Tenagra II Fame

The SITe 1K CCD is manufactured by Scientific Imaging Technologies or SITe. The technical data sheet provided by SITe describes this line of CCDs as follows:

"The SITe® SI03xA family of 24 μm Charge-Coupled Device (CCD) image sensors are full-frame, 100% fill-factor devices intended for use in moderate-resolution scientific, commercial, and industrial applications where high dynamic range, broad spectral sensitivity, high quantum efficiency, and low noise are required." (SITe 2003)

First, 24 micrometers (μm) is the length and width of a single pixel on the CCD. This is important because it plays heavily in the resolution of the CCD. The smaller the pixel, the better the resolution of the CCD. However, as the pixel size gets smaller complications can arise in the data transfer process. The resolution per pixel of this CCD is approximately $0.87''$ ($60''$ being an arcminute denoted $'$) as indicated by the Tenagra Observatories website (Tenagra 2007). This yields a field of view measurable by the CCD of approximately $14.8'$ by $14.8'$ ($60'$ is one angular degree of sky). The CCD's active sensing area is 1,024 pixels by 1,024 pixels, as denoted by the 1K designator in its name. This means that there are just over a million pixels on the surface of the CCD. The terms "full-frame" and "100% fill-factor" indicate that the CCD images over the entire active sensing area. "Dynamic range" is a measure of the device's capacity to measure. "High dynamic range" means that this CCD should be able to measure small values as well as significantly large values.

This CCD is a back-illuminated CCD, which means that it has a thin substrate and its back side is illuminated when exposed, as opposed to the more traditional configuration in which the front side is illuminated. Back-illumination is good because typical CCDs have their readout electronics integrated on the front side of the chip. Illuminating the back side avoids this potentially problematic configuration. However, the thinness of the substrate significantly increases the bias level of the CCD, which is less desirable.

Chapter 2

Methodology

In this chapter the data source is described and procedures for acquiring the data are outlined. The specific procedures for examining each of the characteristics of interest are then given.

2.1 The Data Source

The data for this analysis were gathered from general astronomical data retrieved from the Tenagra II telescope. These data sets were acquired by the observatory in agreement with the BYU Astronomy Group for various research purposes by the group. Michael Joner of the BYU Astronomy Group provided observation requests to the observatory, and these requests were serviced by the observatory's automated observing routines. The BYU observation schedule began on September 17, 2006 and has continued on a four-night rotation, with BYU observing for two nights and other Tenagra patrons observing on the off nights. Some nights, little or no data were returned due to observatory technical difficulties or weather conditions.

The Tenagra routine for gathering data involved taking a series of calibration frames to begin the night's run. The very first frames acquired each night were bias and dark frames, zero length exposures and exposures taken with the shutter closed, as explained in further detail in the motivations section of the previous chapter. Next, the telescope would take several series of flat calibration frames in different *Johnson*, *Cousins* and N and W filters. Until December 5, 2006 the flat series were taken in the following sequence: V filter flats and then R filter flats at the beginning of the night, then I, B, U, W and N filter flats at the end of the night after the actual images were taken. After December 5, 2006 the flat sequence was modified at the request of Tenagra patrons to more effectively acquire flats. This sequence started with W, V and I filter flats taken at the beginning of the night. R, B, U and N filter flats were taken after the completion of the program frames. The program images requested and coordinated by Professor Joner were taken in the dark hours between the two series of flats.

By inspecting, analyzing and manipulating this data in ways specifically designed for this analysis, the characteristics of the CCD have been isolated. Specific procedures for acquiring the data and the analysis of the data follow.

2.2 Downloading, Organization and Automation

Preliminary steps for analysis of the CCD required obtaining workable data from the Tenagra stream. All data from Tenagra were posted to an FTP site. Each day, following a night's observing run the data sets were downloaded by the author to a central server location at BYU. This process was later automated to the point of only requiring monitoring by the author. The routines for automating the downloading process were written by Jake Albretsen and have proven very helpful. These routines are included in Appendix A.1.4.

After downloading the data, some housekeeping tasks on the files needed to be performed. This included moving files into a standard directory structure, deleting extraneous files from Tenagra, updating the header keywords on the images and cataloging each night's data. These steps were also initially completed manually by the author. A set of batch scripts written in the IRAF command language by both the author and Professor Joner were implemented towards the end of 2006 to facilitate this process. The master script serves to catalog a night's worth of data using an IRAF routine called *ccdlist*, reorganizes and cleans up the data, calls header fixing scripts authored by Professor Joner, and then finally runs another IRAF routine called *imstatistics* to generate statistics for each image. The output of the *ccdlist* and *imstatistics* routines are then output to two text files called oblist.txt and imstat.txt in the night's directory. These scripts are included in Appendix A.1.

2.3 Procedures

Each characteristic of the CCD was examined independently. Each of the following analyses followed a similar outline. First, the specific data sets taken by the Tenagra II telescope that would yield applicable results were set aside for analysis. Next, any preparations for the data were performed using the Image Reduction and Analysis Facility (IRAF). After the data sets were prepared, specific IRAF routines were used to manipulate and analyze the data. The results were then consolidated, tabulated and interpreted in Microsoft Excel to determine their significance. Each specific analysis is discussed presently.

2.3.1 CCD Gain and Readnoise

Establishing the detector gain and readnoise was accomplished using flat and bias frames from several nights. These nights consisted of ten evenly spaced nights from November 11, 2006 to January 31, 2007. (September, October and November frames, up to the 9th of November, were excluded from this study because the flat frames from this time period were processed by Tenagra Observatories' automated routines, thus compromising the viability of these frames for this analysis.) The first four bias frames and the first six flat frames of each filter were retained from each night. If six frames did not exist for a given filter for a given night, all of the frames for the filter were retained. No less than four flat frames were retained for each filter. This number of frames over several nights was sufficient to yield a representative sample while minimizing the processing time required to produce results.

These frames were prepared by trimming the outer ten pixels off of every side of every frame. This was done because this area commonly contains blank columns or rows. These blank columns and rows held the potential to compromise the results of the data. The frames were then renamed using automated renaming scripts prepared by the author. Finally, five 20 x 20 pixel regions were selected on the CCD area for analysis. This was done to check for gain and noise consistency across the surface of the CCD while also reducing processing time. The five representative sections included the center of each of the four quadrants of the CCD and the center of the CCD itself. Pixel regions for these sections in [x1:x2,y1:y2] format are: [758:777,246:265] - upper right quadrant, [246:265,246:265] - upper left quadrant, [246:265,758:777] - lower left quadrant, [758:777,758:777] - lower right quadrant, and [502:521,502:521] - center section.

The IRAF task called *findgain* was used to compute the gain and readnoise of each representative section of the CCD. *Findgain* requires two unprocessed flat frames of the same filter and two bias frames as its input. Each flat frame from each filter was used with every other flat frame from that filter, along with each bias frame. All permutations for each filter and section were computed. This was done by using a master script to step through each night, filter and section of the CCD. This master script utilized specialized gain and readnoise scripts written in Perl by SummerDale Beckstrand. These scripts take each bias and each flat in a directory and use them as input for *findgain*. Extensive gratitude is extended to SummerDale for allowing the use of her scripts for this procedure, which greatly facilitated the work. These scripts are included, with permission from SummerDale, in Appendix A.2.

The results for each of the five representative sections of all seven filters for each night were then output to text files. These results included gain, readnoise and their respective error values for each section. These four values were gathered for all five sections of all seven filters, yielding 140 data points per night over 10 nights - a total of 1400 data points. The data contained in the text file output were imported into Microsoft Excel and then analyzed. The analysis and results are found in Section 3.1.

2.3.2 Linearity of the CCD Response

As noted previously, one of the most important features of a CCD is its linear behavior. It "sees" things in a manner that closely resembles how they actually are.

Essentially, a star or other feature on a frame that is twice as bright as another is actually measured as twice as bright by the CCD. This linearity is very important when comparing data because non-linear behavior would lead to erroneous conclusions.

Dr. Eric Hintz requested frames of NGC 225 as well as several other clusters for his research programs. The unique thing about these frames is that Dr. Hintz requested the clusters to be exposed for various lengths of time. These various exposure times played very well to the examination of the CCD's linearity, because a star that was imaged by the CCD for 10 seconds and then later exposed in another frame for 45 seconds should be exactly 4.5 times as bright in the second exposure if the linearity of the CCD held. In order to establish the linearity of this CCD, the following experiment was performed.

All of the NGC 225 frames taken in the B and V filters were set aside from December 1st, 10th, 22nd, 25th, 26th, 30th, January 2nd, 3rd, 7th, 10th, 15th, 18th and 23rd. These frames, along with the corresponding calibration frames for these nights, were then processed in the following manner. First, all of the frames had the outside 10 pixels on each edge trimmed off, as in the gain and readnoise study. Second, all of the bias frames were combined and the representative bias frame was subtracted from the dark, flat and image frames. Third, the dark frames were combined and the representative dark frame was subtracted from all of the flat and image frames. Next, flats for the B filter and V filter were combined and divided into the images taken in the B and V filters. Finally, the frames for each night were aligned to each other using Eran Ofek's autoalign.cl script. This script employs the IRAF task $daofind$ to find all the stars in a given frame and make coordinate files for that frame. See Appendix A.3 for the alignment scripts, included with permission from Eran (Ofek 2007).

After the frames were all processed and aligned, they were examined to establish how these processes affected them. After verifying that the frames were qualitatively sound for the purposes of this study, a process called aperture photometry was employed to establish the brightness of each star in a 10-star ensemble. This

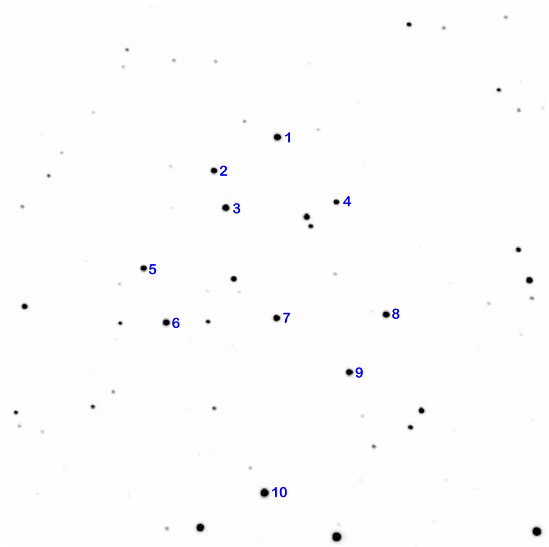


Figure 2.1: NGC 225 10 Star Ensemble for Photometric Study (Numbered by the Author, 14.8″ Tenagra Frame, North is to the Left).

10-star ensemble is shown in Figure 2.1. Aperture photometry is essentially a process of drawing a circle around a star and adding up all of the counts in that circle. Then a circle is drawn around the aperture, creating an annulus (ring) of background sky which allows for adding up the sky count. The sky count per pixel is subtracted from the star count per pixel, leaving a value from which IRAF can generate a magnitude for the star. This process of aperture photometry was used to compare the brightness of star images taken with the CCD for different lengths of time. Each of the ten stars in the ensemble were selected for their roughly equivalent diameters and spacial isolation, thus allowing a single radius for the aperture and preventing contamination from adjacent star light. This also allowed for a single annulus value. The applicable photometric parameters for the IRAF tasks involved in this study are included in Appendix B.

The analysis and results of the photometric linearity study of NGC 225 are included in Section 3.2.

2.3.3 Dark Current and Temperature Stability

Dark frames are generally taken in order to account for the dark current in the detector. These frames are subtracted from the flats and images to remove the dark current count and repeatedly "hot" pixels from the flats and images. Dark current is a manifestation of thermal noise on the detector. The CCD actually counts the ambient temperature around the CCD as noise. Therefore, the temperature stability and dark current are inherently linked, and characterizing and monitoring these aspects of the CCD is essential. Three separate analyses were performed to establish these characteristics. Each will be described below.

Dark Current Linearity Analysis 1

Professor Michael Joner requested that several additional dark and bias frames be taken as part of the regular observing schedule each night. The dark frames were exposures of 150 seconds and 300 seconds, taken at roughly equivalent intervals throughout the night. These frames were used to see how the dark current from a 300-second exposure compared to that of a 150-second exposure. This analysis was similar in form to the photometry linearity study, but did not involve any photometry.

First, each of the 300-second darks, 150-second darks and the bias frames were set aside for every night that they were taken throughout the time period starting with October 19, 2006 and continuing through January 15, 2007. Second, these frames were all trimmed to exclude the outer ten pixels of each frame. Third, the bias frames were combined for each night and the resulting representative bias frame was subtracted from all of the dark frames. Next, fairly intricate scripting procedures were developed to rename each of the dark frames with a number. Odd numbers were given to 300-second exposures and even numbers were given to 150-second exposures. These scripts used the IRAF task *imrename* to rename the images according to the author's design.

