5-2013

Error Modeling and Analysis of Star Cameras for a Class of 1U Spacecraft

David M. Fowler
Utah State University

ERROR MODELING AND ANALYSIS OF STAR CAMERAS FOR A CLASS OF 1U SPACECRAFT

by

David M. Fowler

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Aerospace Engineering

Approved:

Dr. R. Rees Fullmer Major Professor	Dr. Charles M. Swenson Committee Member
Dr. David Geller Committee Member	Dr. Mark R. McLellan Vice President for Research and Dean of the School of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2013

ABSTRACT

Error Modeling and Analysis of Star Cameras for a Class of 1U Spacecraft

by

David M. Fowler, Master of Science

Utah State University, 2013

Major Professor: Dr. R. Rees Fullmer
Department: Mechanical and Aerospace Engineering

As spacecraft today become increasingly smaller, the demand for smaller components and sensors rises as well. The smartphone, a cutting edge consumer technology, has impressive collections of both sensors and processing capabilities and may have the potential to fill this demand in the spacecraft market. If the technologies of a smartphone can be used in space, the cost of building miniature satellites would drop significantly and give a boost to the aerospace and scientific communities.

Concentrating on the problem of spacecraft orientation, this study sets ground to determine the capabilities of a smartphone camera when acting as a star camera. The study developed an overall error model of the camera both through analytical models as well as ground-based observed results and compares these results to those of higher quality imagers. The results of the study reveal the abilities of low-cost off-the-shelf imagers in space and give a starting point for future research in the field.

The study began with a complete geometric calibration of each analyzed imager such that all comparisons start from the same base. The calibration process resulted in the intrinsic and the extrinsic camera parameters. After the cameras were calibrated, image processing techniques were introduced to correct for atmospheric, lens, and image sensor effects. After passing the data through a star identification algorithm, an attitude for each test image was determined. Analyses of these attitude solutions allow the overall errors of each camera to be defined and provide insight into the abilities of low-cost imagers.

(164 pages)

PUBLIC ABSTRACT

Error Modeling and Analysis of Star Cameras for a Class of 1U Spacecraft

by

David M. Fowler, Master of Science

Utah State University, 2013

Major Professor: Dr. R. Rees Fullmer
Department: Mechanical and Aerospace Engineering

As spacecraft today become increasingly smaller, the demand for smaller components and sensors rises as well. The smartphone, a cutting edge consumer technology, has impressive collections of both sensors and processing capabilities and may have the potential to fill this demand in the spacecraft market. If the technologies of a smartphone can be used in space, the cost of building miniature satellites would drop significantly and give a boost to the aerospace and scientific communities.

Concentrating on the problem of spacecraft orientation, this study sets ground to determine the capabilities of a smartphone camera when acting as a star camera. Orientations determined from star images taken from a smartphone camera are compared to those of higher quality cameras in order to determine the associated accuracies. The results of the study reveal the abilities of low-cost off-the-shelf imagers in space and give a starting point for future research in the field.

The study began with a complete geometric calibration of each analyzed imager such that all comparisons start from the same base. After the cameras were calibrated, image processing techniques were introduced to correct for atmospheric, lens, and image sensor effects. Orientations for each test image are calculated through methods of identifying the stars exposed on each image. Analyses of these orientations allow the overall errors of each camera to be defined and provide insight into the abilities of low-cost imagers.

(164 pages)

ACKNOWLEDGMENTS

First, I would like to sincerely thank my major professor, Dr. R. Rees Fullmer, for his guidance and support. His insight and persistence helped me overcome all the hurdles faced in this research. Our talks on southern Utah parks brought relaxation to the times that were tough.

I wish to thank my parents for always being there for me and encouraging me to strive for the best and not to give up. The ideals and beliefs they raised me with made me who I am today.

I would also like to thank some friends, Michal Hradisky and Milada Majerova, for all the good times we spent together. These times helped me clear my head when it needed to be cleared and gave me back my focus.

Lastly and most importantly, I wish to thank my wife, Emilie, for all her support, love, and patience throughout this research endeavor of mine. She has kept by my side keeping me both motivated and encouraged. Finally, I would like to thank my daughter, Jane, for her smiles and laughs when I was both at and away from my desk. She helped keep me in check with the important things in life.

David M. Fowler

CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

RGB	=	A color model based on the three additive primary colors, red, green and blue
FOV	=	Field of View
CMOS	=	Complementary metal–oxide–semiconductor, a type of Active-Pixel-Sensor used in image sensors
CCD	=	Charge-Coupled Device, an image sensor type
BSI	=	Backside Illumination, a technology used in image sensors
ERS	=	Electronic Rolling Shutter, a shutter technology used in cameras
PCB	=	Printed Circuit Board
GUI	=	Graphical User Interface
GPS	=	Global Positioning System
SNR	=	Signal-to-Noise Ratio
PDE	=	Partial Differential Equation
PSF	=	Point Spread Functions
RMS	=	Root Mean Square
ECI	=	Earth Centered Inertial coordinate system
GUI	=	Graphical User Interface
API	=	Application Programming Interface

CHAPTER 1

INTRODUCTION

I. Overview

A spacecraft, whether it is a satellite or a transport vehicle, is reliant on a guidance system capable of determining its orientation in space, otherwise known as attitude. Star cameras are known to provide the most accurate solution to this problem. All satellites requiring high pointing accuracy, such as those designed for communications, employ star tracking cameras. As spacecraft become smaller and more disposable their star cameras must as well. Space certified commercial star cameras are expensive ranging from tens to hundreds of thousands of dollars range and therefore does not fit within the disposable spacecraft model. Though miniature star cameras do exist they are still custom built for space applications. With consumer technologies driving today's market it would seem that alternatives to space fabricated star cameras would be plausible.

II. Star Cameras

Spacecraft with onboard star cameras first started to appear in the early 1970's and brought about the introduction of guiding spacecraft from the stars. Though most of the technology existed decades before and was used in astronomer observatories to guide telescopes it was not until solid-state electronics were developed that space applications became possible. Today, star cameras are the basis of all high-accuracy guidance systems and are able to obtain attitude solution accuracies between 0.01 and 200 arc-seconds [1]–[4].

Numerous commercially available star camera systems exist and have been tested on-orbit including: Terma HE-5AS [5], ST5000 [6], HAST [7], Ball CT-633 [8] and others. Their applications range from Canopus to Gimbaled to Rocket/Missile to Fine Guidance based [1]. Miniature star camera advancements are spreading commercially as well due to the onset of the new micro and CubeSat form-factor spacecraft becoming popular. The Blue Canyon XACT [9], the Berlin Space Technologies BST ST-200 [10] and the Utah State University Space Dynamics Laboratory DISC [11] are among the few available and provide attitude solutions accurate anywhere between 30 and 200 arc-seconds. The sizes of these miniature star

cameras range form 3x3x4cm to 10x10x7cm. Though these specifications are perfect for miniature satellites the question exists can even smaller cameras achieve similar results?

The accuracy of star cameras is divided into two components: (1) pointing error and (2) boresight roll error. The pointing error is a measure of a camera's ability to measure in the pitch and yaw directions. The boresight role error is a measure of a camera's ability to measure rotationally about its bore axis. Combined, these two error components form a star camera's total error. This error is measured in angular units of either: degrees (deg), arc-minutes (arcmin), arc-seconds (arcsec), radians (rad) or milli-radians (mrad). Table 9.2 shows the conversion factors between each of these common units.

Table 1.1 Angular measurement conversions

Unit	Abbreviation	Equivalent Value
degrees	deg	1
arc-minutes	arcmin	60
arc-seconds	arcsec	3600
radians	rad	0.0175
milli-radians	mrad	17.4533

A star camera is nothing more than a digital camera with embedded hardware and software providing the capability to process and identify stars on a captured image. Digital imaging devices are readily available in today's commercial market showing up in everything from ball-point pens to cars driving down the road. However, the largest exploding market for miniature cameras is the smartphone, with nearly 50% of the US population owning or having access to one in their household, a figure which rose from nothing nearly 5 years ago. Smartphones provide a platform for both acquiring and processing data from an array of embedded sensors including digital cameras, accelerometers, gyroscopes, thermometers, magnetometers, altimeters and GPS chips. While most all of those sensors are attractive for space applications, the camera alone could be enough to orientate a spacecraft alongside a star identification algorithm. While this scenario is certainly intriguing, there are many foreseeable hindrances which could prevent execution of such a plan. The greatest of which being whether the miniature electronics can withstand the extreme conditions of space (e.g. temperature swings and radiation exposure)? Moreover, considering a smartphone

camera as a star camera, are the optics and imaging chip capable of meeting the performance and accuracy demands of space applications?

Low-cost, off-the-shelf imagers tend to be designed for personal imaging and therefore generally have properties including: a large field-of-view (FOV) optics on the order of 50°, a lossy-compression image output format, noisy sensor output, unsatisfactory low-light capabilities and poor quality optics. In contrast, normal spacecraft star cameras tend to have just the opposite: small FOV optics on the order of 12°, a RAW image output format, low-noise sensor output, good low-light capabilities and precision optics. For a low-cost camera to perform as a star camera, it needs to be shown that despite these adverse properties an accurate attitude solution is still attainable.

While low-cost imagers seem like a plausible approach to the problem, the basic questions involved in their success are also relevant to any camera system being used as a star camera. Methods need to be made which can take an imperfect system and calibrate out the imperfections. Imperfections are inherent in any system and star cameras are no different. Everything from the optics to the star identification techniques create errors which multiply and cause direct inaccuracies in the spacecraft attitude. For comparison, typical expensive commercial star cameras range in accuracy from 1 arc-minute to less than 1 arc-second. If a star camera based on low-cost components could attain an accuracy to match that of the least accurate available today it would be a breakthrough and would revolutionize how small spacecraft fly. The price difference alone would allow more satellites to go up at a fraction of the cost. However, in order to begin developing and testing such a camera a detailed understanding of how a star camera works is needed.

All star camera systems requires calibration. The calibration process includes parameterization of all camera components from the lens and the imaging sensor as well as their respective orientations. After the system is calibrated, images can be acquired and processed. Image processing must include ways to filter out stars from the image noise, find the centroid of each star in the image plane and finally identify the stars. For ground based testing purposes atmospheric seeing, image distortions due to air turbulence, atmospheric refraction and image distortions due to the declination angle, must also be taken into account. Once the stars are identified an attitude quaternion or rotation matrix is determined which maps the star camera to the stars in the sky [12].

III. Research Objectives

A. Thesis Statement

The objective of this research is to determine the feasibility of low-cost, off-the-shelf imagers to be used as star cameras and to understand the physical limitations of such cameras and how these limitations affect an attitude solution. The feasibility will be determined by the ability of the camera to obtain consistent attitude accuracy less than 5 arc-minutes.

B. Objectives

The following four underlying goals of this research will lead to an understanding of the capabilities of star cameras based on low-cost imaging architectures.

1. Determine calibration and correction methods to compensate for the primary sources of error induced on an image of the night-sky taken from an earth-based observation point. The error sources are limited to those involving the imaging sensor, the camera lens and the atmosphere.
2. Develop computer software to process star images and calculate attitude solutions. MATLAB provides the coding foundation for this software which allows a user to manipulate various input parameters, control a variety of processing stages and view desired solutions.
3. Determine if integrated smartphone cameras are capable of viewing stars and obtaining an attitude solution. Night-sky images will be taken and processed using software.
4. Determine the attitude accuracies of two low-cost imagers. Long-term and short-term night-sky images will be taken, processed and compared using standard error modeling.

IV. Imager Selection

Two low-cost imagers of distinctly different architectures have been chosen as the test platform of this research. The first imager is the integrated camera of a Motorola Droid X2 smartphone and the second imager is the Aptina MT9P031 built on a demo printed-circuit-board (PCB) having a USB 2.0 interface.

A. Motorola Droid X2

The Motorola Droid X2 contains an embedded 8MP image sensor manufactured by OmniVision. This camera system will be used to determine the star imaging abilities of commercial smartphones. The camera's specifications are analogous to those developed for smartphones manufactured in mid to late 2011. The properties listed in Table 9.2 originated from a combination of image metadata and the OmniVision sensor product brief [13].

Table 1.2 Motorola Droid X2 camera specifications

Specifications	
Manufacturer	OmniVision
Model	OV8812
Resolution	8MP
Sensor Type	CMOS
Sensor Technology	BSI
Optical Format	1 / 3.2 inch
Focal Length (∞)	5mm
Aperture	f / 2.6
Active Imager Size	4614 μm x 3506 μm
Active Pixels	2448 (V) x 3264 (H)
Pixel Size	1.4 x 1.4 μm (0.9058 arc-minutes)
Chroma	RGB
Shutter Type	ERS
Responsivity	~600 mV/lux-sec
Pixel Dynamic Range	~68dB
SNR (MAX)	~35dB
Output Format	8-/ 10-/ 12-bit RAW RGB
Image Format	JPEG
Image Bit-Depth	24-bit (8 bit-per-channel)

Figure 1.1 shows the front, back and side views of the Motorola Droid X2. On the upper left corner of the back view lies the camera. The phone is capable of being controlled by a computer while they are linked through the USB interface. The phones display can also be remotely turned off which is necessary to reduce ambient light noise while imaging the night sky. The remote connection allows the operation of a camera application installed on the phone and provides a means to acquire images without disturbing the position of the camera.

Figure 1.1 Motorola Droid X2

B. Aptina MT9P031

The Aptina MT9P031 camera has a 5MP CMOS imaging sensor which has exceptional low-light capabilities. This camera represents the board cameras architecture. The camera has a standard C-mount lens adapter allowing for lens swap-out; however, only one lens is used in this study. The properties listed in originated from a combination of lens data and the Aptina data sheets [14],[15].

Figure 1.2 shows the Aptina MT9P031 camera and the demo PCB's USB 2.0 interface connection. The camera can be controlled by a computer through the USB interface. Camera control software developed by Aptina allows full control of the imaging properties. The GUI software provides the tools necessary to capture images of the night sky using the Aptina imager.

Table 1.3 Aptina MT9P031 camera specifications

Specifications	
Manufacturer	Aptina
Model	MT9P031
Resolution	5MP
Sensor Type	CMOS
Sensor Technology	BSI
Optical Format	1 / 2.5 inch
Focal Length (∞)	6mm
Aperture	f / 1.4 – f / 16
Active Imager Size	5700 µm x 4280 µm
Active Pixels	1944 (V) x 2592 (H)
Pixel Size	2.2 x 2.2 µm (1.14 arc-minutes)
Chroma	Monochrome
Shutter Type	ERS
Responsivity	1.4 V/lux-sec (550nm)
Pixel Dynamic Range	70.1dB
SNR	38.1dB
Output Format	12-bit RAW Monochrome
Image Format	RAW
Image Bit-Depth	12-bit

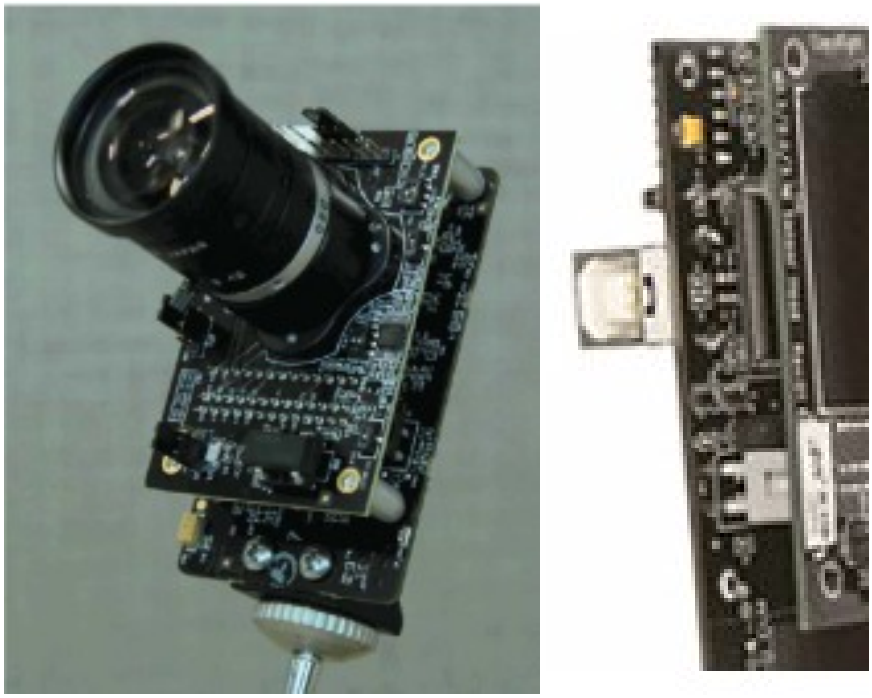

Figure 1.2 Aptina MT9P031 (left), view of the demo PCB's USB 2.0 interface connection (right)

V. Methodological Overview

To best meet the objectives of this research, the overall process has been divided into three major parts: (1) build calibration and correction models, (2) build software to process star images, and (3) acquire, process and analyze star images. This order of operations leads to a logical breakdown of image processing stages. This section will explain the significance of each of these stages and why they are important.

Figure 1.3 illustrates the high-level processing stages involved in acquiring an attitude solution from an image. Each of these processing stages also contains lower-level processes which will be discussed individually and in detail in subsequent chapters.

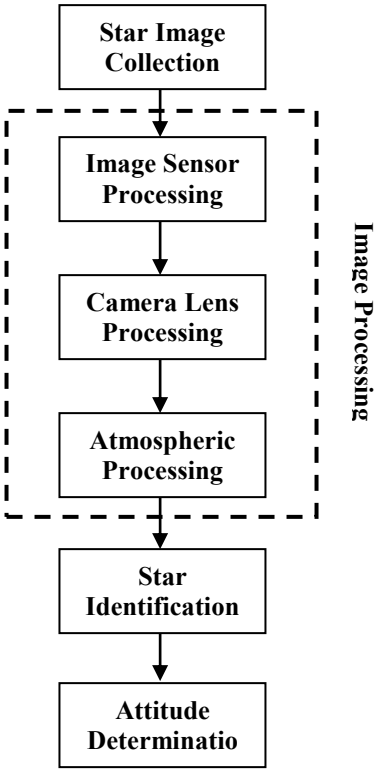

Figure 1.3 Flow diagram of the high-level processing stages involved in acquiring an image attitude solution

A. Star Image Collection

This stage is where all the necessary night sky images are acquired and stored ready for processing. Though it is not truly a part of image processing it is an important part to perform accurately, uniformly

and systematically. The quality and integrity of the images acquired during this stage will affect the final attitude results. This stage is detailed further in Chapter 2.

B. Image Sensor Processing

In this stage image spots and their locations are extracted from the image captured by the imaging sensor. This stage mostly revolves around sensor noise and deciphering the spots pertaining to stars to those originating from the background noise. Sensor noise contributes a major component of error in most digital imaging systems. For star cameras it also has the ability to make imaging a star impossible by completely overshadowing its signal. The calibration and physical correction models involved in this stage are detailed further in Chapter 3.

C. Camera Lens Processing

In this stage image spots and their locations are converted to three-dimensional pointing vectors. In this process lens errors are also dealt with. In a perfect system a lens would be manufactured to match desired optical characteristics without any imperfections. For large budget systems this is feasible; however, low-cost imagers do not have this luxury and therefore require calibration. The calibration steps and methodology of the processes involved in this stage are detailed further in Chapter 4.

D. Atmospheric Processing

This stage deals with errors originating from viewing the night sky on the surface of Earth. A correction is made to the spot pointing vectors to account for average atmospheric refraction. This stage is not required for star cameras in orbit; however, since the star images acquired in this research originate from the ground, atmospheric disturbances will affect the final attitude results and must be accounted for. The correction model and methodology used in this stage are detailed further in Chapter 5.

E. Star Identification

This stage is where spot pointing vectors, extracted from the image, are matched to their corresponding stars. Star identification is the principal objective of star cameras in which the camera measurements (vectors of starlight) are mapped to a truth (vectors of stars). This stage is a precursor to determining the

star camera's attitude. Further details regarding the implementation and usage of the star identification methods is discussed in Chapter 6.

F. Attitude Determination

This stage calculates the attitude and the attitude error of the star camera from the identified stars. This stage brings forward the numerical results necessary to compare and evaluate the two imagers' performance as star cameras. A detailed description of the methods used for evaluation is further discussed in Chpater 7.

CHAPTER 2

STAR IMAGE COLLECTION

I. Introduction

Collecting image data requires both an organized test setup and a robust method of keeping track of the acquired data. This chapter describes the methods and test setup used to collect star images from both the Aptina and Droid X2 cameras. A description of the datasets collected is also defined.

II. Methodology

This star image collection stage detailed in Figure 1.3 is expanded to into its sub-stages and shown in Figure 2.1. These sub-stages layout the methods used to acquire star images.

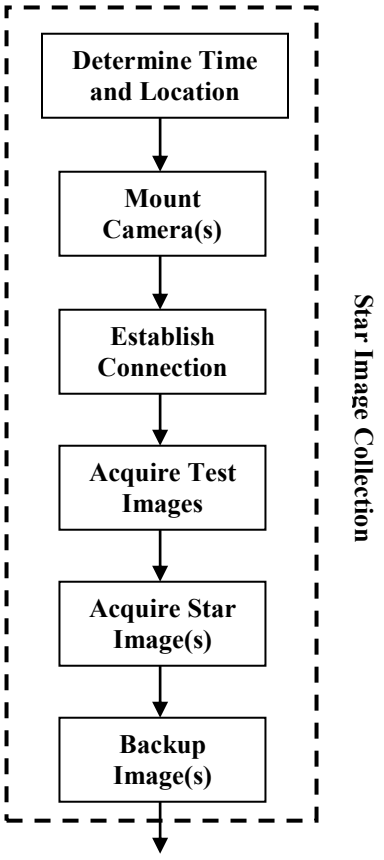

Figure 2.1 Star image collection flow diagram

A. Determine Time and Location

The times and locations of images collected for this research follow the methodology of this section. Their exact details are tabulated in Table 2.1.

When determining a suitable location the following things must be considered:

1. The sky is clear of obstruction within the camera's FOV (i.e. no overhanging trees, no clouds etc.).
2. The location has minimal air pollution. Correcting for air pollution is not a part of the atmospheric processing stage and therefore can drastically affect images by causing sporadic movements of the stars while at the same time reducing the viewable star intensity.
3. Be sure the mounting structure is stable and relatively level.

Weather is an important factor when planning image collection as it affects the ability of stars to appear in an image. The astronomer forecast service [16] provides daily weather predictions separated in intervals of one hour. The seven main weather components affecting star visibility are categorized and color coded to express the conditions at that time. These components include: cloud cover, transparency, seeing, darkness, wind, humidity and temperature. Details of how each of these affect viewing stars can be found within the service. Nights where this service predicts good weather were used to collect images.

B. Mount Cameras

Generally camera mounting would require attaching a camera to a tripod by a mounting screw. However, smartphones do not have a mounting point and even though the Aptina does it is desired that both cameras are mounted together on a fixed apparatus such that the relative pointing direction of each camera is close to the same. To achieve this, a tilting mounting plate was created allowing for both the Droid X2 and the Aptina camera to mount together. This mounting plate was then fixed on top of a table and tilted to the desired angular elevation.

C. Establish Connection

In general night-sky images are captured using either a delayed timer or a remote command to prevent movement of the camera during the exposure of the image. The same was done with both the Aptina and

Droid X2 cameras where a laptop was used as the remote system. Both devices were connected to the laptop by 15ft USB cables. The laptop's light was also shielded such that it did not interfere with the images.