Scripts that called the *imarithmetic* task in IRAF were used to divide each 300-second dark by each 150-second dark. All permutations of this operation were performed. Additionally, each 150-second dark was divided by each 150-second dark. Also, each 300-second dark was divided by each 300-second dark. The *imstatistics* task was called to generate statistics from each of the various quotient frames. It was expected that the 300-second darks divided by the 150-second darks would yield a mean pixel value of 2, while the 300 by 300 and 150 by 150 mean pixel values should both be 1. This would establish the constancy and linearity of the dark current.

This analysis proved to be the most involved of all and yielded some varied results. The analysis and its results are included in Section 3.3.1. Also, the scripts involved in this process are included in Appendix A.4 for reference.

Dark Current Linearity Analysis 2

Special sequences were run during the daytime on February 13th and 14th, 2007 to help further characterize the dark current of the CCD. These sequences involved taking several iterations of 0, 30, 60, 120, 180, 240 and 300 second darks (the 0-second darks are bias frames by definition). This iterative routine was run about 22 times each day. All of these frames were set aside and processed in a similar manner to that performed previously. All of the images were trimmed and then the combined 0-second darks were used to bias correct all of the remaining darks.

After the frames were prepared, each set of frames corresponding to a given exposure length was combined to yield a representative frame for that exposure length. These frames were evaluated statistically on their own. Then the frames were multiplied by various constants to yield equivalent exposure lengths for comparison, and statistics were generated for these equivalent exposures. For example, the 120-second representative dark was multiplied in turn by .25, .5, 1.5, 2 and 2.5, yielding equivalent exposure lengths of 30, 60, 180, 240 and 300 seconds. The statistics for these equivalent exposures were analyzed, and their results can be found in Section 3.3.2.

Due to the very sensitive relationship of dark current and temperature, the temperature for these sequences was monitored closely and its analysis is also given in the next chapter. Also noteworthy is the fact that these data sets were taken outside the time frame of the general study period of September 17, 2006 to January 31, 2007.

Temperature Stability Analysis

This analysis was the least analytically involved and resulted in an enormous amount of raw data. The data sets were distilled to yield the high points in the following manner.

First, each frame for every night contains a special header keyword, CCDTEMP, which contains the recorded value of the CCD temperature for that frame. This temperature information was retrieved for every frame for every night and then output to a text file using an IRAF task called *hselect*. These temperature values were retrieved from each night from September 17, 2006 to January 31, 2007. Next, the data were imported into Microsoft Excel and analyzed to produce overall trends of the CCD's temperature, including the average deviation from the CCD's set temperature for a given night. These results are included in graphical format in Section 3.3.3.

2.3.4 Bad Pixel Analysis

Several attempts have been made to produce a general map of bad pixels on the CCD, but each method attempted has provided inconclusive results. Some attempted methods are described here.

The IRAF task *ccdmask* can be used to construct a bad pixel map of the CCD. The task calls for dome flats of both high counts and relatively low counts. The high count flat is divided by the low count flat, and this resulting ratio is used as the input for the *ccdmask* task. The *ccdmask* task then outputs a map of bad pixels. Unfortunately, the Tenagra II telescope is not configured to take dome flats. As a result, other combinations of frames have been used to create bad pixel maps.

Initially, dark frames were used as the input for the *ccdmask* task. When the results of this attempt proved inconclusive, different combinations of flats were used as input. These results were also somewhat inconclusive. The analysis and results are found in Section 3.4.

Chapter 3

Analysis and Results

In this chapter the analyses and results for each individual experiment are presented.

3.1 CCD Gain and Readnoise

Given the large amount of data generated by the gain and readnoise runs, the following steps were taken to get a better picture of these actual values. Due to the fixed placement of the sections for study and the constant variation of content on the flat field frames, it is quite likely that stars might be located in the area of consideration. This would generate a large error in the respective output values. Therefore, these values and their accompanying standard deviations were first inspected. Any value that did not seem consistent with the data set (e.g., negative values or extremely large or small values) was thrown out. Generally, 10% or less of each data set was rejected. Also, from visual review of the *I* filter flats, it was determined that stars were consistently present in these flats. The data points for the *I* flats were extremely poor, with a very large spread and very large accompanying errors. For this reason the data points from the *I* flats were discounted.

After this initial review of the data, the spread in the data was still not ideal. A second discriminatory review of the data for points with standard deviations above a certain value was performed. These standard deviation limits were different depending on the filter, because some filter data sets were seen to have better natural convergence. These cutoffs were chosen to minimize error while retaining a significant data set. The σ cutoffs by filter are listed in Table 3.1.

The resulting data were averaged by filter and by quadrant. Errors were also averaged. These results are tabulated in Tables 3.2 and 3.3.

Table 3.1. Sigma Cutoff Values by Filter for Gain and Readnoise

Filter	Gain Max σ	Noise Max σ
B	.4	3
N	.4	3
R	.6	5
U	.5	3
V	.5	4
W	.7	5

Table 3.2. Gain and Readnoise by CCD Section Averaged Over All Filters but I

Quadrant	Gain	Gain Error	Noise	Noise Error
	(e^-/ADU)	(Average σ)	(e^-)	(Average σ)
I	4.0063	0.3651	26.0231	2.5223
II	3.9164	0.3706	24.6778	2.4709
III	4.0411	0.3625	31.0578	2.9929
IV	4.0866	0.3266	33.9220	2.9190
Center	4.1318	0.3729	29.8606	2.8606
Average	4.0364	0.3596	29.0609	2.7531

Table 3.3. Gain and Readnoise by Filter Averaged Over All Quadrants

Filter	Gain	Gain Error	Noise	Noise Error
	(e^-/ADU)	(Average σ)	(e^-)	(Average σ)
B	4.2038	0.2712	30.3171	2.1518
N	4.1654	0.2778	29.9751	2.1893
R	3.8360	0.4578	27.5391	3.4095
U	4.1944	0.2995	30.3488	2.3595
V	4.0315	0.3595	28.9404	2.7698
W	3.7875	0.4915	27.2445	3.6389
Average	4.0364	0.3596	29.0609	2.7531

3.2 Linearity of the CCD Response

The photometric linearity study of NGC 225 yielded some of the best results of the entire project. After obtaining magnitudes for each of the ten ensemble stars for each frame, the magnitudes were compared for different exposure lengths to determine the flux difference based on exposure length. The following logarithmic relationship was used to establish flux difference from the differential magnitudes:

$$\Delta\text{Flux}{=}2.512^{(magfaint-magbright)}$$

The ratios of the 10s B filter exposures in relation to the 120s B filter exposures were determined. The ratios of the 10s V filter exposures in relation to the 45s and 120s V filter exposures were also determined. Finally, the ratios of the 45s V filter exposures to the 120s V filter exposures were established. These flux differences were averaged for each of the ten stars for each night. The results are tabulated in Table 3.4 on the next page. For the first two nights, 120s V filter exposures were not obtained. The values from December 30th and January 15th have been excluded from the 120s vs. 10s V filter results. These values seemed uncharacteristically low, possibly due to bad seeing conditions or saturated pixels.

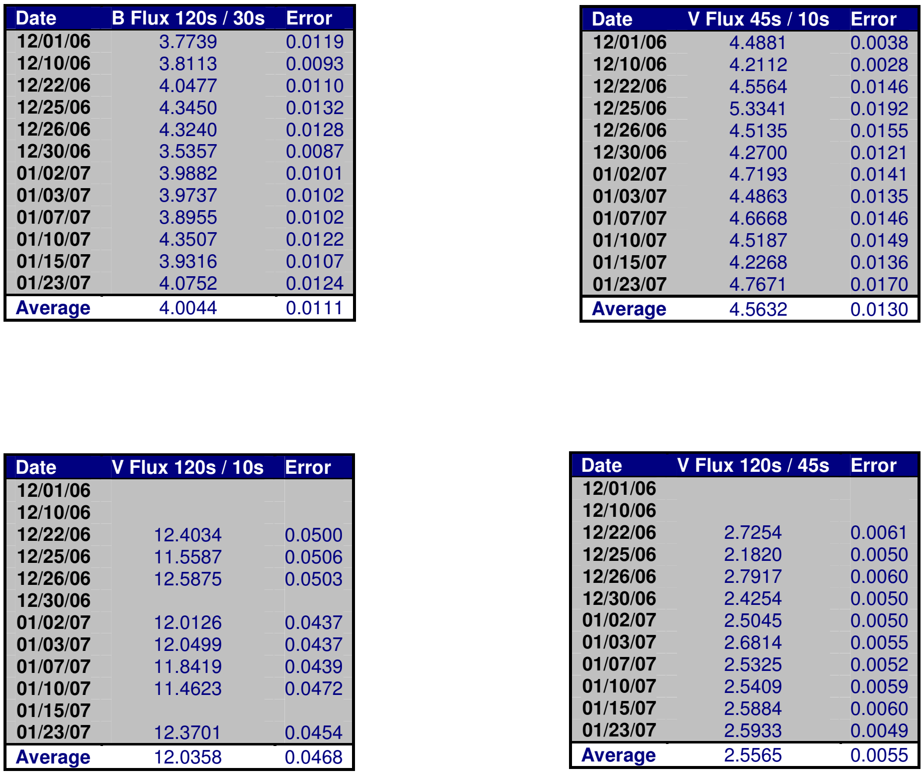
The magnitudes generated by the aperture photometry task *phot* in IRAF generated accompanying errors. Final errors were determined by a standard propagation of error, as follows:

$$\delta z{=}|f'(x)|\delta x.$$

Where, in this case,

$$f'(x){=}ln(2.512){\times}2.512^{x}.$$

The resulting flux ratios are very consistent with what would be expected. For example, the 120s exposures are about 12 times as bright as the 10s exposures, and so forth. Also determined in this study is the fact that the CCD performs linearly up to approximately 60,000 raw counts. Beyond this point, the linearity of the response starts to fail.



Date	B Flux 120s / 30s	Error
12/01/06	3.7739	0.0119
12/10/06	3.8113	0.0093
12/22/06	4.0477	0.0110
12/25/06	4.3450	0.0132
12/26/06	4.3240	0.0128
12/30/06	3.5357	0.0087
01/02/07	3.9882	0.0101
01/03/07	3.9737	0.0102
01/07/07	3.8955	0.0102
01/10/07	4.3507	0.0122
01/15/07	3.9316	0.0107
01/23/07	4.0752	0.0124
Average	4.0044	0.0111

Date	V Flux 45s / 10s	Error
12/01/06	4.4881	0.0038
12/10/06	4.2112	0.0028
12/22/06	4.5564	0.0146
12/25/06	5.3341	0.0192
12/26/06	4.5135	0.0155
12/30/06	4.2700	0.0121
01/02/07	4.7193	0.0141
01/03/07	4.4863	0.0135
01/07/07	4.6668	0.0146
01/10/07	4.5187	0.0149
01/15/07	4.2268	0.0136
01/23/07	4.7671	0.0170
Average	4.5632	0.0130

Date	V Flux 120s / 10s	Error
12/01/06		
12/10/06		
12/22/06	12.4034	0.0500
12/25/06	11.5587	0.0506
12/26/06	12.5875	0.0503
12/30/06		
01/02/07	12.0126	0.0437
01/03/07	12.0499	0.0437
01/07/07	11.8419	0.0439
01/10/07	11.4623	0.0472
01/15/07		
01/23/07	12.3701	0.0454
Average	12.0358	0.0468

Date	V Flux 120s / 45s	Error
12/01/06		
12/10/06		
12/22/06	2.7254	0.0061
12/25/06	2.1820	0.0050
12/26/06	2.7917	0.0060
12/30/06	2.4254	0.0050
01/02/07	2.5045	0.0050
01/03/07	2.6814	0.0055
01/07/07	2.5325	0.0052
01/10/07	2.5409	0.0059
01/15/07	2.5884	0.0060
01/23/07	2.5933	0.0049
Average	2.5565	0.0055

Table 3.4: Photometric linearity study results.

3.3 Dark Current and Temperature Stability

3.3.1 Dark Current Linearity Analysis 1

The first dark current analysis involved dividing the average mean pixel statistics for the 300s dark exposures for a given night by the average mean pixel statistics for the 150s dark exposures for that night. This was done for 34 nights over the observing period, as indicated in Section 2.3.3. Each night's ratio is plotted in Figure 3.1. We would expect the ratio to be consistently close to 2; however, from the data it appears that the 300s darks have on average ∼1.6 times as much dark current as the 150s darks. Averaging all of the ratios for all permutations, the actual value is 1.6168.