Upon connecting the cameras to the laptop, two applications were launched: (1) DevWare from Aptina to control and capture images from the Aptina camera; (2) AndroidScreencast [17] a java application allowing remote control of the Droid X2 (this program requires root access to the phone, a procedure which is not discussed here).

D. Acquire Test Images

To guarantee the orientation of the camera matches that which is desired three to five test images are taken. These images are analyzed to verify that there are no obstructions in the viewing frame and also that the intended section of sky is viewable.

As a part of the test image step the proper camera settings are determined. Specifically when available the exposure time, focus, digital gain (ISO speed) and aperture are adjusted such that starlight luminaries appear on the test images as distinctive spots having few to no saturated pixels. When the proper settings cannot be determined datasets having different settings can be collected and analyzed at a later time. The Aptina camera allows control of all four of these parameters. The Droid X2 controls the exposure time and digital gain as an automated process contained within an exposure value adjustment, the focus has two distinct positions, infinity and auto, and the aperture is fixed as per the specification listed in Table 1.2.

E. Acquire Star Images

As neither the Aptina nor the Droid X2 have an automated way of capturing sequential images a timing system is required. For datasets requiring multiple sequential images the timing system followed a 1 to 3 minute capturing interval. For other general datasets where the mounting position is variable timing is not considered and thereby allowed images to be captured when needed.

F. Backup Images

Data loss leads to an unpleasant experience and therefore following all data collection campaigns all images were moved and redundantly backed up. The Droid X2 requires all images to be initially stored on

the phone and therefore backing up its data required the preliminary step of transferring the images from the phone onto another system.

This backup step also allowed for verification of all image and directory names and state in which they were saved. Any discrepancies were scrutinized.

III. Collected Images

Many image datasets were collected during the course of this research. The datasets described in Table 9.2 represent those used for analysis. All other datasets and additional details are listed in Appendix B.

Table 2.1 Collected datasets

Date	# of Images	Camera	Sequential	Latitude	Longitude	Directory Name
10/5/2012	170	Aptina	Yes	41.771948	-111.749832	2012_10_05
11/6/2012	252	Aptina	Yes	41.771948	-111.749832	2012_11_06
10/5/2012	213	Droid X2	Yes	41.771948	-111.749832	2012_10_05
11/6/2012	238	Droid X2	Yes	41.771948	-111.749832	2012_11_06
3/30/2012	18	Droid X2	No	38.77356	-109.58103	2012_3_30

IV. Summary

Following the methodology defined in this section allowed for the acquisition of repeatable, quality star images. The necessary time was taken to acquire the best possible images thereby preventing undesired errors from arising at later processing stages. Organization and preparation had key parts in collecting sufficient and useable data.

CHAPTER 3

IMAGE SENSOR PROCESSING

I. Introduction

The purpose of image sensor processing is to extract two-dimensional coordinates of spots from the raw image data. Ideally the extracted spot coordinates belong only to imaged stars; however, this does not always happen as the image sensor has induced noise which in some cases can completely overshadow the amplitude of the starlight. Besides noise, spot coordinates are also affected by the accuracy of their centroided spot pixel data. This chapter describes the techniques taken to both remove image noise and properly centroid spots.

II. Background

A. Factors Affecting Image Noise and Intensity

The ability to distinguish between what is being imaged and the inherent noise generated by imaging processes is a function of the signal-to-noise ratio (SNR). The SNR value for star images is defined by Eq. (3.1) where I_{S-RMS} is the root-mean-square of the pixel intensities of a starlight spot and I_{N-RMS} is the root-mean-square of all pixel intensities not part of a starlight spot.

$$SNR = \frac{I_{S-RMS}}{I_{N-RMS}} \tag{3.1}$$

1. Pixel Size

The size of a pixel determines the pixel's ability to collect photons. The larger the size the more photons it can collect and therefore the better the SNR for a given exposure. Individual pixels generate electrons by collecting photons. This generation is proportional to the sensor area and the quantum efficiency. Whereas noise power is proportional to the overall sensor area, noise voltage is proportional to the square root of the power and therefore the area. If the linear dimensions of a pixel are doubled, then SNR is doubled. The capacity of a pixel to hold electrons is also proportional to its area which directly affects the dynamic range.

2. Sensor Technology

There are two primary image sensor technologies CMOS and CCD. Since the early 2000's both technologies are comparable in noise yet before this point CCD's were considered to be the lower noise sensor of choice. CMOS technology is less expensive because it is easier to add new technologies to an originating chip design.

3. ISO Speed / Digital Hardware Gain

Depending on the camera and the control software either an ISO speed setting or a digital hardware gain setting is available. Both amplify the signal (along with the noise) at the pixel level. Therefore, the higher the gain (ISO speed) the more noise exists. Hardware gains differ from software gains in that a hardware gain is controlled by the pixel control circuitry prior to the image exposure and therefore differs for each pixel whereas a software gain is applied after the exposure to the entire sensor array. The amplified noise levels of a hardware gain are not linear like they are for a software gain [18].

4. Exposure Time

Short exposures with bright light tend to be less noisy than long exposures with dim light. This means that reciprocity does not work perfectly for sensor noise. To fully characterize a sensor, tests should be taken at the longest exposure time possible [19]. These long exposure times allow a broad spectrum of sensor noise to be extracted and analyzed. Short exposures would not capture the full spectrum of sensor noise and therefore a thorough understanding of the sensor noise would not be achievable.

5. Digital Processing

Image sensors generally have either an 8-, 10- or 12-bit analog-to-digital (A-to-D) converter. At 10 or 12 bits digitization noise is not normally an issue, even 8-bits is normally acceptable. However, when image conversion is performed which does not maintain the original bit-depth or lossy image compression is added then processing noise starts to become an issue. JPEG images which many consumer level point-and-shoot cameras use as the primary image format contains both lossy compression and bit-depth down-conversion to normally 8-bits per color-channel (24-bit color). The bit-depth down-conversion does not really generate noise (more of a blending of data) but the compression does by introducing what is known

as compression artifacts. These artifacts are categorized into the following five groups: ringing, contouring, posterizing, staircase noise along curved edges and blockiness in areas of high image activity [19].

6. Raw conversion

Lowpass filters to reduce noise and image frequency adjustments to improve sharpness are often applied during the raw image conversion process. This is normally true even if these setting are turned off because it reflects how the manufacturer best feels the images should be viewed. This can affect how a camera's intrinsic properties are measured and therefore directly affects the accuracy of a camera's calibration [19].

7. Sensor Quantum Efficiency

A sensor's quantum efficiency is defined as the percentage of photons that generate electron-hole pairs (the primary principle behind a signal voltage). There are two components of the quantum efficiency: the internal and the external. The internal quantum efficiency refers to the percentage of photons which do not reflect off or pass through the detector. The external quantum efficiency refers to the percentage of photons that generate electron-hole pairs collected by the electrodes of the detector [20],[21]. The combination of the two form the sensors overall quantum efficiency. The higher the value the greater the signal generated by a given number of photons and the higher the SNR.

8. Dark Current

An image sensor's dark current is a reference to the small electric current that flows through the sensor even when no photons are present. It is due to the random generation of electrons which are moved by a high electric field such as that created by the chips underlying circuitry. Dark current can be estimated by taking the average of a series of images while the lens cap is on. In reality the dark current varies every time the sensor is turned on and off, which is why many photographers will take a series of dark images before they begin taking images [20].

9. Dead / Hot Pixels

Dead and hot pixels refer to pixels which do not produce a signal and those which always produce the same signal no matter how many photons are present respectively. In both cases this affects how pixel data is interpreted during image processing which is why these pixels are normally corrected either internally through onboard programmable hardware or through external software before image processing. Many modern sensors are also starting to include embedded circuitry on the sensor which detects and corrects these errors [22].

B. Noise Removal Techniques

Noise exists in every digital image and is most prominent in low-light images. Various techniques of removing noise from an image are discussed below. It is desired to find a technique or combination thereof which helps distinguish star light from the various sources of noise previously described.

1. Linear Smoothing

Linear smoothing involves convoluting the original image with a mask representing a low-pass filter or smoothing operation. In essence the convolution will bring the value of each pixel closer to that of its neighbor thereby reducing pixel-by-pixel high contrast points [23].

2. Anisotropic Diffusion

Anisotropic diffusion removes noise using a smoothing partial-differential-equation (PDE) similar to that of the heat equation. Using a diffusion coefficient designed to detect edges, noise can be removed without the blurring side effect of linear smoothing [24].

3. Signal Combination

Signal combination refers to the combination of one or more images taken of the exact same scene and normally in sequence. The combination of multiple images is done by averaging the pixel array values. This can both increases the SNR value and improve the dynamic range. Since many noise sources are random their values will not be the same image to image whereas the source of light photons should be

very similar. The SNR increases simply because averaging random noise reduces the intensity of the overall noise.

4. Thresholding

Thresholding represents the simplest method of noise removal. Magnitude dependent thresholding removes any signal value which is below the given threshold value. This can remove dark current and other low-intensity random noises. However, if the SNR value is low, such that the noise has a similar intensity to the signal, basic thresholding will not have a desirable effect. A standard threshold value for a quality star image (high SNR) is three times the root-mean-square (RMS) noise floor; for lower SNR images trial and error may be needed.

5. Limiting Spot Area

For images where the signal is generated from point sources, such as stars, putting upper, lower or both limits on the size of the spot pixel clusters can help distinguish these sources from noise, especially when the SNR value is low. To remove random small spotty noise this method defines a minimum spot pixel area. To remove heavy noise, such as that created by image banding or compression, this method defines a maximum spot pixel area. These two bounds combined can help separate a spots signal from the noise and imaging artifacts.

6. Spot Eccentricity

For images where the signal is generated from point sources, such as stars, comparing the planar shape of extracted spots to ellipses having low eccentricities can be useful. By building a bounding ellipse around a spot the ellipse's eccentricity can be used to represent the spot's shape. Setting a limiting value on this eccentricity allows extracted spots which are more linear in shape to be removed. This aids in removing banding and compression noises. This can also remove obstructions or other unwanted features from a set of spot clusters such as parts of the sun or moon.

7. Spot Flatness Factor

For images where the signal is generated from point sources of light, such as stars, categorizing spots based on their profile shape can be useful. The profile shape can be defined by the flatness factor which is the percent deviation of one standard deviation of the spot intensity to the spot's maximum value. Normal point light sources leave near-Gaussian intensity distributions on image sensors when properly focused and therefore will have flatness factors greater than 10 percent. This technique is useful in removing noise spots which contain pixels of near equal intensity.

C. Centroid Methods

To accurately obtain the two-dimensional spot coordinates from imaged stars, the illuminated pixel clusters defining spots must be centroided. Various centroid techniques exist and are discussed below [25].

1. Peak Value Centroiding

The peak value centroiding method takes the peak pixel value location in a spot pixel cluster as the spot cluster's centroid. This technique is only pixel level accurate and therefore does not yield a high level of accuarcy. However, if the angular resolution of a single pixel is greater than the angular spread of light from a single star this method is as accurate as any other technique. How a star's point-light-source is spread during imaging is discussed further in Chapter 4.

2. Area Centroiding

Area based centroiding involves calculating the spot pixel clusters geometric center or barycenter. For images the simplest way to calculate this form of centroid is to average all the x and y coordinates of the spot pixels in the cluster independently. The resulting average x and average y coordinate values are obtained and combine to form the centroid. This type of centroid is sub-pixel accurate and is used on either binary images or images where intensity variations are inaccurate. Equation (3.2) details the calculation process

$$\sigma_x = \frac{x_1 + x_2 + \cdots + x_n}{n}, \qquad \sigma_y = \frac{(y_1 + y_2 + \cdots + y_n)}{n} \tag{3.2}$$

where σ_x and σ_y represent the x and y coordinates of the centroid respectively and n represents the number of pixels in the spot cluster.