The underlying reason for this non-linear scaling was a matter of concern, and for this reason several variations of the study were carried out. For example, a comparison of small sections of the dark frames was performed. These variations on the study yielded the same results.

Further analysis was carried out by dividing the 300s dark exposures by each other, and the 150s dark exposures by each other, to verify that these values consistently yielded unity. Unfortunately, these average values were 1.1138 and 1.1500 respectively. These results were also puzzling because logically the values should offset each other and average out to about 1. The reason that this was not the obtained result is uncertain. However, inspection of the median ratios as opposed to the mean ratios did yield unity. Graphs for the results of each analysis are included in Figures 3.2, 3.3, 3.4 and 3.5.

Unfortunately, an error analysis of this data proved difficult as the only error values obtained were the standard deviations of the raw pixel values in the frames. These errors do not correspond to the errors in the mean or median pixel values. Further consideration of this matter may be warranted.

Due to the inconsistency in this analysis, a second dark current analysis was performed and these results are included in the following section.

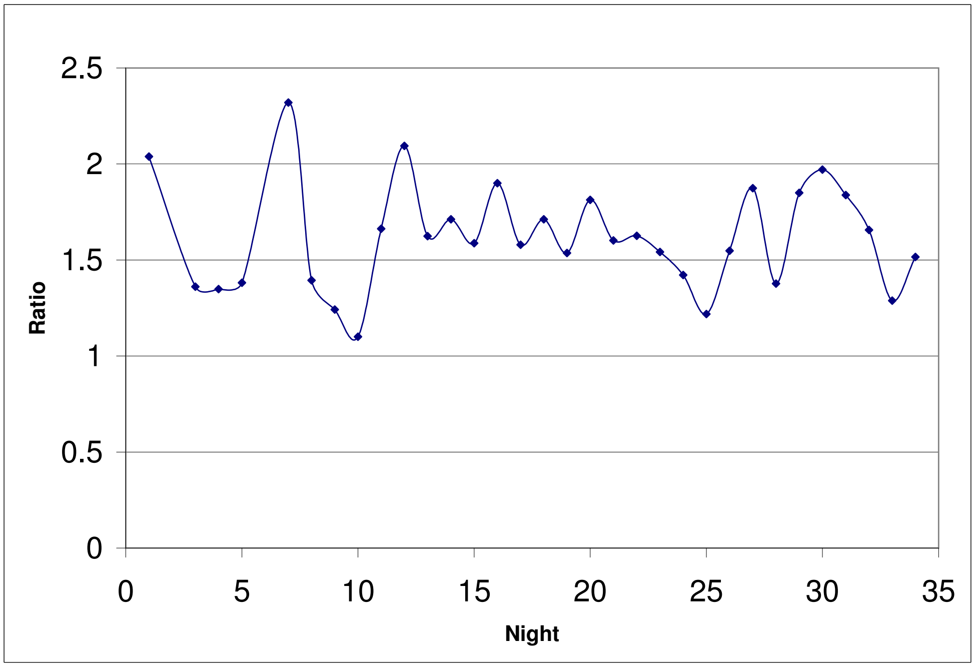


Figure 3.1: 300s Mean Dark Count Divided by 150s Mean Dark Count - Averaged by Night.

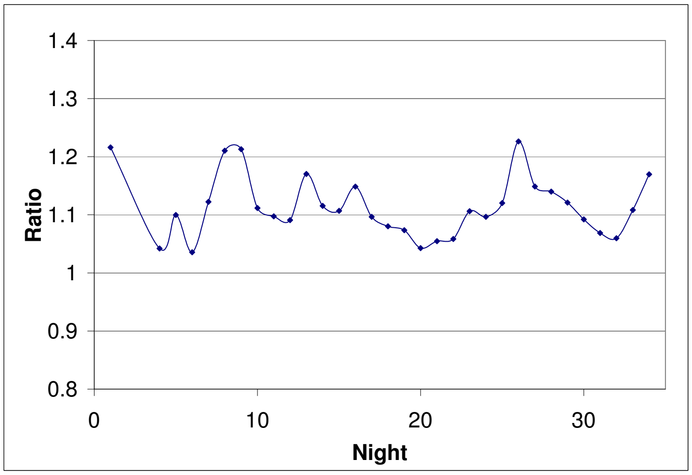


Figure 3.2: 300s Mean Dark Count Divided by 300s Mean Dark Count - Averaged by Night.

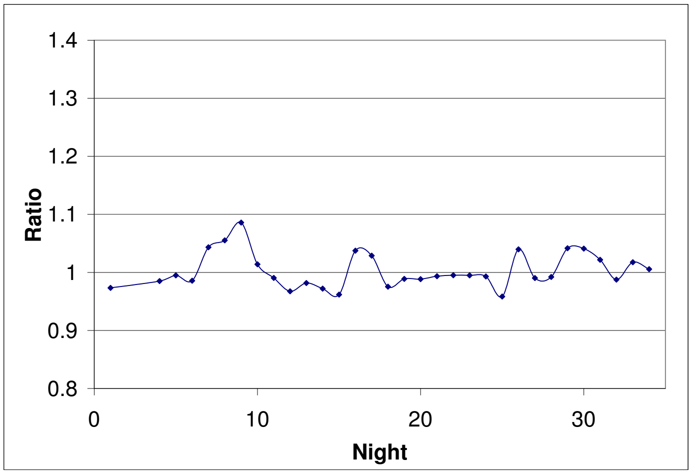


Figure 3.3: 300s Median Dark Count Divided by 300s Median Dark Count - Avg by Night.

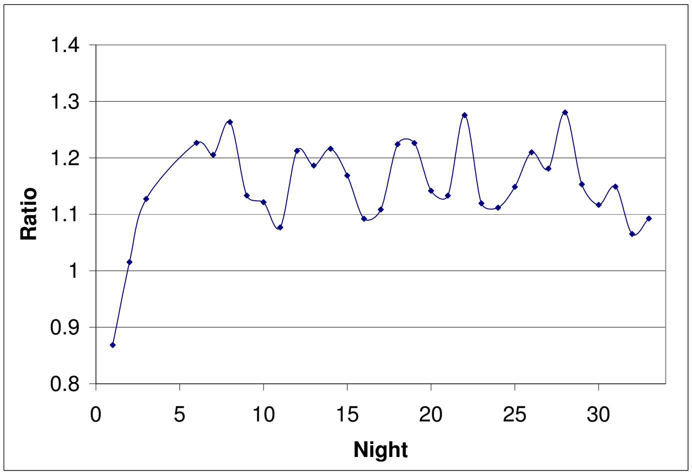


Figure 3.4: 150s Mean Dark Count Divided by 150s Mean Dark Count - Averaged by Night.

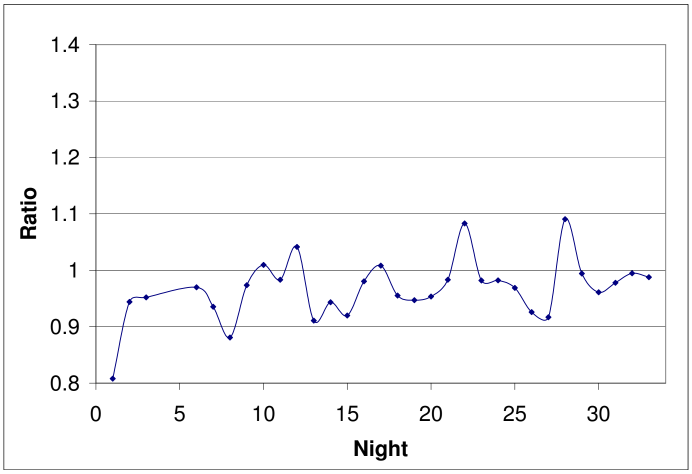


Figure 3.5: 150s Median Dark Count Divided by 150s Median Dark Count - Avg by Night.

3.3.2 Dark Current Linearity Analysis 2

The second dark current investigation was conducted on February 13 and 14, 2007. Professor Joner requested several hours' worth of darks taken for the following exposure lengths: 0, 30, 60, 120, 180, 240 and 300 seconds. This translated into about 22 darks of each exposure length for each day.

After processing the frames as indicated in Section 2.3.3, the statistics were generated for each representative frame and the mean, midpoint and mode values were plotted for each night in Figures 3.6 and 3.10. This revealed some interesting trends. After bias correction, the dark count for the 30 and 60 second exposures for each night was close to or less than zero.

The dark current reached a maximum of about 7 counts for the 300-second exposures on the 14th as compared to a maximum of 14 counts for the 300-second exposures on the 13th. The reason for this is not entirely apparent. It is possible that variations in the combined bias signal could account for this.

It is also interesting to note that the mean values for the different exposure lengths did not scale exactly. Included in Figures 3.7 and 3.11 are graphical representations of how the dark count for different exposure lengths scaled. These figures were obtained by multiplying each exposure by whatever multiplicative factor would generate its equivalent for all other exposure lengths. Table 3.5 contains the values by which each exposure length was multiplied to produce Figures 3.7, 3.11 and 3.12. The flatness of the lines in these figures indicates how linear the scaling is.

After the initial review of these results Professor Joner suggested that the data for the 14th of February be examined again, omitting the first few data points because these points were taken before the CCD had a chance to cool sufficiently. This additional analysis was conducted, and the results are graphed in Figure 3.12. The curves for the second analysis are flatter and show better concordance with the trends from the 13th of February.

Because the scaling of the dark current is varied and differs from what would be expected, these procedures and data were double checked and the same results were found. It is possible that the differences can be attributed in part to the sensitivity of the dark current to temperature fluctuations. While the CCD cooler may have been operating at nominal temperatures, the ambient observatory temperature may have been a factor in these numbers. However, there is no way to quantitatively substantiate this hypothesis with the available data.

Close monitoring and analysis of the CCD temperature was also performed and graphs of the CCD temperature as a function of time are found for the respective days in Figures 3.8 and 3.13. Graphs showing the CCD temperature for the different dark series are shown in Figures 3.9 and 3.14.

From the two main dark linearity studies some general conclusions can be drawn. First, the dark current for short exposures is difficult to quantify, and the dark current for short exposures is suspected to be washed out in the readnoise of the CCD. Generally, it can be concluded that the dark current is negligible for very short exposures. Also, the measured dark current is only accurate with good confidence when the exposure length is 3 minutes or longer.

Table 3.5. Multiplicative Scaling Constants for Figures 3.7, 3.11 and 3.12

Equivalent Exposure Length	120s×	180s×	240s×	300s×
030s	0.250	0.167	0.125	0.100
060s	0.500	0.333	0.250	0.200
120s	1.000	0.667	0.500	0.400
180s	1.500	1.000	0.750	0.600
240s	2.000	1.333	1.000	0.800
300s	2.500	1.667	1.250	1.000

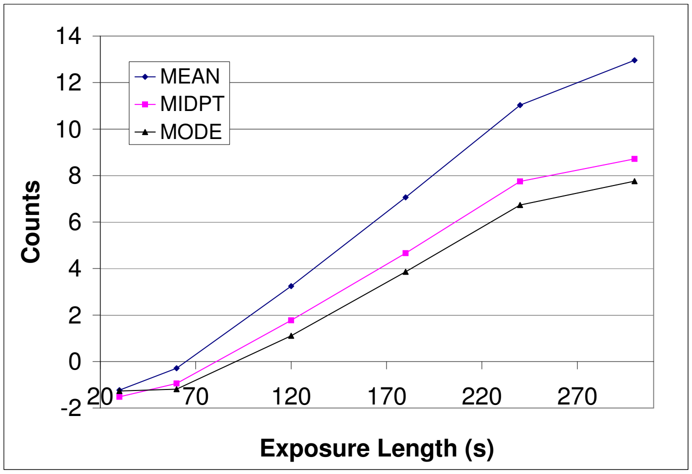


Figure 3.6: Dark Current vs. Exposure Time for February 13, 2007.

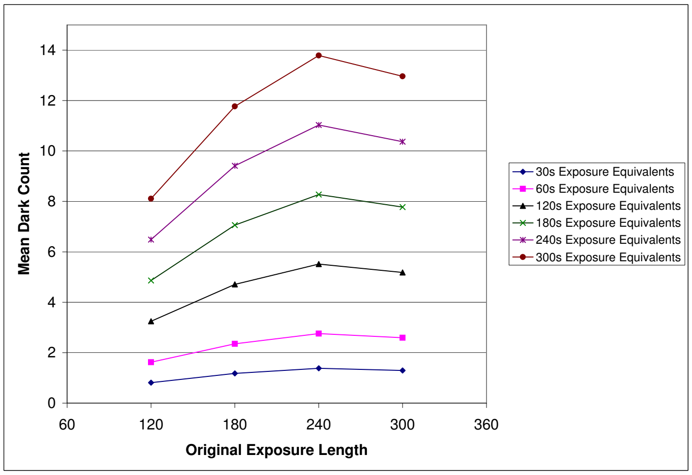


Figure 3.7: Dark Current of Scaled Exposures for February 13, 2007.