3. Intensity-Weighted Centroiding

An intensity-weighted centroid follows the methodology of a center-of-mass calculation where the individual pixel coordinates are weighted by the intensity of the pixel value. This type of centroid is sub-pixel accurate and is used on images where intensity variations are accurate. Equation (3.3) details the calculation process

$$\sigma_x = \frac{\sum_{i=1}^{n} P_i \times x_i}{\sum_{i=1}^{n} P_i}, \qquad \sigma_y = \frac{\sum_{i=1}^{n} P_i \times y_i}{\sum_{i=1}^{n} P_i} \tag{3.3}$$

where σ_x and σ_y represent the x and y coordinates of the centroid, n represents the number of pixels in the spot cluster, P represents the individual pixel values and x and y represent the individual pixel horizontal and vertical coordinates.

4. Predictive Centroiding

Predictive centroiding was developed specifically for star trackers and involves predicting centroids based on previous image data containing the same stars. The scheme is initialized by intensity-weighted centroids. All recurring images use a predictive processing method to predict future centroids. The predictive processing method requires attitude projection abilities such that as new images become available there is an attitude solution which can be used to predict where the stars should appear on the focal plane. This allows the extraction of spot cluster based on the positions where stars are predicted to be and therefore prevents the need of searching the entire sensor array for new spots every time there is a new image [26].

III. Methodology

This section will describe the methods and steps used to extract and centroid spot data from star images. As is seen in Figure 3.1 there are four main stages to this process. The stages described below all

follow the standard image coordinate system definition; the top-left corner of the image has the coordinate (0,0), x is positive left to right and y is positive top to bottom.

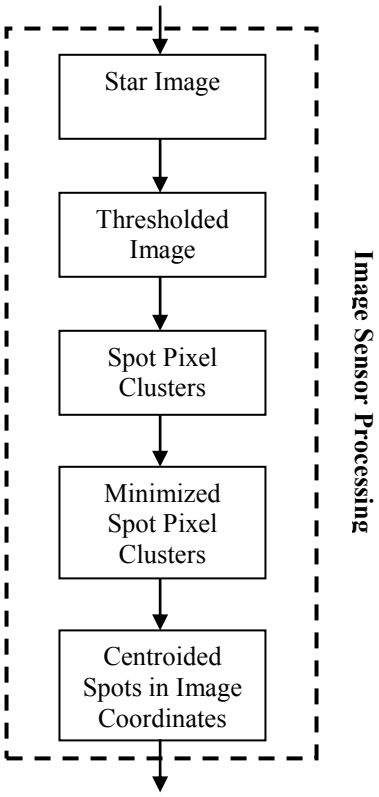

Figure 3.1 Flow diagram of the image sensor processing stages

A. Threshold Image

The first stage upon receiving image pixel data is to threshold the image. The threshold value is camera and scene dependent. Various values following the thresholding technique described earlier in this chapter will be used. For implementation MATLAB code utilizing logical indexing will be used to set pixels having intensities lower than the chosen threshold to zero.

B. Extract Spot Pixel Clusters

To extract spot pixels from the thresholded images two methods will be used. The first will utilize the MATLAB regionprops function from the Image Processing Toolbox. The second will use a custom MATLAB function which scans the image row-by-row until finding all non-zero pixels. As non-zero pixels are found they are added to spot clusters. Both techniques use 8-point boundaries on all pixels as seen in

Figure 3.2. If any of the 8 pixels surrounding the central pixel is non-zero, it is included in the same spot cluster as the central pixel.

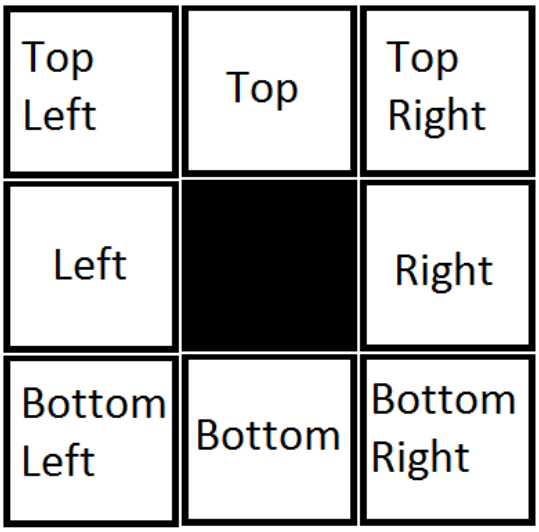

Figure 3.2 Spot pixel boundary criteria

The extracted spots will be categorized into tow groups: starlight spots and noise spots. Starlight spots are those spots created by starlight passing onto the sensor. Noise spots are those spots created by the noisy image capturing processes.

C. Apply Spot Size Selection Criteria

Spot size selection criteria are defined on a camera-by-camera base. After spots have been extracted for images various minimum spot areas (number of individual pixels in a spot cluster) are applied to help remove excess noise. In cases where extreme noise is visible maximum spot areas are also applied. For implementation MATLAB code using logical indexing is used to remove spot clusters outside the bounds of any given size criteria.

D. Centroid Spots

For spot centroiding the intensity-weighted centroid method described previously in this chapter will be implemented alongside the area based centroid method. Both techniques will be implemented as part of the regionprops function call in MATLAB called in the spot cluster extraction stage.

IV. Summary

Due to the testing nature of this research, algorithm and processing methods devised to only save time or processing power are not considered due to their higher degree of complexity to implement. This included techniques such as predictive centroiding and anisotropic diffusion. It can also be noted that both dark current and hot pixel data were not taken into account as it was found during preliminary observations that their values had little to no effect on results and that what was present was removed during the thresholding stage.

CHAPTER 4

CAMERA LENS PROCESSING

I. Introduction

The purpose of camera lens processing is to take the two-dimensional spot points and convert them to three-dimensional camera vectors. A perfect camera lens would pass a point from a viewed object straight through the lens without any loss in intensity or change in direction. This is a variation of what the pin-hole model describes. In actuality any form of lens will cause a change in both a light-rays direction and intensity. To account for this, lens systems must be calibrated. This chapter provides an understanding of the basic models used to describe lenses as well as the various methods used for calibration.

II. Background

A. Factors Affecting Lens Accuracy

There are many factors which affect how a lens performs. Lens performance is described by the degree in which a lens changes both a light-rays direction and intensity.

1. Aperture Setting

Although the camera aperture is not a part of the lens system, it does have the ability to change what is viewed through the lens. A camera's aperture limits the amount of light which can pass from the lens to the image sensor through a mechanical opening which is sometimes adjustable. The only affect on accuracy the aperture will have is if a different aperture is used for imaging than is used for calibration. This one-to-one match is important for a high accuracy to be achieved, especially with color imagers as chromatic aberration (discussed further in a moment) is intensified.

2. Refractive Index

The refractive index of a lens only affects the accuracy of the imaged intensities. Every time light is bent (refracted) it loses energy and therefore intensity. Due to the curved nature of lenses as light passes

through them various degrees of refraction occur and is dependent on the position and direction in which the light passed through the lens.

3. Optical Resolution

A digital camera system will have a given image sensor pixel resolution; however, a lens system also has its own limiting resolution. A lens can either be diffraction limited or aberration limited; both results in image blur though the resolution at which the blur occurs differs. Aberrations arise from geometric issues with the lens and can by principle be overcome by increasing the quality of the optics. Diffraction differs in that it is controlled by the wave nature of light and is determined by the finite aperture of the individual optics. When light passes through a lens it interferes with itself creating a ring-shaped pattern known as the Airy pattern. In a diffraction limited system the angular resolution is governed by Eq. (4.1) where θ is the optical resolution, λ is the wavelength of light and D is the diameter of the lens aperture [27].

$$\sin\theta = \frac{1.220\lambda}{D} \tag{4.1}$$

The optical resolution of a lens can be determined by the FWHM (full width at half maximum) value measured from the intensity pattern of point source of light. Light point sources when properly focused can be approximated by Gaussian distributions of which the FWHM is measured. If the FWHM is equal to the optical resolution calculated by Eq. (4.1) then the lens is considered to be diffraction limited otherwise the lens is aberration limited.

4. Vignetting

Vignetting, by definition, is a reduction in an image's intensity around the periphery compared to the center. Four primary types of vignetting can occur: (1) mechanical, (2) optical, (3) natural and (4) pixel. Mechanical vignetting is due to lens extensions or elements which protrude into the image FOV. Normally, this will cause radial shadows to appear in the image corners. Optical vignetting is a function of the camera aperture and results in a radial illumination falloff away from the image center. It is the result of obliquely incident light having a smaller apparent aperture to pass through. Natural vignetting is form of radial illumination falloff which follows the cosine-fourth law. As light moves away from the image center its

luminosity diminishes by a value approximated by $\cos^4$. Pixel vignetting is a form of radial illumination falloff due to the fact that individual pixels collecting light produce a stronger signal for light entering straight-on than for light entering obliquely. This form is similar by principle to optical vignetting.

5. Distortion

The most common type of lens distortion is radial distortion. Radial distortion is divided into two categories: barrel distortion and pincushion distortion. When combined it is called mustache distortion. Figure 4.1 shows these two types of radial distortion as well as their blended result.

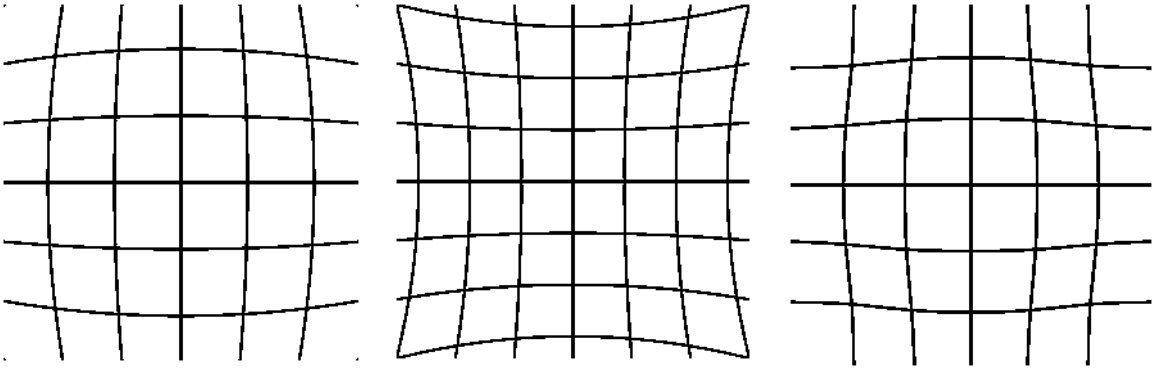

Figure 4.1 Distortion types: (left to right) barrel, pincushion and mustache

Distortion typically arises from the type and shape of a lens. Radial affects are often part of a spherical lenses design. Knowing how a lens system distorts an image allows a correction model to compensate for the distortion. Many forms of distortion can be modeled by various degree polynomials. Both barrel and pincushion distortion are quadratic in nature whereas their combination (mustache) is quartic. Other forms of radial distortion can be described by higher order polynomials such as those of the 6^{th} and 8^{th} degree, though their effects are generally negligible when compared to the primary barrel and pincushion distortions.

Tangential distortion is another form of distortion which is due to lens centering misalignments and other manufacturing defects involving compound (multiple lens element) lens systems. Unlike radial distortion tangential distortion cannot be modeled with pure polynomials and therefore has various definitions.

6. Chromatic Aberration

Lens aberrations describe the phenomenon in which different light-rays refract through a lens at different angles resulting in a locus of focal points instead of just one. Chromatic aberration is an alternative of this in which light-rays of various wavelengths refract through a lens at different angles resulting in a different focal point for each passing wavelength. This is also known as dispersion [28]. Only color imaging systems are able to account for this effect since they store pixel data for 3 different wavelengths.

B. Lens Models

Lens models describe lens geometry mathematically. The basic lens types and models include: (1) simple lenses, (2) the thin-lens model, (3) the pin-hole model, and (4) a 6^{th} order lens distortion model.