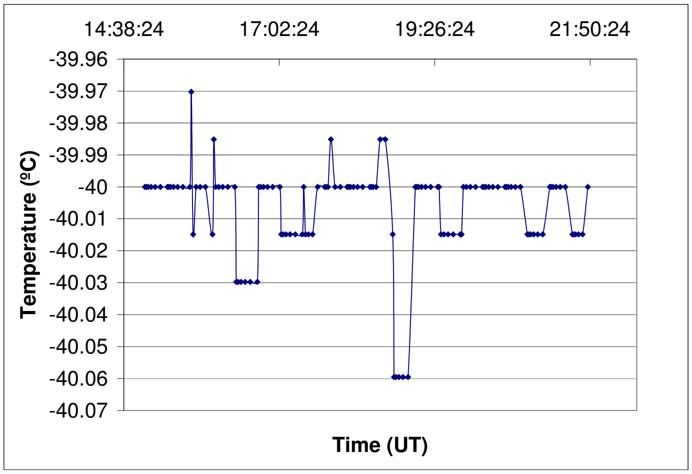


Figure 3.8: CCD Temperature as a Function of Time for February 13, 2007.

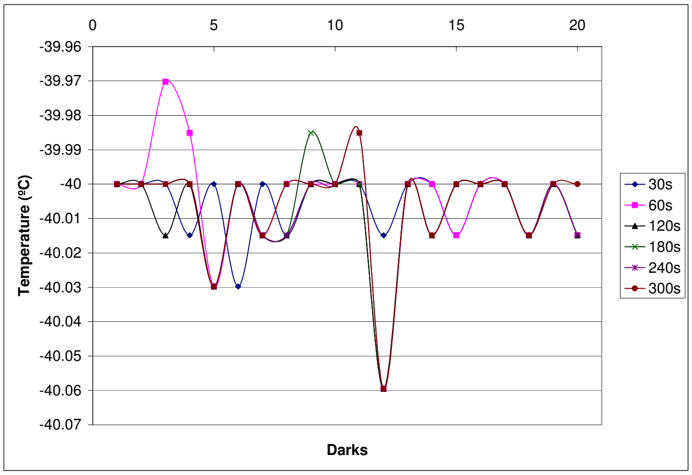


Figure 3.9: CCD Temperature for Different Dark Series for February 13, 2007.

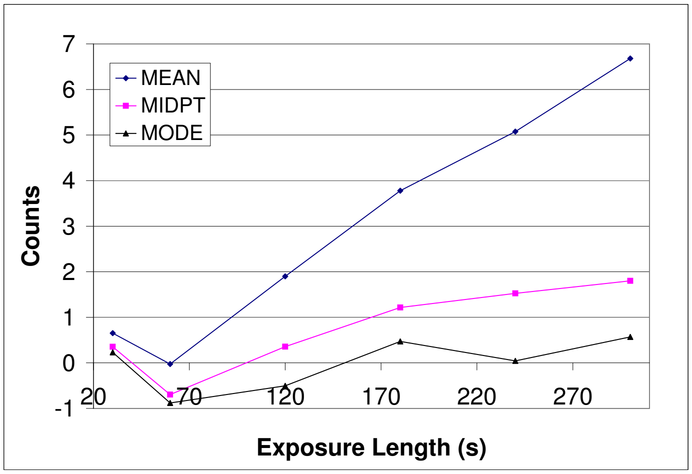


Figure 3.10: Dark Current vs. Exposure Time for February 14, 2007.

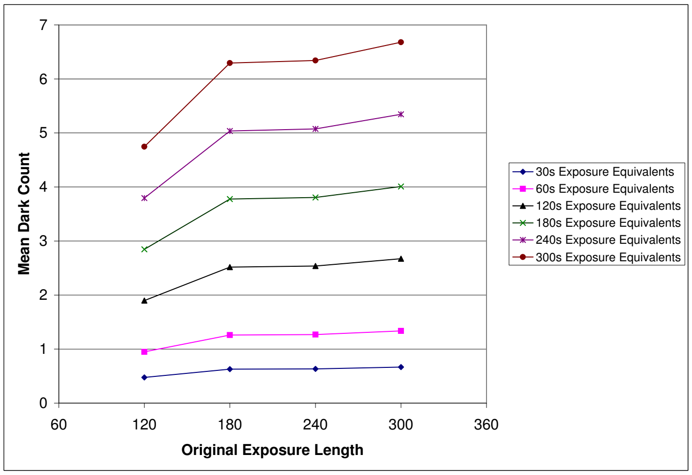


Figure 3.11: Dark Current of Scaled Exposures for February 14, 2007 I.

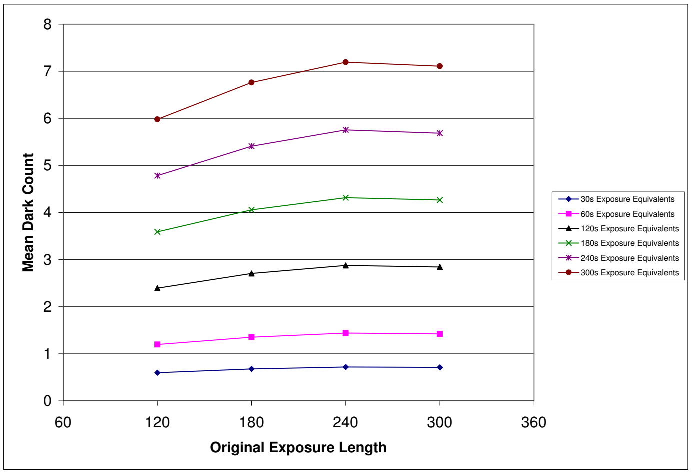


Figure 3.12: Dark Current of Scaled Exposures for February 14, 2007 II.

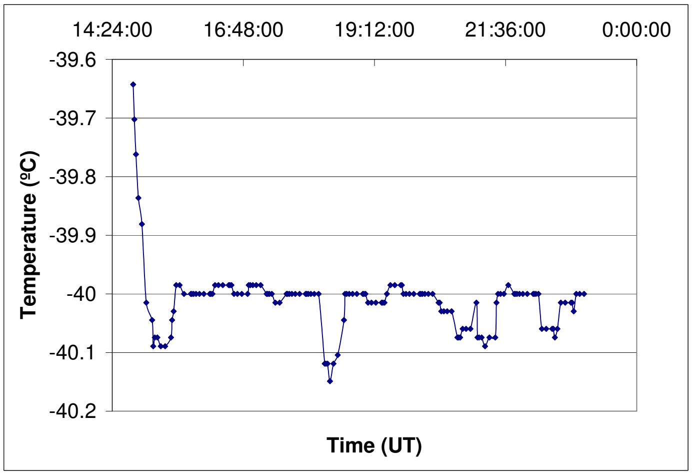


Figure 3.13: CCD Temperature as a Function of Time for February 14, 2007.

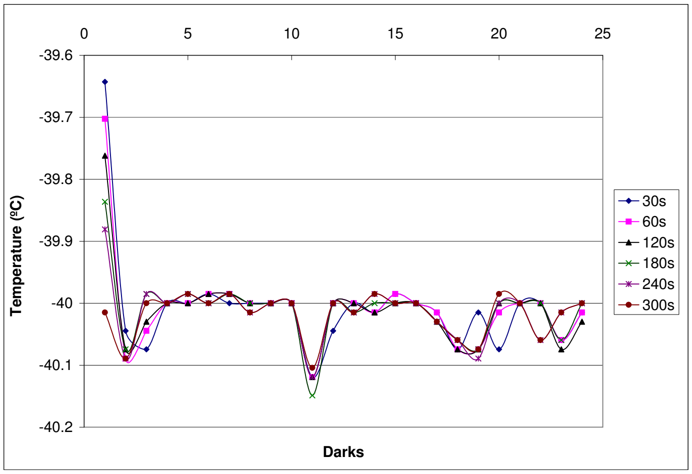


Figure 3.14: CCD Temperature for Different Dark Series for February 14, 2007.

3.3.3 Temperature Stability Analysis

As previously indicated, the monitoring of the temperature stability of the CCD didn't involve a great deal of synthesis. However, since there was a temperature value for every frame taken during the study period, a very large data set resulted. The analysis of this data follows.

First, the temperature values for a night were averaged to give a representative temperature value for the night. Also, the maximum and minimum temperature values for a given night were ascertained. These values were plotted by night for a given month for comparison. Plots for each month are included starting on the next page. Error bars indicate one standard deviation in the night's values.

Average variations from the set temperature of $-40°$ Celsius were figured by taking the absolute value of the difference between the mean temperature for a night and the set point. These values are also plotted below by night for each month.

For the night of September 25, 2006 several bias frames were taken before the CCD had been allowed to cool to its set point. These frames were excluded from the temperature statistics. This was also the case for several bias frames from October 22, 2006, November 2, 2006, December 29, 2006 and January 14, 2007.

An apparent cyclical trend can be seen in the following graphs. This is actually due to smaller data sets alternating in a four-night rotation as a result of only having calibration frames from the nights that BYU did not observe. It should be noticed that these nights with smaller data sets yield higher errors.

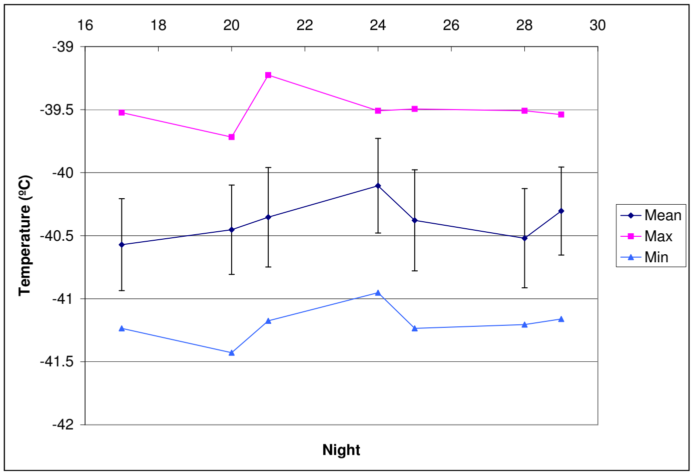


Figure 3.15: CCD Temperature by Night for September 2006.

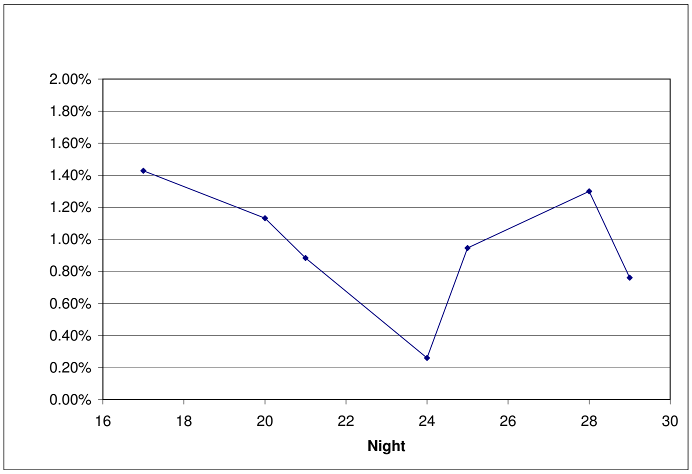


Figure 3.16: Average Variance from Set Temperature for September 2006.

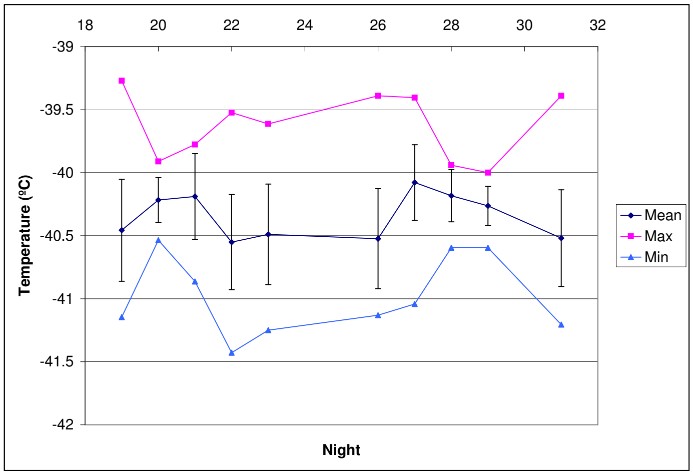


Figure 3.17: CCD Temperature by Night for October 2006.

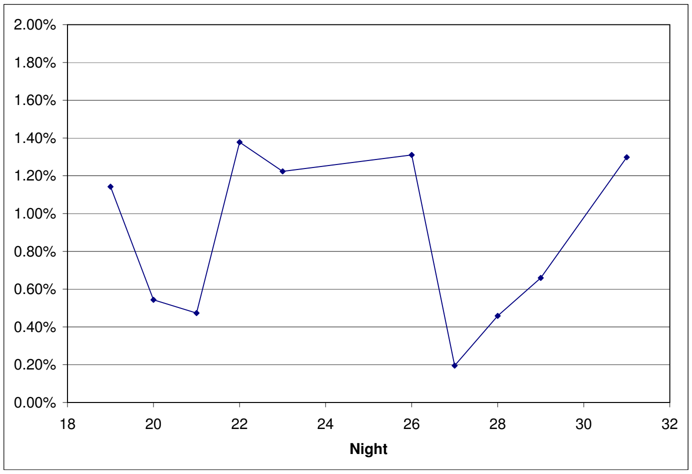


Figure 3.18: Average Variance from Set Temperature for October 2006.

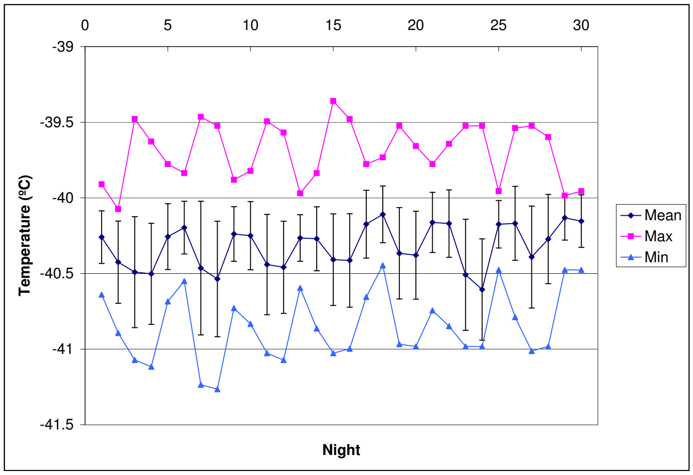


Figure 3.19: CCD Temperature by Night for November 2006.

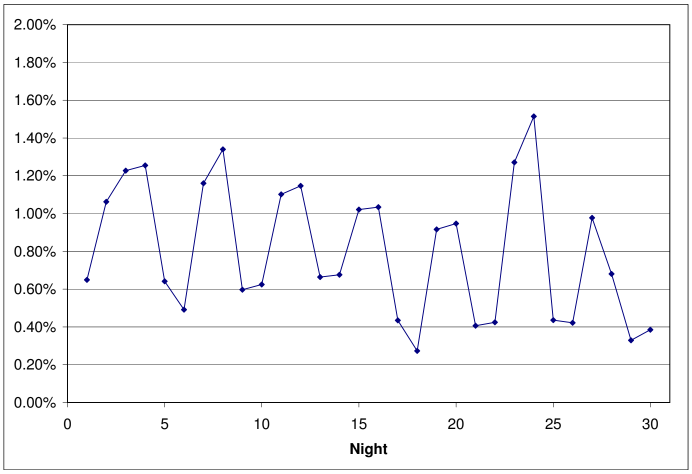


Figure 3.20: Average Variance from Set Temperature for November 2006.

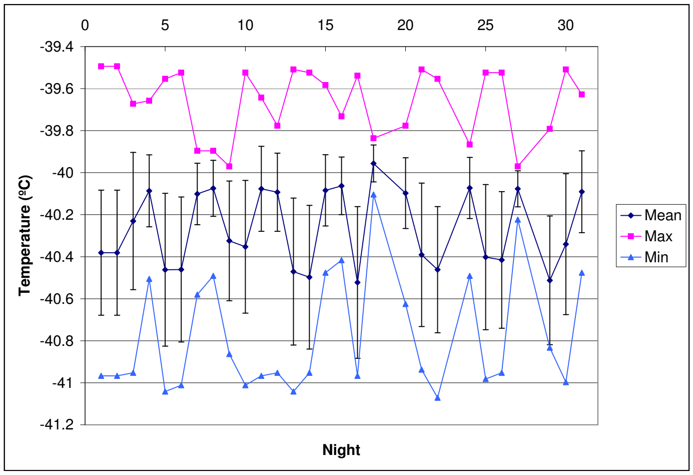


Figure 3.21: CCD Temperature by Night for December 2006.

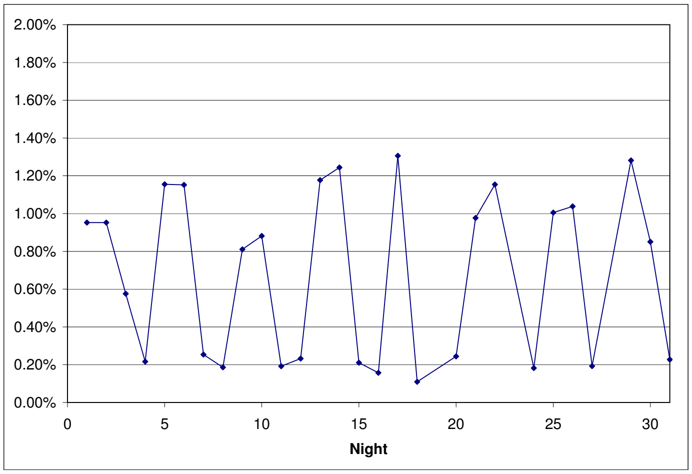


Figure 3.22: Average Variance from Set Temperature for December 2006.

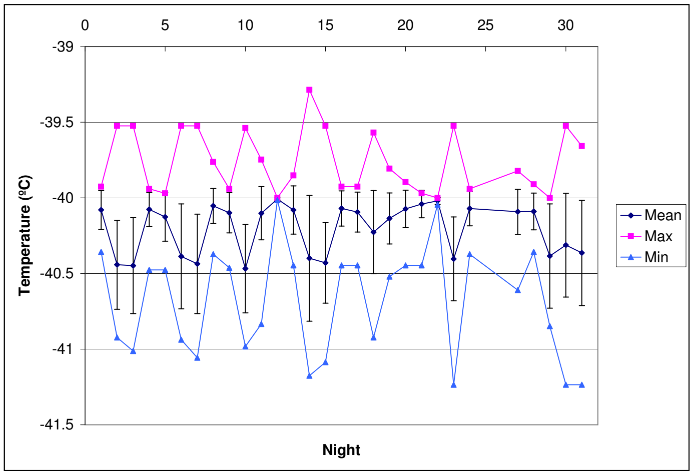


Figure 3.23: CCD Temperature by Night for January 2007.

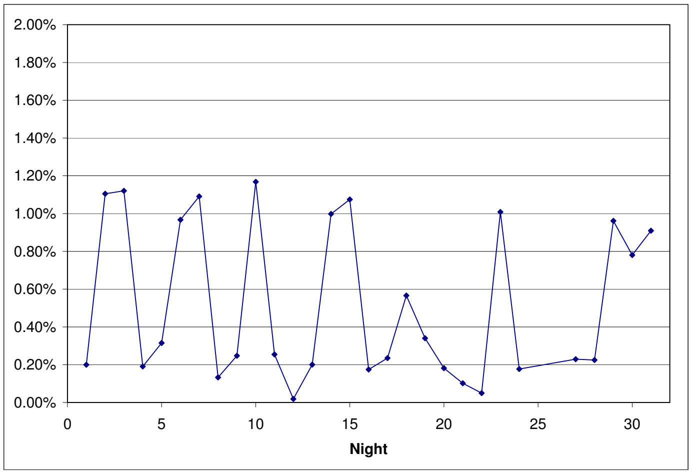


Figure 3.24: Average Variance from Set Temperature for January 2007.

3.4 Bad Pixel Analysis

As previously noted, dark frames were used to produce the first bad pixel maps. However, this yielded only a map of pixels with abnormal responses which would not propagate into the flats and images after dark correction. Additionally, maps made with different dark frames yielded different pixels with abnormal responses, one frame varying from the next. Some pixels could be seen to have a continually abnormal response. Typically these were "hot" pixels with higher than normal count levels. Some of these regions of abnormal pixels that did not remain from map to map may be attributed to cosmic ray hits. However, cosmic ray hits are generally less frequent than would be represented by these variations. A bad pixel map resulting from dark frames is shown in Figure 3.25.

Additional maps were made using combined flats. These flats were processed and combined by filter and then used as input for *ccdmask*. They were combined by filter to remove any stars that may have appeared in one flat field frame or another. The resulting maps were different for each filter and seemed only to be maps of defects in the optical path that appear for that filter's wavelength sensitive range. These included possible dust annuli and dark regions. These features are typical of features that are removed from image frames after flat correction, and therefore do not seem particularly helpful. A few of the bad pixel maps created from flat frames are included in Figure 3.26.

From general visual analysis of the CCD data, some regions are known to be consistently abnormal in their response. These include the approximate regions (in [x1:x2,y1:y2] format): [1:1024,1015:1024], [1:3,1:1024] and [1022:1024,1:1024]. These regions are frequently blank, and should be trimmed from any frames that would be used for full frame analysis. Another region, pointed out by Craig Swenson, consists of part of a single column of pixels with bad response. This column can be seen in some frames at [244:244,1:∼340]. This region corresponds roughly to the line seen in Figure 3.25.

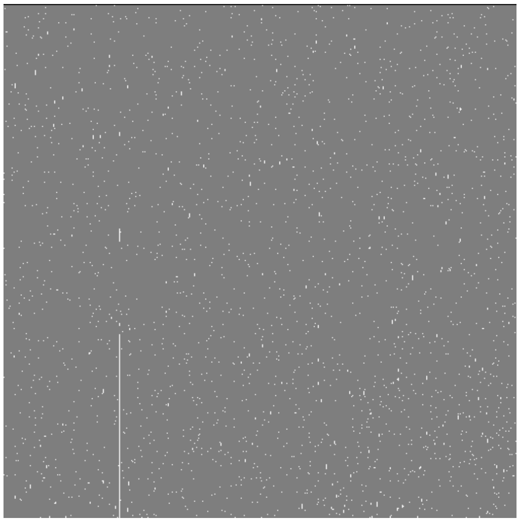


Figure 3.25: Bad Pixel Map Created With Dark Frames.

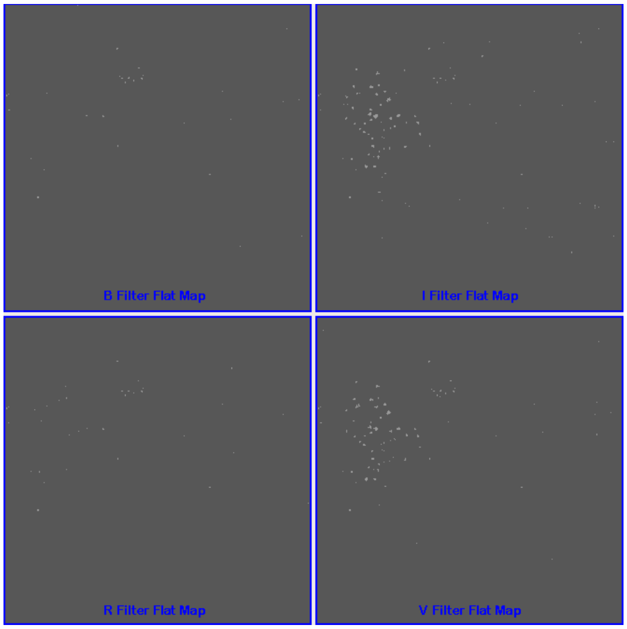


Figure 3.26: Bad Pixel Maps Created With Flat Frames.

Chapter 4

Conclusions and Suggestions for Further Study

This analysis of the SITe 1K CCD of the Tenagra II telescope has proven to be frustrating at times. However, a large amount of quality data has been generated and some interesting conclusions have been obtained.

The gain and readnoise have been established, and these are figures that can be used with some confidence. These figures are approximately 4 e^-/ADU and 29 e^- respectively.

The linearity of the CCD response has been verified, which in itself is very reassuring to know. This fact alone lends credence to the results attained from other studies conducted on the data using differential photometry.

The results for the analysis of the dark current lead to the conclusions that the dark current does not scale entirely linearly over all integration periods. However, the dark current does seem fairly consistent when dark exposures of 3 minutes or longer are used. The dark current for short exposures (up to approximately 100 seconds) is seemingly negligible. From about 100 seconds up to 3 minutes, the dark current is marginally linear and can be significant.