1. Simple Lenses

There are six basic types of simple lenses including: biconvex, plano-convex, positive meniscus, negative meniscus, plano-concave and biconcave; see Figure 4.2 [29]. There are three categories of simple lens, two of which are used in normal cameras: spherical and aspheric lenses. Camera lens systems are generally designed using various combinations these basic lens types and therefore their modeling requires cumulative models of each individual lens.

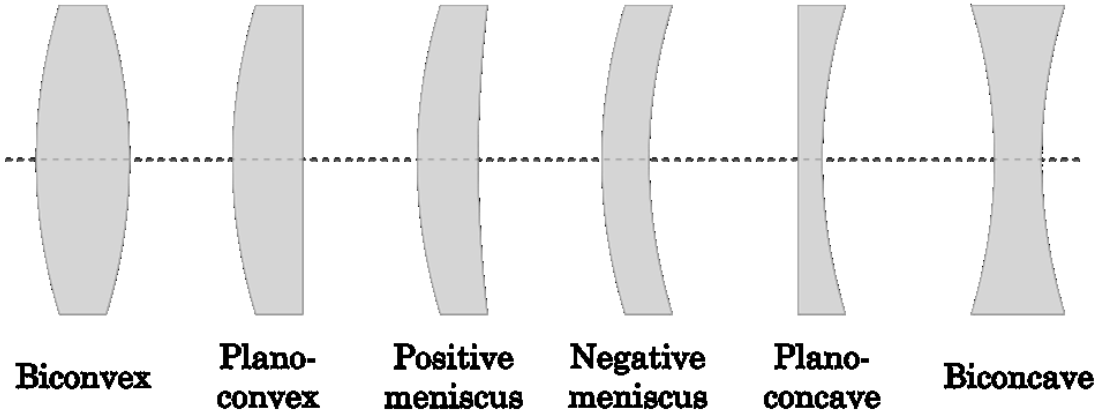

Figure 4.2 Types of simple lens

Spherical simple lenses are modeled by the lensmaker's equation shown in Eq. (4.2) where P is the lens power, f is the focal length of the lens, n is the refractive index of the lens material, R_1 is the radius of curvature of the lens surface closest to the light source, R_2 is the radius of curvature of the lens surface farthest from the light source and d is the thickness of the lens. This equation accounts for the optical effects due to both the lens thickness and refractive properties.

$$P = \frac{1}{f} = (n-1)\left[\frac{1}{R_1} - \frac{1}{R_2} + \frac{(n-1)d}{nR_1R_2}\right] \tag{4.2}$$

Figure 4.3 shows how a biconvex lens focuses collimated rays of light. It demonstrates how all the rays of a light column merge together to a point which is thus called the focal point. The focal point lies on the optical axis, a line following the path of the light ray which is neither reflected nor refracted when passing through the lens. The optical center is the location on the lens where the optical axis passes through.

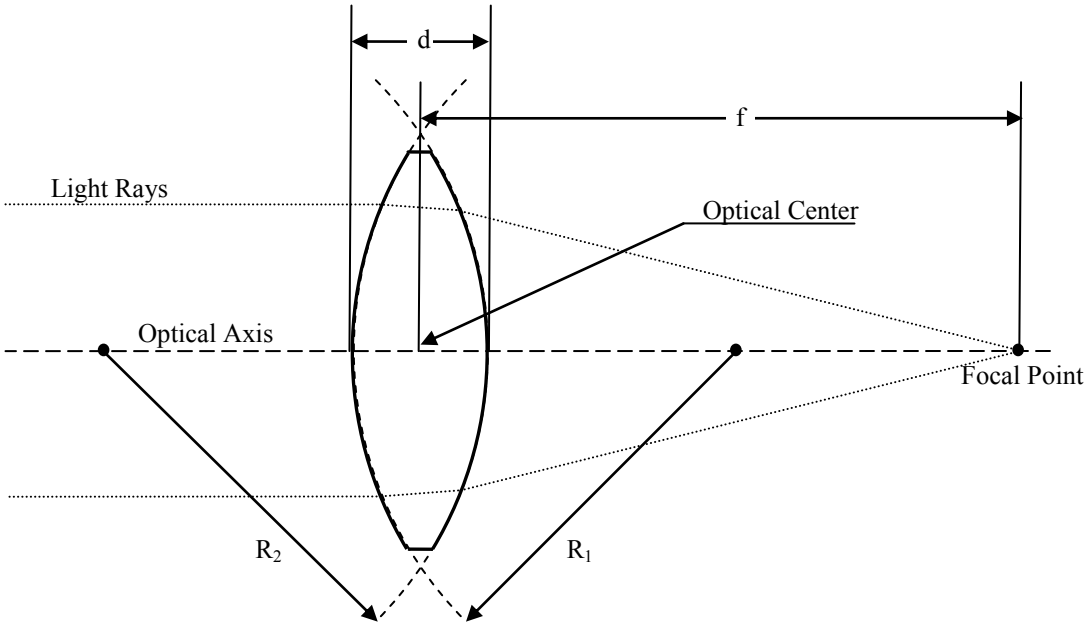

Figure 4.3 Biconvex lens focusing collimated rays of light

Aspheric simple lens are normally modeled using the surface equation Eq. (4.3) where z(r) represents the sag —the z-component of the displacement of the surface from the vertex at a distance r from the

optical axis. The coefficients α_i account for surface deviations away from the axially symmetric quadric surface specified by R and k.

$$z(r) = \frac{r^2}{R\left(1 + \sqrt{\left(1 - \frac{(1+k)r^2}{R^2}\right)}\right)} + \alpha_1 r^2 + \alpha_2 r^4 + \alpha_3 r^6 + \cdots, \tag{4.3}$$

Many other techniques can be used to design and model aspheric lenses [30]. The primary benefits of aspheric lenses over spherical lenses are that they can be designed to have fewer aberrational defects and they are smaller in size. These benefits come at the price of manufacturing difficulty due to their highly abnormal surface shapes.

2. Thin-Lens Model

For lenses where $d \ll f$ (the lens thickness is much smaller than the focal length) the thin-lens simplification can be made. The thin-lens equation is detailed in Eq. (4.4) where f is the focal length, n is the refractive index of the lens material, R_1 is the radius of curvature of the lens surface closest to the light source and R_2 is the radius of curvature of the lens surface farthest from the light source. This equation neglects all internal lens refractions deeming them negligible since the lens is considered thin compared to its optical properties.

$$\frac{1}{f} \approx (n-1)\left[\frac{1}{R_1} - \frac{1}{R_2}\right] \tag{4.4}$$

Figure 4.4 demonstrates how the distances from the object to the lens and the lens to the image are related in a thin lens. From this and Eq. (4.5) it is understood that an object at a distance S_2 away from the lens will become focused on the image positioned a distance S_1 away from the lens.

$$\frac{1}{S_1} + \frac{1}{S_2} = \frac{1}{f} \tag{4.5}$$

It is important to note that when an object is placed an infinite distance away from the lens it will be in focus when $S_2 = f$, the focal length. Also, when $S_1 < f$ the lens system acts in a magnifying nature in that the object appears larger on the image than in real life.

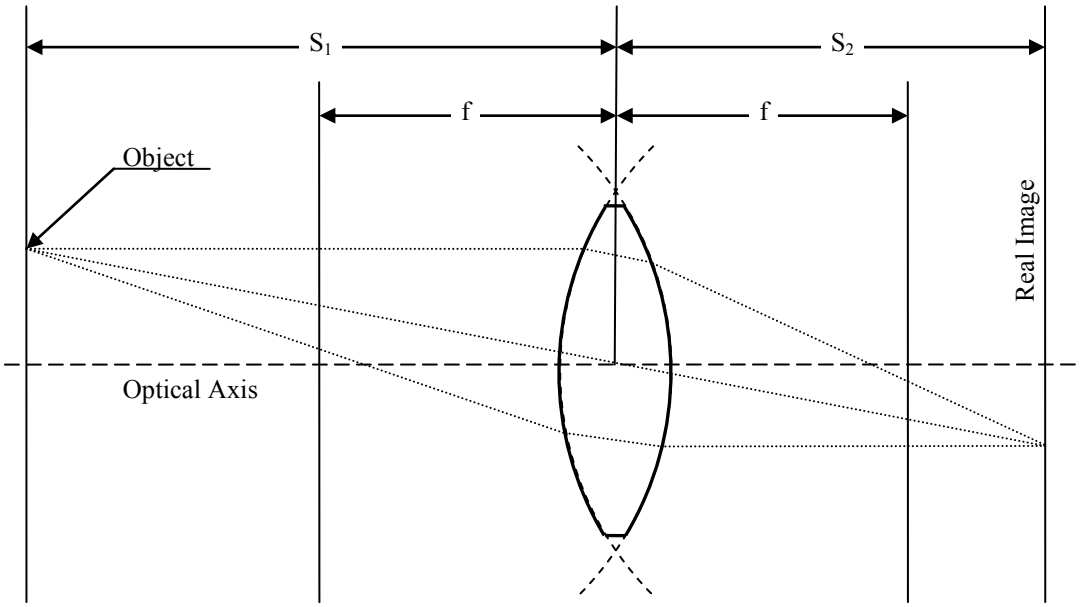

Figure 4.4 Biconvex lens focusing an object at a distance S_1 away from the lens

3. Pin-Hole Model

The pin-hole camera model is the simplest of all camera models. Instead of a normal lens the model uses a tiny hole of an infinitely small diameter to pass light into the camera. A camera modeled this way is error free as there is no medium which the light must pass through and therefore the light remains undisturbed. Figure 4.5 shows a visual representation of this model. The image on the image plane represents a two-dimensional view of the three-dimensional world which has been mirrored over both of its planar axes (the X_p and the Y_p). To obtain the true view of the image the image can either be rotated by 180 degrees or mirrored over both of its planar axes; both yield the same result. The main principle behind the pin-hole model which differs from other lens models is that from any single point, either in space or on an object, only one ray of light can pass from it to the image plane.

The pin-hole model is normally written in terms of the camera matrix, Eq. (4.6). The camera matrix is a representation of a camera's basic intrinsic properties where f_c is the cameras focal length (the

components of which are equal when an image sensor's pixels are square) and c_c is the principal point (the point at which the optical axis intersects the image plane).

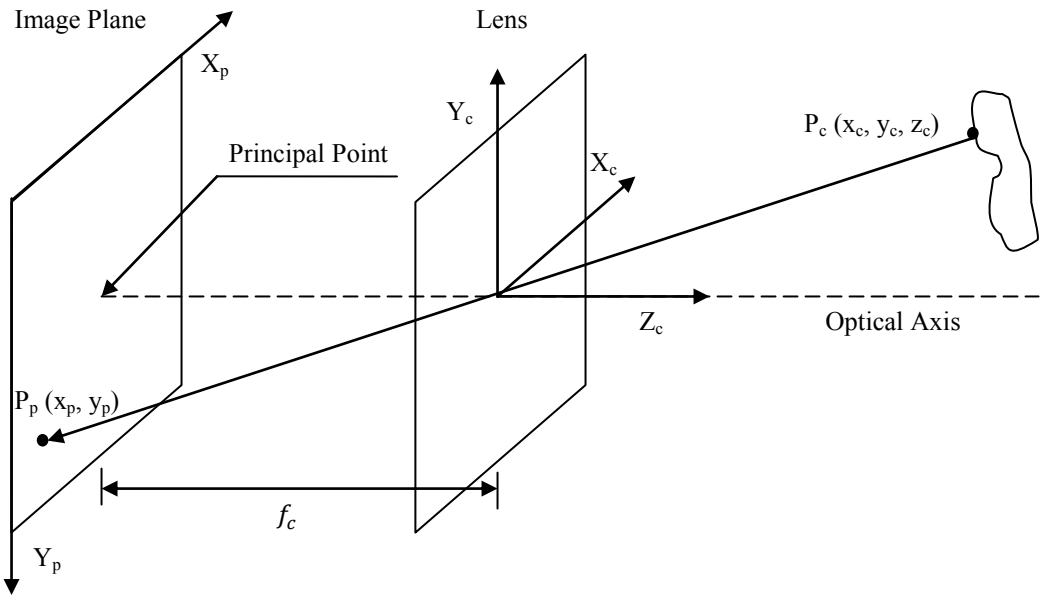

Figure 4.5 Pin-Hole Camera Model

$$KK = \begin{bmatrix} f_c(1) & 0 & c_c(1) \\ 0 & f_c(2) & c_c(2) \\ 0 & 0 & 1 \end{bmatrix} \tag{4.6}$$

When post-multiplied by a three-dimensional spatial vector, representing a point in space, a two-dimensional image coordinate is obtained. By convention the spatial vector is normalized by its Z_c component as seen in Eq. (4.7).

$$x_n = \begin{bmatrix} \frac{x_c}{z_c} \\ \frac{y_c}{z_c} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{4.7}$$

Once normalized the image plane coordinates are calculated by matrix multiplication as shown in Eq. (4.8).

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = KK \begin{bmatrix} x_n(1) \\ x_n(2) \\ 1 \end{bmatrix} \tag{4.8}$$

Equation (4.8) demonstrates the use of homogenous coordinates and allows the mapping of three-dimensional space to two-dimensional space to be performed by linear systems.

4. 6^{th} Order Distortion Model

Since distortion is a lens property which can be modeled, many lens models will include a distortion model. A common technique is to start from the pin-hole camera model and add distortion terms to it. The reason other lens models are not commonly used originates from the fact that the pin-hole model describes a perfect camera in the simplest form.

The following method adds both radial and tangential distortion components to the pin-hole camera model [31],[32]. Five distortion coefficients are used: three radial and two tangential. Equations (4.9) & (4.10) detail the radial and tangential components respectively where k_c represents the distortion coefficients, x and y represent the normalized three-dimensional planar components as defined in Eq. (4.7) and r is defined by $r^2 = x^2 + y^2$. Let it be noted that the 1st, 2nd and 5th components of k_c represent the radial distortion coefficients and the 3rd and 4th components of k_c represent the tangential distortion coefficients.

$$d_r = (1 + k_c(1)r^2 + k_c(2)r^4 + k_c(5)r^6) \tag{4.9}$$

$$d_t = \begin{bmatrix} 2k_c(3)xy + k_c(4)(r^2 + 2x^2) \\ 2k_c(4)xy + k_c(3)(r^2 + 2y^2) \end{bmatrix} \tag{4.10}$$

x_d represents the distorted normalized point coordinates and is defined by Eq. (4.11).

$$x_d = \begin{bmatrix} x_d(1) \\ x_d(2) \end{bmatrix} = d_r x_n + d_t \tag{4.11}$$

In addition to lens distortion, pixel skew can also be easily accounted for and is represented by α_c (the value of which is zero for rectangular image sensor pixels). With the addition of skew the camera matrix is now defined by Eq. (4.12).

$$KK = \begin{bmatrix} f_c(1) & \alpha_c f_c(1) & c_c(1) \\ 0 & f_c(2) & c_c(2) \\ 0 & 0 & 1 \end{bmatrix} \tag{4.12}$$

Replacing the standard pin-hole model normalized pixel coordinates with those defined in Eq. (4.11) the new distorted image coordinates x_p and y_p are found by Eq. (4.13) which continues the use of homogenous coordinates

$$\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = KK \begin{bmatrix} x_d(1) \\ x_d(2) \\ 1 \end{bmatrix} \tag{4.13}$$

C. Lens Calibration Methods

Lenses can be calibrated for both intensity and geometric errors. Various calibration methods exist.

1. Intensity Calibration

As mentioned previously camera aperture, vignetting sources and the refractive index of the lens can affect light's intensity as it passes through the lens system. To account for the nonuniformities in the intensity loss a flat-field correction is normally applied. A flat-field represents a lens map of the intensity fluctuations incurred by light as it passes through a lens. This intensity map is generally applied to a given image sensor and normalized such that when the intensity map array is multiplied by the signal array the intensities of the signal will match those prior to passing through the lens. Flat-field maps made with a complete camera system will also account for pixel-by-pixel sensor sensitivity differences.

2. Geometric Calibration

Many techniques exist for calibrating the geometry of a camera. Most of these methods involve processing a series of images to extract the desired camera and lens parameters. One such method is an extension of the 6^{th} order lens distortion model defined previously. This method uses images of checkerboard grids to converge on a camera's intrinsic and extrinsic parameters by using the definable checkerboard square intersections as known points. Normally, twenty or more images are used in this process where the checkerboard appears in various orientations. Once the checkerboard square intersections are extracted from all the images the camera parameters are initialized by using a closed-form solution to

the pin-hole camera model without distortion. Once initialized a non-linear optimization is performed to solve for the camera parameters by minimizing the reprojection error. The optimization includes 9 intrinsic parameters (focal length, principal point and the distortion coefficients) and 6 x (number of images) extrinsic parameters (mounting locations and rotations). The optimization process uses an iterative gradient descent method with an explicit computation of the Jacobian matrix. The process yields all the optimized intrinsic and extrinsic camera parameters and their corresponding uncertainties [33]–[37].

3. Adaptive Geometric Calibration

For spacecraft running attitude data through a Kalman filter, the adaptive lens calibration methods are both easily implemented and can yield an improvement in accuracy. Since star cameras are mechanical systems the vibrations and shocks they are subjected to during launch can potentially alter the results of a ground based calibration. Adaptive lens calibration methods include various camera parameters as states of the onboard Kalman filter. A camera's intrinsic parameters are always states: the focal length and the principal point. They may also include various lens distortion parameters and lens and camera mounting positions (known as extrinsic camera parameters). As the star camera operates, the Kalman filter will converge on the best possible values for each of the included states [12],[38]. Generally, ground based calibration results will be used to initialize the camera states.

III. Methodology

This section will describe the methods and steps used to take the two-dimensional spot points and convert them to three-dimensional camera vectors. The chosen camera model implemented for this task will be described and a discussion will follow as to why and how this model was selected. Figure 4.6 shows the stages required to arrive at the required camera vectors.

The chosen geometric camera model matches that of the 6^{th} order distortion model defined in a previous section. The decision to use this model over other models was made because it allows for the imperfections of a lens to be accounted for thereby resulting in fewer errors due to the camera being present in the final spot vectors. Using methods which only account for the focal length and principal point allows

all other lens variabilities to appear in the resulting spot position vectors. This is not desirable when one of the overall goals is to determine the repeatable accuracies of cameras.

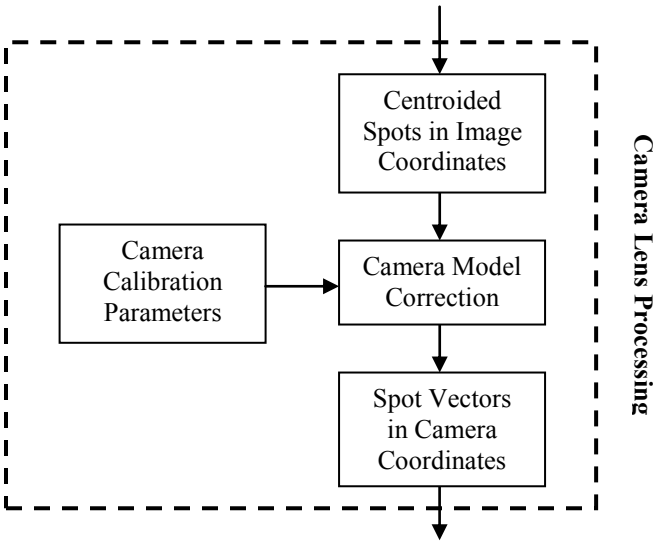

Figure 4.6 Flow diagram of the camera lens processing stages

Intensity fluctuations and losses due to lens imperfections, aperture and vignetting are not accounted for. Preliminary tests showed that image thresholding, described in Chapter 3, provides adequate spot data for this research. Though spot intensities are not uniform due to these unaccounted effects the spot position data was deemed suitably accurate to fulfill these research objectives.

To provide the required camera parameters to the camera model, the previously described geometric calibration process involving checkerboard images will be implemented. An open source toolbox for MATLAB which implements this particular geometric calibration process is available from both the University of Oulu and Intel [39],[40]. This toolbox will be used to obtain the required camera parameters and their uncertainties. The results of this process are shown in the next section.

IV. Calibration Methodology and Results

This section presents the camera calibration results for both the Droid X2 and the Aptina MT9P031 camera systems. As noted in the previous section the calibration toolbox for MATLAB was used for all calibration processes. To guarantee proper calibration of the cameras, checkerboards having three different

size grids were used and their results compared. The idea was that a finer checkerboard grid would yield a more accurate solution than a course grid. However, a limitation was met and the images of the finest grid were deemed useless due to selection limitations within the calibration software. Therefore, results were obtained for both cameras using two different grids: a 32mm grid and a 12.75mm grid. The checkerboards themselves were made from grids printed on a laser printer and adhered to a flat particulate board. This provided both an accurate, uniform checkerboard grid and the necessary rigidity to guarantee a flat surface.

A. Perform the Calibration

To perform the calibration the steps listed in the calibration toolbox's instructions were followed [41].

1. Acquiring Calibration Images

To begin, images must be collected for each calibration dataset. The toolbox recommends twenty or more images per dataset. Table 4.1 details for each camera the grid size and number of images taken for the calibration datasets. As per the requirements of this type of calibration the images acquired must have various different checkerboard orientations. It is important to get images at as many different orientations as possible while still maintaining a clear checkerboard image. If the orientation angle of the checkerboard is too steep the grid extraction phase of the toolbox will yield unacceptable results. It was found that the best results were obtained when the checkerboard was at no more than a 45 degree viewing angle. Figure 4.7 and Figure 4.8 show examples of the multiple orientations used for a dataset as viewed from the camera and world reference frames respectively.

Table 4.1 Geometric calibration datasets

Camera	Grid Size(mm)	# of Images
Droid X2	6	25
Droid X2	12.5	22
Aptina MT9P031	6	26
Aptina MT9P031	12.5	20