The temperature monitoring analysis has value in its generally informative nature. It can be assumed with fairly good confidence that the temperature stability of the CCD is qualitatively good. Quantitatively speaking, the numbers indicate that the stability is constant to within about $\pm1\%$.

The bad pixel analyses did not produce the results desired, and further consideration to this area is recommended. The results in this area from this study should not be discounted, but at the same time they should be used as more of a starting point rather than a final result. Future analysis in this area may be relegated to further visual inspection of the CCD data.

Many, many hours have been spent developing and synthesizing these results and, while they may not be the most exciting results, getting them has been enjoyable. Hopefully this information will be of use to the students and faculty of Brigham Young University's Astronomy Group as they seek to expand our knowledge by interpreting the information contained in the Tenagra data.

Appendix A

IRAF and Other Scripts

A.1 General Tenagra Scripts

A.1.1 ttasks.cl

```
#ttasks.cl written by Cody Short January 2007, requires ten1headfix.cl, ten2headfix.cl
#and tcalheadfix.cl tasked into login.cl as t1hf, t2hf and tcalhf respectively
#first this script checks to see if there are program frames for a downloaded night of
#data and runs a ccdlist to list what objects were observered writing this information to
#oblist.txt next the script looks through the various folders if they exist and
#restructures everything into a calibrate and images directory the script then removes
#empty folders and runs an imstat on all frames writing this information to imstat.txt
#finally the script calls tencalheadfix in all cases to update the headers for the
#calibration frames if program frames exist the script calls ten1headfix or ten2headfix
#to update the headers for the program frames (at this point whether it calls ten1headfix
#or ten2headfix has to be manually specified in the last few lines of the script)

imred
ccdred
if (access ("images")) {
ccdlist images/BY*.fit > oblist.txt
}
;

mkdir calibrate
if (access ("Bias")) {
mv Bias/* calibrate
}
;
if (access ("Dark")) {
mv Dark/* calibrate
}
;
if (access ("EveningFlat")) {
mv EveningFlat/* calibrate
}
;
if (access ("MorningFlat")) {
mv MorningFlat/* calibrate
}
;
if (access ("images")) {
mv images/B0* calibrate
}
;
if (access ("images")) {
mv images/D0* calibrate
}
;
if (access ("images/calibrate.zip")) {
mv images/calibrate.zip calibrate
}
;

!rmdir Bias Dark EveningFlat MorningFlat;
imstat calibrate/*.fit > imstat.txt

if (access ("images")) {
imstat images/B* >> imstat.txt
}
;

del calibrate/*in2*
del calibrate/*Center.fit

tcalhf  #calls tencalheadfix.cl as tasked to tcalhf in login.cl, runs regardless
        #if there are actual program frames for the night

if (access ("images"))
{
t1hf    #calls ten1headfix.cl or ten2headfix.cl as tasked to t1hf and t2hf
        #(change this line to reflect what software Tenagra is using)
        #respectively in login.cl
t1hf    #runs a second time to change UT value to ##:##:## format (for some
        #reason it doesn't do this the first time through)
}
;
```

A.1.2 trfits.cl

```
#trfits.cl written by Cody Short October 2006, rfitses a nights worth of data
#with all program frames in the images subdirectory and all calibration frames
#in the calibrate subdirectory - uses the .imh image type

if (access ("images")) #checks to see if the images directory exists
{
cd images
mkdir head #creates directory to put the .imh files into
files BYU*.fit > infile.lis #uses the files task to create a list of input files
files BYU*.%fit%imh% > outfile.lis #makes output files list with .fit replaced by .imh
rfits ("@infile.lis", "", "@outfile.lis")
del *.lis
imrename *.imh head #moves the .imh files to the head subdirectory
cd ..
}
;

if (access ("calibrate")) #checks to see if the calibrate directory exists
{
cd calibrate
mkdir head
files *.fit > infile.lis
files *.%fit%imh% > outfile.lis
rfits ("@infile.lis", "", "@outfile.lis")
del *.lis
imrename *.imh head
cd ..
}
;
```

A.1.3 tenheadfix.cl

```
# tenheadfix - cl script by Michael D. Joner - modified to conform to Tenagra
# directory structure by Cody Short Adjusts Tenagra headers to include standard
# keywords. Uses original headers with the format including a single keyword for
# DATE-OBS that includes both the UT date and the time of the observation. Another
# version exists for headers that contain separate keywords for date and time.
# Keywords are added for SUBSET, UT, RA, DEC, EPOCH, and OBSERVAT.  IMAGETYP,
# TELESCOP, INSTRUME, and OBSERVER are corrected.  The setairmass and setjd
# routines are run at the end of the script.  This script assumes a calibrate
# subdirectory containing files with a first letter 'B' for bias (zero) frames,
# 'D' for darks and 'F' for flats.  Header keywords are also edited for these
# calibration frames.  Program frames in the main directory are assumed to start
# with any prefix.fit and so it is important to have only program frames in the
# base folder. (This last sentence does not apply after modification).

cd images
hedit *.fit SUBSET "(FILTER)" add+ ver-
hedit *.fit TELESCOP 32-inch ver-
hedit *.fit INSTRUME "SITE 1K" ver-
hedit *.fit OBSERVER BYU ver-
hedit *.fit OBSERVAT ten add+ ver-
hedit *.fit EPOCH '2000.0' add+ ver-
hedit *.fit IMAGETYP object ver-
hedit *.fit RA "(OBJCTRA)" add+ ver-
hedit *.fit DEC "(OBJCTDEC)" add+ ver-
hselect *.fit $I,DATE-OBS yes > datalist1
!sed "s/[0-9-]*[T]//g" datalist1 > datalist2
list = "datalist2"
while (fscan (list, s1, s2) != EOF)
    hedit (s1, "UT", s2, add+, ver-)
del datalist1
del datalist2
noao
astu
!echo "st = mst(@'date-obs', ut, obsdb (observat, \"longitude\"))" > st.cmds
asthedit *.fit st.cmds table="" verbose+
del st.cmds
hselect *.fit $I,RA,DEC yes > datalist1
!sed "s/\ /:/g" datalist1 > datalist2
list = "datalist2"
while (fscan (list, s1, s2, s3) != EOF) {
    hedit (s1, "RA", s2, add+, ver-)
    hedit (s1, "DEC", s3, add+, ver-)
}
del datalist1
del datalist2
setairmass *.fit
setjd *.fit
cd ..
cd calibrate
hedit F*bin1*.fit SUBSET "(FILTER)" add+ ver-
hedit *.fit TELESCOP 32-inch ver-
hedit *.fit INSTRUME "SITE 1K" ver-
hedit *.fit OBSERVER BYU ver-
hedit *.fit OBSERVAT ten add+ ver-
hedit B*.fit IMAGETYP zero ver-
hedit D*.fit IMAGETYP dark ver-
hedit F*bin1*.fit IMAGETYP flat ver-
hedit B*.fit OBJECT Zero ver-
hedit D*.fit OBJECT Dark ver-
hselect F*bin1*.fit $I,FILTER yes > datalist1
!sed "s/\t/\tFlat/g" datalist1 > datalist2
list = "datalist2"
while (fscan (list, s1, s2) != EOF)
    hedit (s1, "OBJECT", s2, add+, ver-)
del datalist1
del datalist2
cd ..
```

A.1.4 getdata

```
#!/bin/sh
#########################################################
#                                                       #
#  File:     getdata                                    #
#  Purpose:  Automatically downloads data from the      #
#            Tenagra FTP server for a given date.       #
#  Author:   Jacob Albretsesn                           #
#  Misc:     Jake got some help from sites found with   #
#            the google and the force ghost of          #
#            Phil Warner.                               #
#                                                       #
#########################################################

# Give any output pretty colors
green="\033[01;32m";
yellow="\033[01;33m";
blue="\033[01;34m";
red="\033[01;31m";
plain="\033[00m";

# Check to see if running manually or in the cron
if [[ -z "$1" ]]; then verbose=no;                      # No input, most likely
                                                        # running from cron
elif [[ -n "$1" && "-v" == "$1" ]]; then verbose=yes;   # Input was -v, verbose
else verbose=error;                                     # Input was not -v, error
fi

#echo "$verbose" > /home/tenagra/verbose.txt

# Error message for bad input.  Exit script.
if [[ "$verbose" == "error" ]]; then
        echo -e "\n"$red"You FAIL!  Only option is -v for verbose!"$plain"\n";
        exit;
fi

# Get the date to use in part of the directory names
today=`date +%y%m%d`;

# Path to the raw data and log files, change as needed
path='/home/tenagra';                                   # Tenagra User
#path='/data/tenagra/download';                         # Jake Testing

# Run first commands from within the user path
cd "$path";

# Make sure a couple of directories are really there
if ! [ -d "$path/Receipts" ]; then mkdir "$path"/Receipts; fi
if ! [ -d "$path/Notifications" ]; then mkdir "$path"/Notifications; fi

if [ "$verbose" == "no" ]; then
        download_date=$today

        # Download receipts and notifications
        if ! [ -d "$path/logs" ]; then mkdir "$path"/logs; fi
        wget --ftp-user=byu --ftp-password=********** -c -r -nH -a
          "$path"/logs/receipts_"$download_date" ftp://72.165.141.161/Receipts/;
        wget --ftp-user=byu --ftp-password=********** -c -r -nH -a
          "$path"/logs/notifications_"$download_date"
          ftp://72.165.141.161/Notifications/;

        # If the directory to store the files does not already exist, create it
        if ! [ -d "$path/raw" ]; then mkdir "$path"/raw; fi
        if ! [ -d "$path/raw/t$download_date" ]; then mkdir
          "$path"/raw/t"$download_date"; fi

        # Download the data for the day
        cd "$path"/raw/t"$download_date";
        wget --ftp-user=byu --ftp-password=********** -c -r -nH --cut-dirs=1 -a
          "$path"/logs/data_"$download_date"
          ftp://72.165.141.161/"$download_date"_32in/;
        exit;
fi
```

A.2 Gain and Readnoise Scripts

A.2.1 gnrungain.cl

```
# gnrungain.cl written by Cody Short, Feb 07
# runs findgain for different sections of the
# CCD by calling SummerDale Beckstrands
# sdallgain scripts

cd flatb
findgain.section="[758:777,246:265]"
sdallgain
cl <fgain.sd
type allgainoutput.txt > ../gainstatsI.txt
findgain.section="[246:265,246:265]"
sdallgain
cl <fgain.sd
type allgainoutput.txt > ../gainstatsII.txt
findgain.section="[246:265,758:777]"
sdallgain
cl <fgain.sd
type allgainoutput.txt > ../gainstatsIII.txt
findgain.section="[758:777,758:777]"
sdallgain
cl <fgain.sd
type allgainoutput.txt > ../gainstatsIV.txt
findgain.section="[502:521,502:521]"
sdallgain
cl <fgain.sd
type allgainoutput.txt > ../gainstatsV.txt
cd ..

### repeats for all flat directories ###
```

A.2.2 sdallgain.pl

```
#!/usr/bin/perl
#"sdallgain" created by SummerDale Beckstrand Oct 19, 2006
#"sdrdallgain" is required to run this script
#Updates and instructions can be found at http://astronomy.byu.edu/sdb/iraf/
#Email me with questions/suggestions/problems/bugs at SummerDale@gmail.com

@fitlist=`ls *.fit\n`;
$i=$f=$z=0;
while($fitlist[$i]) {
      if ($fitlist[$i] =~ /zero|bias/i) {
            chomp($zero[$z]=$fitlist[$i]);
            $zero[$z] =~ s/.fit//;
            $z++;
      }
      if ($fitlist[$i] =~ /flat/i) {
            chomp($flat[$f]=$fitlist[$i]);
            $flat[$f] =~ s/.fit//;
            $f++;
      }
      $i++;
}
if (($z < 2) or ($f < 2)) {
      die "\nI couldn't find enough bias or flat frames in this directory.\n\n";
}
open(CL,">fgain.sd");
print CL "!rm -f allgain.sd\n";
$Z=$z-1;
$F=$f-1;
$z=$f=0;
$fdiv=$zdiv=1;
while ($zero[$z+1]) {
      while ($zero[$z+$zdiv]) {
            while ($flat[$f+1]) {
                  while ($flat[$f+$fdiv]) {
                        if (($f == 0) and ($z == 0) and ($fdiv == 1) and ($zdiv == 1)) {
                              print CL "findgain $flat[$f] $flat[$f+$fdiv] $zero[$z] $zero[$z+
$zdiv] >allgain.sd\n";
                        }
                        else {
                              print CL "findgain $flat[$f] $flat[$f+$fdiv] $zero[$z] $zero[$z+
$zdiv] >>allgain.sd\n";
                        }
                        $fdiv++;
                  }
                  $f++;
                  $fdiv=1;
            }
            $zdiv++;
            $f=0;
            $fdiv=1;
      }
      $z++;
      $zdiv=1;
      $f=0;
      $fdiv=1;
}
print CL "print \" Data written to allgain.sd \"\n";
print CL "!rm -f header.txt\n";
print CL "imhead $flat[0] >header.txt\n";
print CL "!/home/cshort/scripts/sdrdallgain.pl\n";
close(CL);
print "Type the following:\n\n";
print "    cl <fgain.sd\n\n";
```

A.2.3 sdrdallgain.pl

```
#!/usr/bin/perl
#"sdrdallgain" created by SummerDale Beckstrand Oct 19, 2006
#This is neccessary for running sdallgain
#Updates and instructions can be found at http://astronomy.byu.edu/sdb/iraf/
#Email me with questions/suggestions/problems/bugs at SummerDale@gmail.com

open(FILE,">allgainoutput.txt");
open(DAT,">allgaindat.txt");

$telescope=`grep -i TELESCOP header.txt\n`;
$date=`grep -i DATE-OBS header.txt\n`;
$instru=`grep -i INSTRUME header.txt\n`;

@tele=split(/=/,$telescope);
@dat=split(/=/,$date);
@instr=split(/=/,$instru);

$telescope=$tele[1];
$date=$dat[1];
@dat=split(/T/,$date);
$date=$dat[0];
$instru=$instr[1];

$telescope =~ s/\'//g;
$date =~ s/\'//g;
$instru =~ s/\'//g;

@noise=`grep noise allgain.sd\n`;
@gain=`grep Gain allgain.sd\n`;
@flats=`grep flat allgain.sd\n`;
@zeroes=`grep zero allgain.sd\n`;

$n=$g=0;

while($flats[$g]) {
      @flatline = split(/=/,$flats[$g]);
      $flats[$g] = $flatline[1];
      $flats[$g] =~ s/flatV//g;
      $flats[$g] =~ s/\s//g;
      @flatline = split(/\&/,$flats[$g]);
      $flatdiv[$g] = $flatline[1]-$flatline[0];

      @zeroline = split(/=/,$zeroes[$g]);
      $zeroes[$g] = $zeroline[1];
      $zeroes[$g]  =~ s/zero//g;
      $zeroes[$g]  =~ s/\s//g;
      @zeroline = split(/\&/,$zeroes[$g]);
      $zerodiv[$g] = $zeroline[1]-$zeroline[0];

      $g++;
}

while($noise[$n]) {
      @noises = split(/=/,$noise[$n]);
      $noise[$n] = $noises[1];
      $noise[$n] =~ s/\s//g;
      @noises = split(/e/,$noise[$n]);
      $noise[$n] = $noises[0];
      @gains = split(/=/,$gain[$n]);
      $gain[$n] = $gains[1];
      $gain[$n] =~ s/\s//g;
      @gains = split(/e/,$gain[$n]);
      $gain[$n] = $gains[0];
      $n++;
}

$g=0;
print DAT "FlatDiv ZeroDiv Gain Noise\n\n";
while($noise[$g]) {
      print DAT "$flatdiv[$g] $zerodiv[$g] $gain[$g] $noise[$g]\n";
      $g++;
}
close DAT;

$N=$n;
$n=$Noise=$Gain=0;

while($noise[$n]) {
      $Noise = $Noise + $noise[$n];
      $Gain = $Gain + $gain[$n];
      $n++;
}

$Noise=$Noise/$N;
$Gain=$Gain/$N;

$n=$Nvar=$Gvar=$nhi=$ghi=0;

while($noise[$n]) {
      $nvar[$n]=($noise[$n]-$Noise);
      $nvar[$n]=$nvar[$n]*$nvar[$n];
      if (sqrt($nvar[$n]) > $nhi) { $nhi = sqrt($nvar[$n]); }
      $gvar[$n]=($gain[$n]-$Gain);
      $gvar[$n]=$gvar[$n]*$gvar[$n];
      if (sqrt($gvar[$n]) > $ghi) { $ghi = sqrt($gvar[$n]); }
      $Nvar=$Nvar+$nvar[$n];
      $Gvar=$Gvar+$gvar[$n];
      $n++;
}

$Nvar=$Nvar/$N;
$Gvar=$Gvar/$N;
$Nstd=sqrt($Nvar);
$Gstd=sqrt($Gvar);

if (length($Noise) > 7) {
      $Nois=substr($Noise,0,7);
      $ncheck=$Nois;
      $ncheck .= 5;
      if ($ncheck > $Nois) {
            $last=chop($Nois);
            $last++;
            $Nois .= $last;
      }
      $Noise=$Nois;
}

if (length($Gain) > 7) {
      $Gai=substr($Gain,0,7);
      $gcheck=$Gai;
      $gcheck .= 5;
      if ($gcheck > $Gai) {
            $last=chop($Gai);
            $last++;
            $Gai .= $last;
      }
      $Gain=$Gai;
}

if (length($Nstd) > 5) {
      $Nst=substr($Nstd,0,5);
      $ncheck=$Nst;
      $ncheck .= 5;
      if ($ncheck > $Nst) {
            $last=chop($Nst);
            $last++;
            $Nst .= $last;
      }
      $Nstd=$Nst;
}
```

```
if (length($Gstd) > 5) {
      $Gst=substr($Gstd,0,5);
      $gcheck=$Gst;
      $gcheck .= 5;
      if ($gcheck > $Gst) {
            $last=chop($Gst);
            $last++;
            $Gst .= $last;
      }
      $Gstd=$Gst;
}


if (length($nhi) > 5) {
      $Nh=substr($nhi,0,5);
      $ncheck=$Nh;
      $ncheck .= 5;
      if ($ncheck > $Nh) {
            $last=chop($Nh);
            $last++;
            $Nh .= $last;
      }
      $nhi=$Nh;
}


if (length($ghi) > 5) {
      $Gh=substr($ghi,0,5);
      $gcheck=$Gh;
      $gcheck .= 5;
      if ($gcheck > $Gh) {
            $last=chop($Gh);
            $last++;
            $Gh .= $last;
      }
      $ghi=$Gh;
}

print "\n$date\n";
print "$telescope $instru\n";
print "For $N permutations:\n\n";
print "Average Read Noise = $Noise electrons\n";
print "         Std. Dev. = $Nstd\n";
print "Highest deviation  = $nhi\n\n";
print "Average Gain = $Gain electrons per ADU\n";
print "   Std. Dev. = $Gstd\n";
print "Highest deviation = $ghi\n\n";

print FILE "\n$date\n";
print FILE "$telescope $instru\n";
print FILE "For $N permutations:\n\n";
print FILE "Average Read Noise = $Noise electrons\n";
print FILE "         Std. Dev. = $Nstd\n";
print FILE "Highest deviation  = $nhi\n\n";
print FILE "Average Gain = $Gain electrons per ADU\n";
print FILE "   Std. Dev. = $Gstd\n";
print FILE "Highest deviation = $ghi\n\n";
close(FILE);

print "This output written to allgainoutput.txt\n\n";
```

A.3 Linearity Scripts

A.3.1 autoalign.cl

```
procedure autoalign (ilist, prefix, fwhm, readnoise, gain, xytol, objectn, bug_log, succeed)
#-----------------------------------------------------------------
# autoalign.cl  -
#
#   Documentation
#   -------------
#
# the list:
#   #field name, and list of images in the following lines, etc.
# Example:
#    #GRB990316
#    990316.015
#    990316.016
#    990316.017
#    #AD Leo
#    990316.018
#    990316.019
#    .
#    .
#    .
#
# INSTALL: edit and add the following lines to the login.cl
#        task    $xyshift = $/home/wise-cdr/eran/iraf/bin/xyshift
#        task autodaofind   = /home/wise-cdr/eran/iraf/script/autodaofind.cl
#
#
#
# Written By Eran Ofek, October 1998, Last update: 061098
#-----------------------------------------------------------------

string  ilist     {"",prompt="list of images to align"}
string  prefix    {"a",prompt="prefix for shifted output images"}
real    fwhm      {3.0,prompt="PSF FWHM in pixels"}
real    readnoise {6.50,prompt="CCD read out noise in electrons"}
real    gain      {8.42,prompt="CCD gain in electrons per count"}
real    xytol     {3.0, min=0.0,prompt="matching tolerance for pgshift"}
int     objectn   {50, prompt="Max. Number of stars to match"}
string  bug_log   {"buglog",prompt="logfile name"}
bool    succeed   {no,prompt="succeeded to find astrometric solution"}


struct *lis1
struct *lis2
struct *lis3
struct *lis4

begin

string   imname
string   refimage
string   magfile
real     avshiftx     # shift in X axis.
real     avshifty     # shift in Y axis.
real     pershift     # number of stars used to shift the image.
bool     last_ast
int      match_n      # number of stars matched


delete (bug_log,verify=no,>>&"/dev/null")

#-----------------------------------
# create list without '#'
#-----------------------------------
delete ('tmp_ilist',verify=no,>>&"/dev/null")
lis2 = ilist
while (fscan(lis2, imname)!=EOF)
{
   if (substr(imname,1,1) == '#')
   {
      #jump to next line
   }
   else
   {
      print (imname, >> 'tmp_ilist')
   }
```

```
}


# call autodaofind
lis1 = 'tmp_ilist'
while (fscan(lis1, imname)!= EOF)
{
   autodaofind(imname=imname,out_file="default",fwhm=fwhm,readnoise=readnoise,gain=gain,threshold_sig=
4.0)
}

# find shifts between images
lis3 = ilist
last_ast = yes
while (fscan(lis3, imname)!=EOF)
{
   print ('-----------------------------------')
   print (' Field Line : ',imname)
   print ('-----------------------------------')

   if (substr(imname,1,1) == '#')
   {
      # next field
      last_ast = yes
   }
   else
   {
      if (last_ast==yes)
      {
         last_ast = no
         # set image to be reference image
         refimage = imname
         print ('==================================')
         print ('Reference Image : ', refimage)
         print ('==================================')
         magfile = imname // '.coo.1'
         #----------------------------------------
         # Prepare the image list file for pgshift
         #----------------------------------------
         print("Prepare catalog for findshift")
         delete ('tmp_object',verify=no,>>&"/dev/null")

         print(imname, > 'tmp_object')

         #creating the .smag file
         delete (imname//'.smag.1',verify=no,>>&"/dev/null")
         txdump
(textfile=magfile,fields="ID,XCENTER,YCENTER,MAG,MERR,MSKY,NITER,SHARPNESS,CHI",expr="MAG[1] !
=INDEF",headers=yes,
>>imname//'.smag.1')

         delete (imname//'.rmag.1',verify=no,>>&"/dev/null")
         txdump
(textfile=magfile,fields="ID,XCENTER,YCENTER,MAG,MERR,MSKY,NITER,SHARPNESS,CHI",expr="MAG[1] !
=INDEF",headers=no, >>imname//'.rmag.1')

         print(' sorting '//imname//'.rmag.1')
         delete (imname//'.afnl.1',verify=no,>>&"/dev/null")

         # creating the .afnl file
         # sort the rls file by decreasing magnitude
         sort(input_fi=imname//'.rmag.1',column=4,numeric=yes, >> imname//'.afnl.1')

         imcopy (input=imname, output=prefix//imname)
      }
      else
      {

         magfile = imname // '.coo.1'

         #-------------------------------------------
         # Prepare the image catalog file for pgshift
         #-------------------------------------------
         print("Prepare catalog for findshift")
         delete ('tmp_object',verify=no,>>&"/dev/null")
         print(imname, > 'tmp_object')

         #creating the .smag file
         delete (imname//'.smag.1',verify=no,>>&"/dev/null")
         txdump
(textfile=magfile,fields="ID,XCENTER,YCENTER,MAG,MERR,MSKY,NITER,SHARPNESS,CHI",expr="MAG[1] !
=INDEF",headers=yes,
>imname//'.smag.1')

         delete (imname//'.rmag.1',verify=no,>>&"/dev/null")
         txdump
(textfile=magfile,fields="ID,XCENTER,YCENTER,MAG,MERR,MSKY,NITER,SHARPNESS,CHI",expr="MAG[1] !
=INDEF",headers=no, >>imname//'.rmag.1')

         print(' sorting '//imname//'.rmag.1')
         delete (imname//'.afnl.1',verify=no,>>&"/dev/null")

         # creating the .afnl file
         # sort the rls file by decreasing magnitude
         sort(input_fi=imname//'.rmag.1',column=4,numeric=yes, >> imname//'.afnl.1')

         #---------------
         # compute shifts
         #---------------
         print (' Computing Shifts for image : ',imname)
         delete ('input_shift',verify=no,>>&"/dev/null")
         print (imname//'.afnl.1', >> 'input_shift')
         print (refimage//'.afnl.1', >> 'input_shift')
         print (xytol, >> 'input_shift')

         print(imname//'.afnl.1','\n',refimage//'.afnl.1','\n',xytol,'\n',objectn) | xyshift | scan
(avshiftx, avshifty, pershift, match_n)

            print("------------- Shift in pixels ----------------")
            print("          X = ", avshiftx)
            print("          Y = ", avshifty)

            print("    % = ", pershift)
            print("    n = ", match_n)
            print('Image : ',imname,'     shift% ',pershift, >> bug_log)


         # shift the image
         print ('   ===> Shifting image : ', imname)
         imshift(input=imname, output=prefix//imname, xshift=-avshiftx, yshift=-avshifty)
      }
   }
}
succeed = yes

terminate:

print("End...  Bye.")
beep
beep

end
```