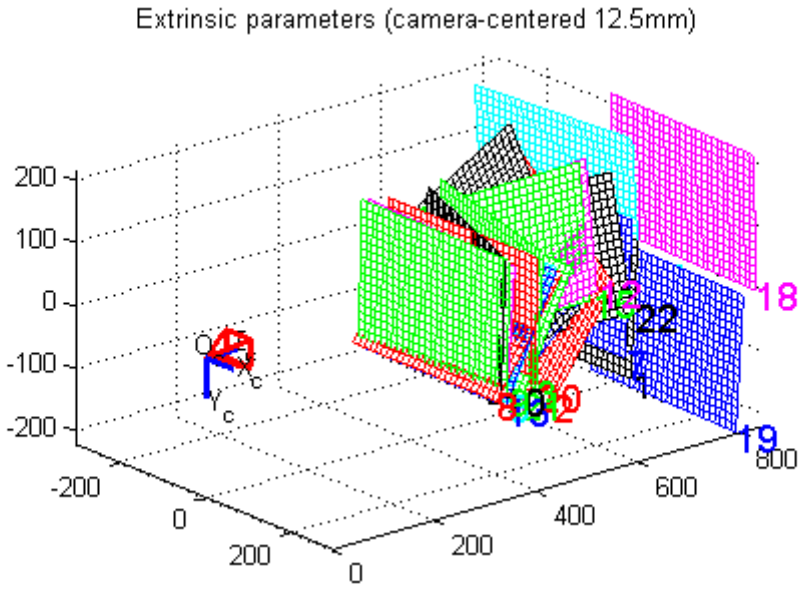

Figure 4.7 Calibration orientations as viewed from the camera reference frame

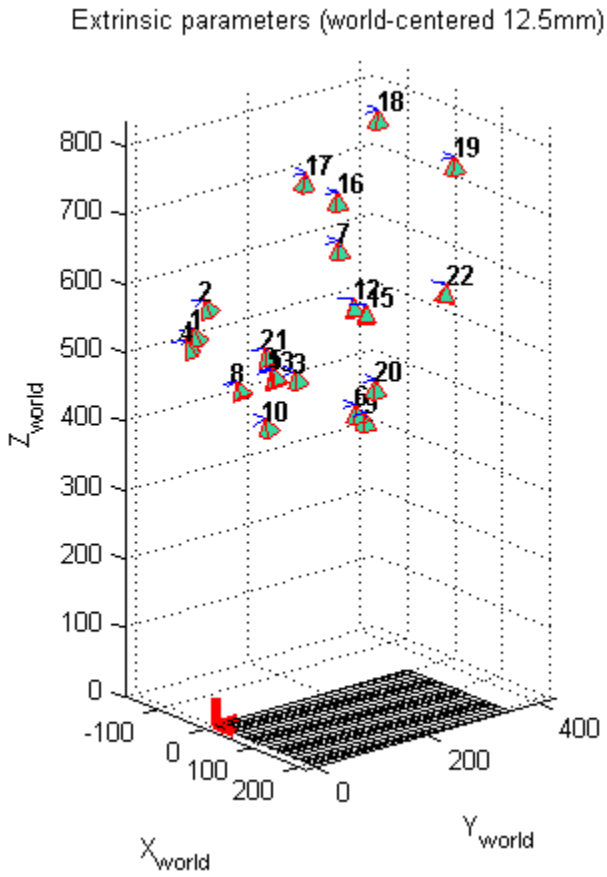

Figure 4.8 Calibration orientations as viewed from the world reference frame

2. Process Calibration Images

After a dataset of images is acquired the images must be renamed such that the prefix of all the images is identical and the suffix is a number incrementing from 1 to the number of images in the dataset. With the images renamed the toolbox will now read in the image. Once read in the grid corner extraction process is started which involves manually selecting the outer four grid corners (leaving a border of squares around the perimeter) of each image starting from the origin and moving in a clockwise fashion. The origin is a user choice which must stay constant. For these calibrations the origin border square was marked with an X on the checkerboard such that it was easy to identify in each image. While the corners of each image are extracted the toolbox automatically finds the intersections of all the grid squares which are used for the calibration. Examples of the corner extraction and grid square intersections results are seen in Figure 4.9 and Figure 4.10, respectively.

After the grid intersection points of all dataset images have been found the toolbox can begin to calibrate the camera. The preliminary calibration results are displayed in the MATLAB command line. At this point the magnitudes of the distortion parameter uncertainties are checked against their respective parameter values to verify that the 3-sigma values are not greater than the parameter values. Where this problem exists it is resolved by omitting the calculation of the distortion parameter and re-running the calibration. Now the reprojection error is analyzed for images which have excessive error outside the clustered bounds. For this problem the calibration is re-run with the problem images omitted. Figure 4.11 shows an example of this type of problem. These among other post-calibration checks and corrections were made in accordance with the toolbox instructions [41] on how to minimize the calibration error.

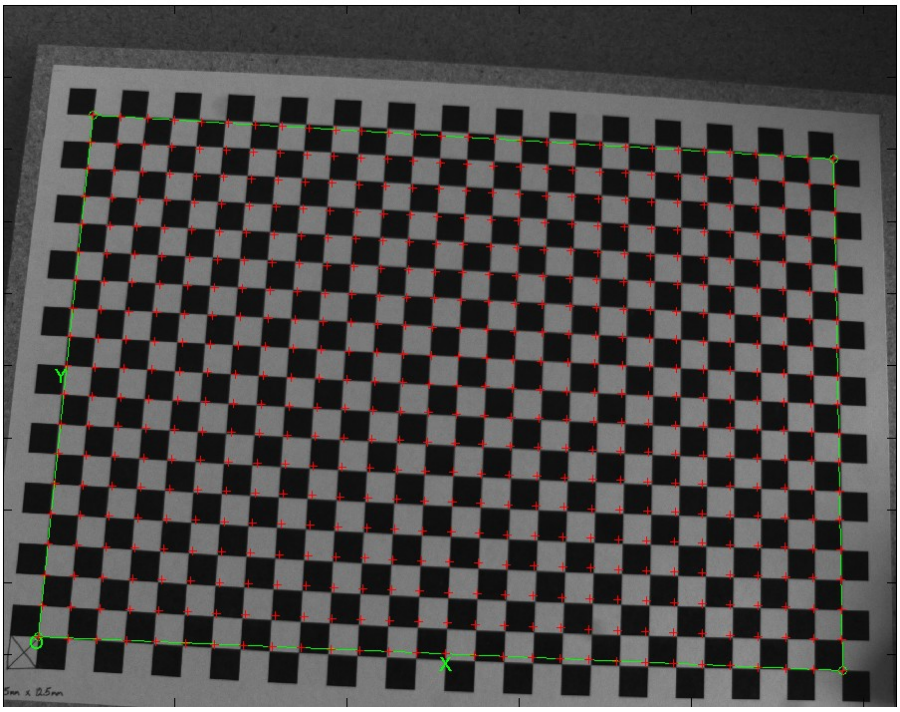

Figure 4.9 Example of the calibration toolbox corner extraction process

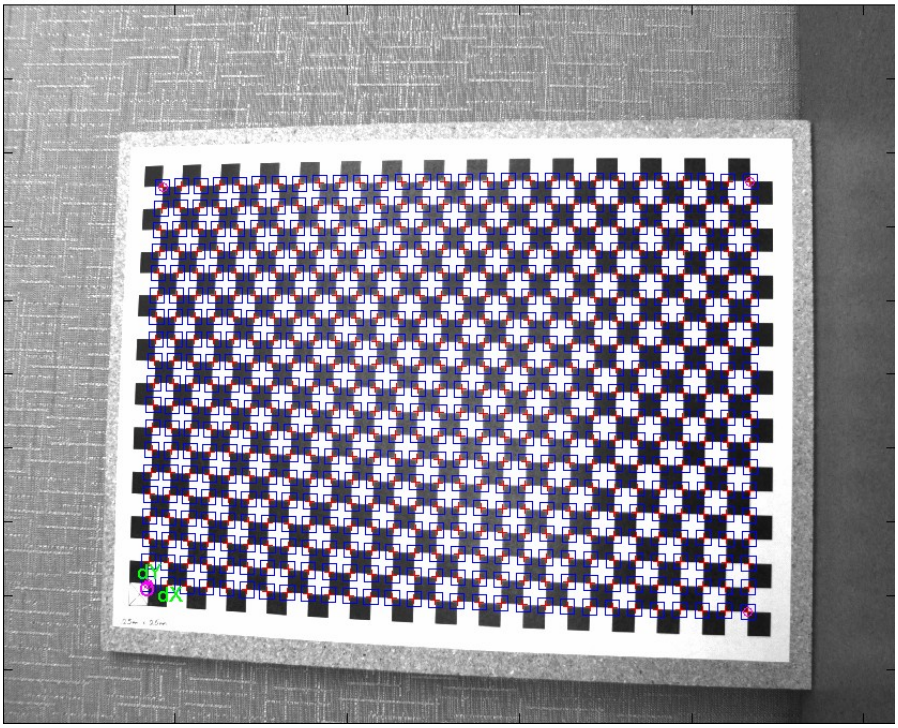

Figure 4.10 Example of the calibration toolbox grid square intersection results

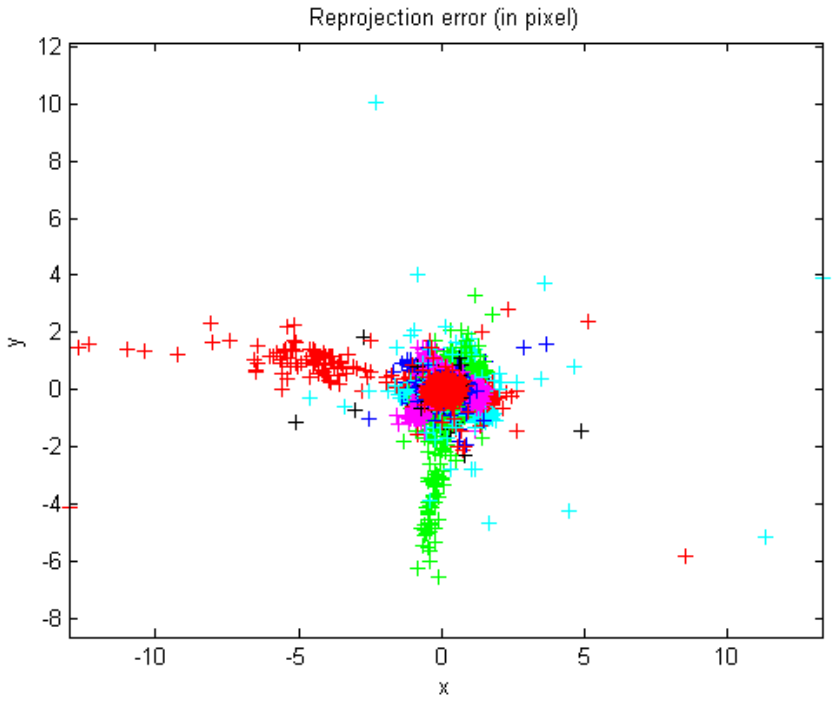

Figure 4.11 Example of calibration images causing excessive error

B. Analysis of the Calibration Results

Following the various camera calibrations their results were organized and compared to identify calibration trends and any visible camera limitations.

1. Droid X2 12.5mm Grid Calibration Results

Table 4.2 represents the final camera calibration results for the Droid X2 camera using a 12.5mm grid. As shown the overall pixel error lies within 0.935 pixels of resolution which for the Droid X2 is equivalent to 0.8469 arc-minutes of accuracy. It is also shown that the distortion of this calibration is primarily due to radial effects as the tangential components are small. It can also be seen that the last component of radial distortion and the first tangential component were manually set to zero due to their errors being larger than their respective values.

Figure 4.12 and Figure 4.13 show the reprojection error and overall lens distortion effects for the 12.5mm grid calibration, respectively. The reprojection error consists of a uniform statistical cluster of points with few outliers. This type of point cluster represents a proper calibration and shows uniformity

between all input calibration images. The lens distortion effects consist of a relatively uniform central lens area with significant distortion appearing in the corners of the focal plane. It is also seen that the principal point is slightly off-center yielding more distortion on the left half of the image. The distortion effects also show slightly more tangential distortion on the right half of the lens in the form of a non-uniform central contour. This contour however is the effect of a highly undistorted lens center causing minor distortions to show through.