A.3.2 autodaofind.cl

```
procedure autodaofind
#-----------------------------------------------------------------
# autodaofind.cl  - Automatic daofind for image.
#
#
#
#
# By : Eran O. Ofek
# Written: August 1998,    Last Update: Aug 10th, 1998
#-----------------------------------------------------------------

string  imname        {"",prompt="image name"}
string  out_file      {"default",prompt="output file name"}
real    fwhm          {3.0,prompt="PSF FWHM in pixels"}
real    readnoise     {6.50,prompt="CCD read out noise in electrons"}
real    gain          {8.42,prompt="CCD gain in electrons per count"}
real    threshold_sig {4.0,prompt="threshold on sigma above background"}


struct *lis1

begin


real       sky_noise
real       im_sigma




findthresh (images=imname, gain=gain, readnoi=readnoise, ,coaddtype="average", nframes=1,
center="mode", verbose=no) | scan(sky_noise, im_sigma)


print ('  Find stars using Daofind')

delete (imname//'.coo.*',verify=no,>>&"/dev/null")

daofind(image=imname,output=out_file,verify=no, verbose=yes, sigma=sky_noise, scale=1,
fwhmpsf=fwhm, readnoi=readnoise, epadu=gain, thresho=threshold_sig)

print("END autodaofind")

end
```

A.3.3 xyshift.f

```
c this program calculates the shift between two coordinate-files
c written By Uri Giveon
c modified Eran Ofek

       program xyshift
c23456789 123456789 123456789 123456789 123456789 123456789 123456789
       IMPLICIT NONE

       integer      MaxN
       parameter (MaxN=250)

       integer id,i,j,k,l,m,n
       character*70 ImName
       character*70 ImRef

       integer      ObjectN
       real         x1(MaxN)
       real         y1(MaxN)
       real         x2(MaxN)
       real         y2(MaxN)
       real         shiftx
       real         deltax(MaxN,MaxN)
       real         deltay(MaxN,MaxN)
       real         shifty
       real         num1
       real         num(MaxN,MaxN)
       real         object1
       real         object2
       real         sumx(MaxN,MaxN)
       real         sumy(MaxN,MaxN)
       real         sumx1
       real         sumy1
       real         avshiftx
       real         avshifty
       real         pershift
       real         Tolerance


c------ read data from line
       read (5,'(a70)') ImName
       read (5,'(a70)') ImRef
       read (5,*) Tolerance
       read (5,*) ObjectN

       open(2, file=ImName, status='old')
       open(3, file=ImRef, status='old')

       object1=1.
       object2=1.
50     continue
          read(2,*,end=100)id,x1(object1),y1(object1)
             if(object1.eq.ObjectN)then
                goto 110
             endif
          object1=object1+1
       goto 50
100    object1=object1-1
110    continue
          read(3,*,end=200)id,x2(object2),y2(object2)
             if(object2.eq.ObjectN)then
                goto 210
             endif
          object2=object2+1
       goto 110
200    object2=object2-1


210    close (2)
       close (3)

c------ calculate all the possible differences
```

```
       do i=1,object1
          do j=1,object2
             deltax(i,j)=x1(i)-x2(j)
             deltay(i,j)=y1(i)-y2(j)
          enddo
       enddo

c------ find the shift
       num1=1.
       do k=1,object1
       do l=1,object2
          num(k,l)=1.
          sumx(k,l)=deltax(k,l)
          sumy(k,l)=deltay(k,l)
          do m=k,object1
             if(m.eq.k)then
                goto 600
             endif
          do n=1,object2
             if(deltax(m,n).le.deltax(k,l)+Tolerance.and.deltax(m,n)
     >             .ge.deltax(k,l)-Tolerance)then
                if(deltay(m,n).le.deltay(k,l)+Tolerance.and.deltay(m,n)
     >                .ge.deltay(k,l)-Tolerance)then
                   num(k,l)=num(k,l)+1
                      sumx(k,l)=sumx(k,l)+deltax(m,n)
                      sumy(k,l)=sumy(k,l)+deltay(m,n)
                   goto 600
                endif
             endif
          enddo
600       enddo
             if(num(k,l).gt.num1)then
                num1=num(k,l)
                   sumx1=sumx(k,l)
                      sumy1=sumy(k,l)
                   shiftx=deltax(k,l)
                shifty=deltay(k,l)
             endif
       enddo
       enddo

c------ compute the average shifts & the star percent contributing to it

       avshiftx=sumx1/num1
       avshifty=sumy1/num1
       pershift=100*num1/object1

c------ output data

       write(6,'(f8.2,3x,f8.2,5x,f8.2,5x,f6.1)')
     >           avshiftx,avshifty,pershift,num1

       end
```

A.4 Dark Scripts

A.4.1 darkren.cl

```
#darkren.cl written by Cody Short to rename 300 second dark exposures
#and 150 second dark exposure to odd#.fit and even#.fit respectively
#November, 2006

hselect *.fits $I,EXPTIME yes > exptime.txt
match 300 exptime.txt > 300.lis
match 150 exptime.txt > 150.lis
list = "300.lis"
i=1 #rename the 300 second darks odd#.fit
while (fscan (list, s1) != EOF) {
rename (s1,i,field="root",mode="ql")
i=i+2
}
;
list = "150.lis"
i=2 #rename the 150 second darks even#.fit
while (fscan (list, s1) != EOF) {
rename (s1,i,field="root",mode="ql")
i=i+2
}
;
del *.lis
del exptime.txt
hselect *.fits $I,EXPTIME yes > exptime.txt
```

A.4.2 darkimar.cl

```
# darkimar.cl written by Cody Short, November 2006
# creates file lists for use as the operands for
# imarithmatic designed to divide all 300 second
# darks by all 150 second darks. Runs all permutations.
match 300 exptime.txt > 300.lis
match 150 exptime.txt > 150.lis
list = "300.lis"
      while (fscan (list, s1) != EOF) {
      print (s1, >> "numer.lis")
};
list = "150.lis"
      while (fscan (list, s1) != EOF) {
      print (s1, >> "denom.lis")
};
list = "denom.lis"
while (fscan (list, s1) != EOF) {
      s2 = substr (s1, 1, 1)
      print (s2, >> "end.lis")
};
list = "numer.lis"
while (fscan (list, s1) != EOF) {
      s2 = substr (s1, 1, 1)
      print (s2,"by", >> "front.lis")
};
list = "150.lis"
int ndlist=0
while (fscan (list, s1) != EOF) {
      ndlist = ndlist+1
      print (ndlist)
};
list = "300.lis"
int nnlist=0
while (fscan (list, s1) != EOF) {
      nnlist = nnlist+1
      print (nnlist)
};
string *tlist = "end.lis"
string cur_val
for (i=1; i <= ndlist; i=i+1) {
      cur_val = tlist
      list = "front.lis"
      while (fscan (list, s1) != EOF) {
            print (s1,cur_val,".fits", >> "outfile.lis")
};
};
string *numer = "numer.lis"
string cur_num
string *denom = "denom.lis"
string cur_den
string *out = "outfile.lis"
string cur_out
for (i=1; i <= ndlist; i=i+1) {
      numer = "numer.lis"
      cur_den = denom
      print (cur_den)
      for (j=1; j <= nnlist; j=j+1) {
            cur_num = numer
            print (cur_num)
            cur_out = out
            print (cur_out)
            imarith(cur_num,"/",cur_den,cur_out);
};
};
del *.lis
```

Appendix B

Photometry Parameters

B.1 centerpars, datapars, photpars and polypars

```
centerpars

noao - digiphot - apphot - centerpars

(calgori   =  centroid)
(cbox      =        7.)
(cthresh   =      0.25)
(minsnra   =         1)
(cmaxite   =        10)
(maxshif   =       8.5)
(clean     =        no)
(rclean    =         1)
(rclip     =         2)
(kclean    =         3)
(mkcente   =        no)
(mode      =        ql)
```

```
datapars

noao - digiphot - apphot - datapars

(scale     =         1)
(fwhmpsf   =       2.5)
(emissio   =       yes)
(sigma     =     INDEF)
(datamin   =     INDEF)
(datamax   =     INDEF)
(noise     =   poisson)
(ccdread   =          )
(gain      =          )
(readnoi   =        0.)
(epadu     =        1.)
(exposur   =          )
(airmass   =          )
(filter    =          )
(obstime   =          )
(itime     =        1.)
(xairmass  =     INDEF)
(ifilter   =     INDEF)
(otime     =     INDEF)
(mode      =        ql)
```

```
photpars

noao - digiphot - apphot - photpars

(weighti   =  constant)
(apertur   =        3.)
(zmag      =        25)
(mkapert   =        no)
(mode      =        ql)
```

```
polypars

noao - digiphot - apphot - polypars

(zmag      =  25)
(mkpolyg   =  no)
```

B.2 findpars, fitskypars and phot

```
findpars

noao - digiphot - apphot - findpars

(thresho   =    4)
(nsigma    =  1.5)
(ratio     =    1)
(theta     =    0)
(sharplo   =  0.2)
(sharphi   =    1)
(roundlo   =   -1)
(roundhi   =    1)
(mkdetec   =   no)
(mode      =   ql)
```

```
fitskypars

noao - digiphot - apphot - fitskypars

(salgori   = centroid)
(annulu    =       10)
(dannulu   =       10)
(skyvalu   =        0)
(smaxite   =       10)
(sloclip   =        0)
(shiclip   =        0)
(snrejec   =       50)
(slorejec  =        3)
(shireje   =        3)
(khist     =        3)
(binsize   =      0.1)
(smooth    =       no)
(rgrow     =        0)
(mksky     =       no)
(mode      =       ql)
```

```
phot

noao - digiphot - apphot - phot

(coords    =      ds9.reg)
(output    =             )
(plotfile  =             )
(datapar   =             )
(centerp   =             )
(fitskyp   =             )
(photpar   =             )
(interac   =           no)
(radplot   =           no)
(icomman   =             )
(gcomman   =             )
(wcsin     =     )_.wcsin)
(wcsout    =    )_.wcsout)
(cache     =     )_.cache)
(verify    =           no)
(update    =           no)
(verbose   =          yes)
(graphic   =  )_.graphics)
(display   =   )_.display)
```

References

The following references include both sources cited and sources referenced generally.

Anderson, E. 1989, “An Introductory Users Guide to IRAF Scripts”, IRAF.net Documents, Retrieved September, 2006 from: http://iraf.net/irafdocs/script/

Barnes, J. 1993, “A Beginners Guide to Using IRAF”, IRAF.net Documents, Retrieved September, 2006 from: http://iraf.net/irafdocs/beguide/

Beckstrand, S. 2007, private communications

Hintz, E. G. 2007, private communications

Janesick, J. & Blouke, M. 1987, “Sky on a Chip: The Fabulous CCD”, Sky & Telescope, Sept. 1987, 238

Joner, M. D. 2007, private communications

Massey, P. (1997), “A Users Guide to CCD Reductions with IRAF”, IRAF.net Documents, Retrieved September, 2006 from: http://iraf.net/irafdocs/ccduser3/

Ofek, E. 2007, private communications

Peterson, C. 2001, “How it Works: The Charge-Coupled Device, or CCD”, Journal of Young Investigators, 1 Retrieved February, 2007 from: http://www.jyi.org/volumes/volume3/issue1/features/peterson.html

Scientific Imaging Technologies 2003, “SITe SI03xA 24 μm Charge-Coupled Device Family”, 12, Retrieved January, 2007 from: http://www.site-inc.com/downloads/SITe-S103(403).pdf

Tenagra Observatories 2007, Astronomical Data & Observatory Information, Tenagra Observatories FTP and HTTP sites, http://www.tenagraobservatories.com/

Index