Table 4.2 Droid X2 camera calibration parameters resulting from a 12.5mm grid

Parameter Name	Value	Standard Error (1σ)
Overall Pixel Error	(0, 0) pixels	+/- (0.935, 0.6744) pixels
Focal Length	(3510.99, 3511.11) pixels	+/- (5.096, 5.033) pixels
Principal Point	(1646.59, 1221.46) pixels	+/- (2.379, 2.513) pixels
Skew	0	+/- 0
Radial Coefficients	(0.1245,-0.5885, 0)	+/- (0.002534, 0.01148, 0)
Tangential Coefficients	(0, 0.002903)	+/- (0, 0.0002204)

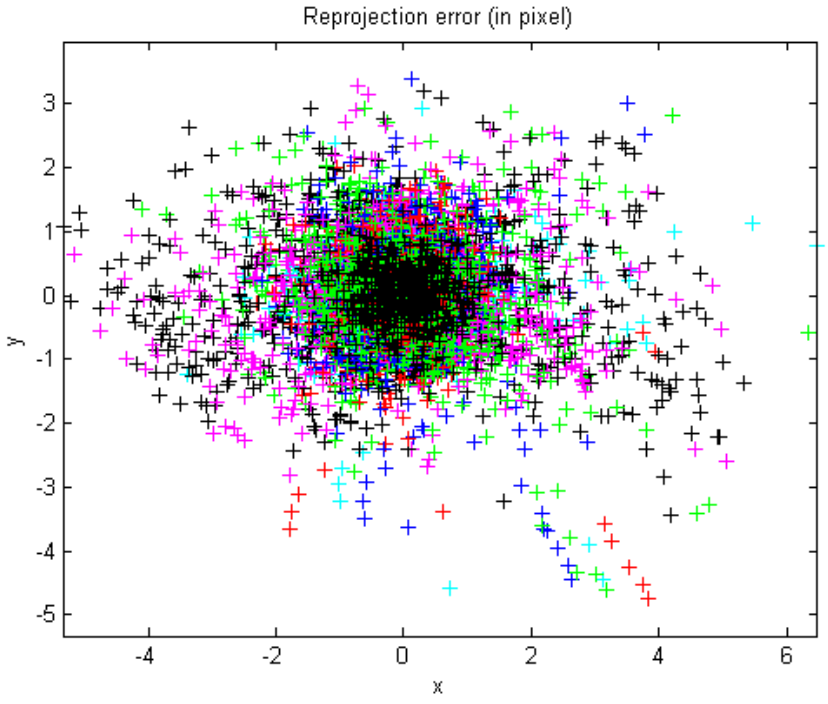

Figure 4.12 Droid X2 12.5mm checkerboard reprojection error

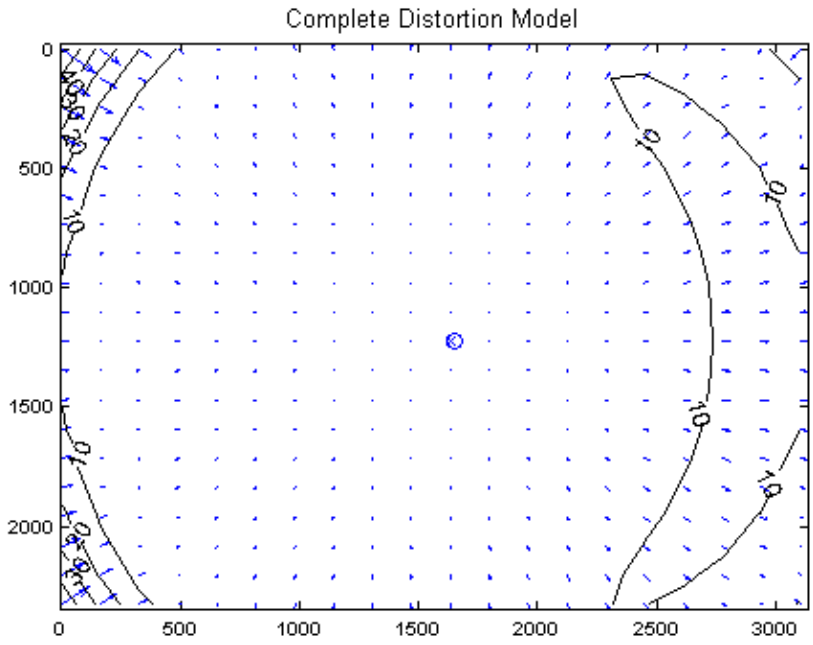

Figure 4.13 Droid X2 12.5mm checkerboard overall distortion effects showing displacement in pixels

2. Droid X2 6mm Grid Calibration Result

Table 4.3 represents the final camera calibration results for the Droid X2 camera using a 6mm grid. As shown the overall pixel error lies within 0.642 pixels of resolution which for the Droid X2 is equivalent to 0.5815 arc-minutes of accuracy. It is also shown that the distortion of this calibration is due to only radial effects as the tangential components were manually set to zero due to their standard errors being greater than their respective values. The last radial coefficient component was also manual set to zero for the same reason.

Table 4.3 Droid X2 camera calibration parameters resulting from a 6mm grid

Parameter Name	Value	Standard Error (1σ)
Overall Pixel Error	(0, 0) pixels	+/- (0.6419, 0.5182) pixels
Focal Length	(3520.75, 3521.15) pixels	+/- (1.142, 1.118) pixels
Principal Point	(1620.66, 1214.52) pixels	+/- (0.3997, 0.6557) pixels
Skew	0	+/- 0
Radial Coefficients	(0.1043, -0.4592, 0)	+/- (0.0008596, 0.003787, 0)
Tangential Coefficients	(0, 0)	+/- (0,0)

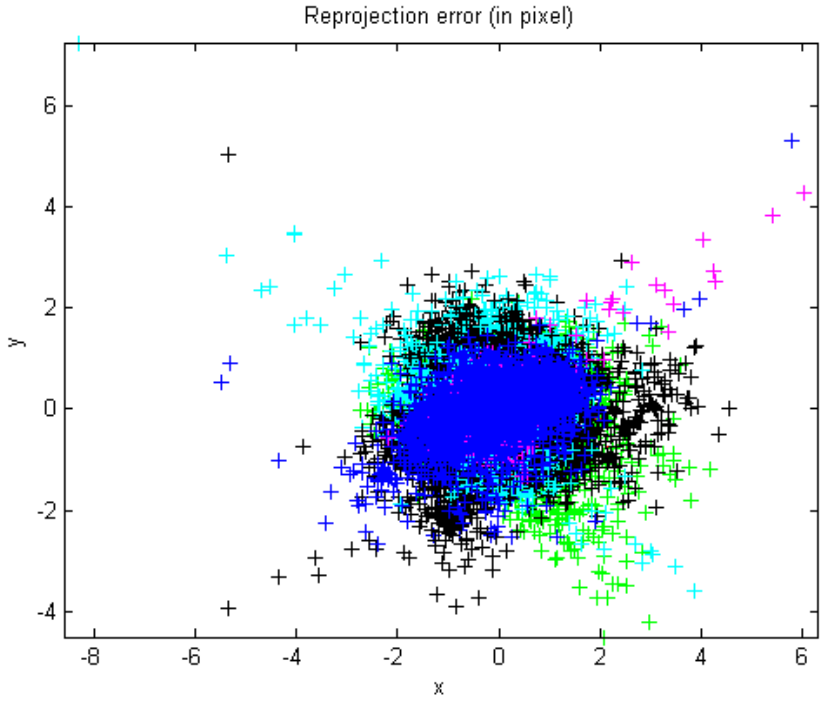

Figure 4.14 Droid X2 6mm checkerboard reprojection error

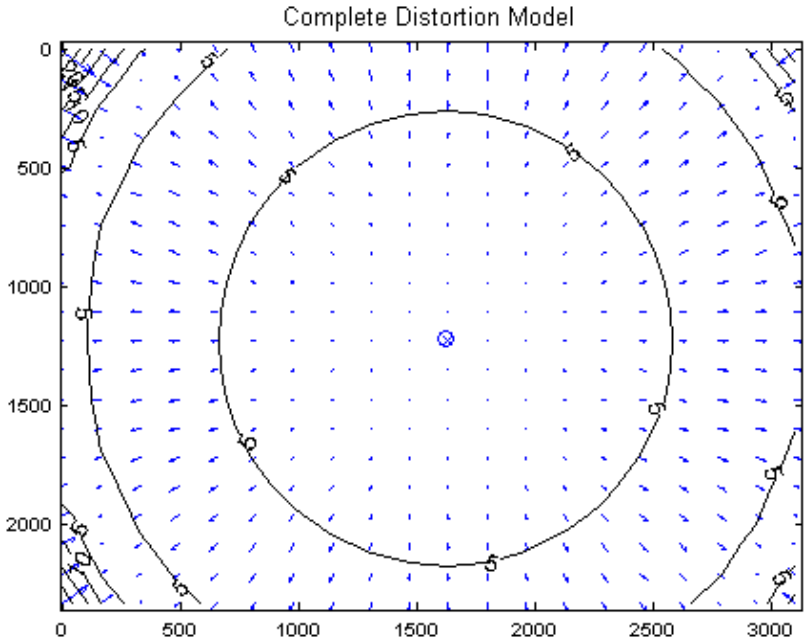

Figure 4.15 Droid X2 6mm checkerboard overall distortion effects showing displacement in pixels

Figure 4.14 and Figure 4.15 show the reprojection error and overall lens distortion effects for the 6mm grid calibration, respectively. The reprojection error consists of a uniform statistical cluster of points with few outliers. Though a few images resulted in cross-axial error which if removed should yield a higher accuracy calibration, when tested the effects were negligible and therefore it was decided that the higher number of calibration points would be better. The lens distortion effects consist of a relatively uniform central lens area with significant distortion appearing in the corners of the focal plane. It is also seen that the principal point is slightly off-center yielding more distortion on the left half of the image.

3. Aptina 12.5mm Grid Calibration Results

Table 4.4 represents the final camera calibration results for the Aptina camera using a 12.5mm grid. As shown the overall pixel error lies within 0.259 pixels of resolution which for the Aptina is equivalent to 0.2953 arc-minutes of accuracy. It is also shown that the distortion of this calibration is primarily due to radial effects as the tangential components are small. It can also be seen that the last components both the radial and tangential coefficients were manually set to zero since their errors were larger than their respective values.

Table 4.4 Aptina camera calibration parameters resulting from a 12.5mm grid

Parameter Name	Value	Standard Error (1σ)
Overall Pixel Error	(0, 0) pixels	+/- (0.259, 0.1994) pixels
Focal Length	(2745.88, 2747.75) pixels	+/- (1.212, 1.219) pixels
Principal Point	(1351.9, 975.691) pixels	+/- (0.4928, 0.4857) pixels
Skew	0	+/- 0
Radial Coefficients	(-0.1755, 0.1983, 0)	+/- (0.0005591, 0.001984, 0)
Tangential Coefficients	(0.0002264, 0)	+/- (0.0000379, 0)

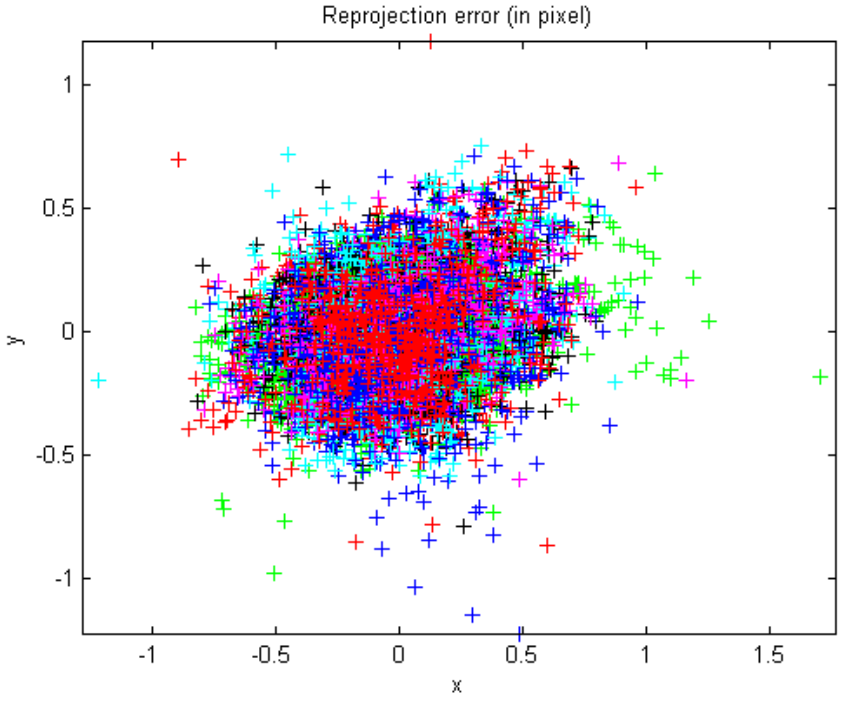

Figure 4.16 Aptina 12.5mm checkerboard reprojection error

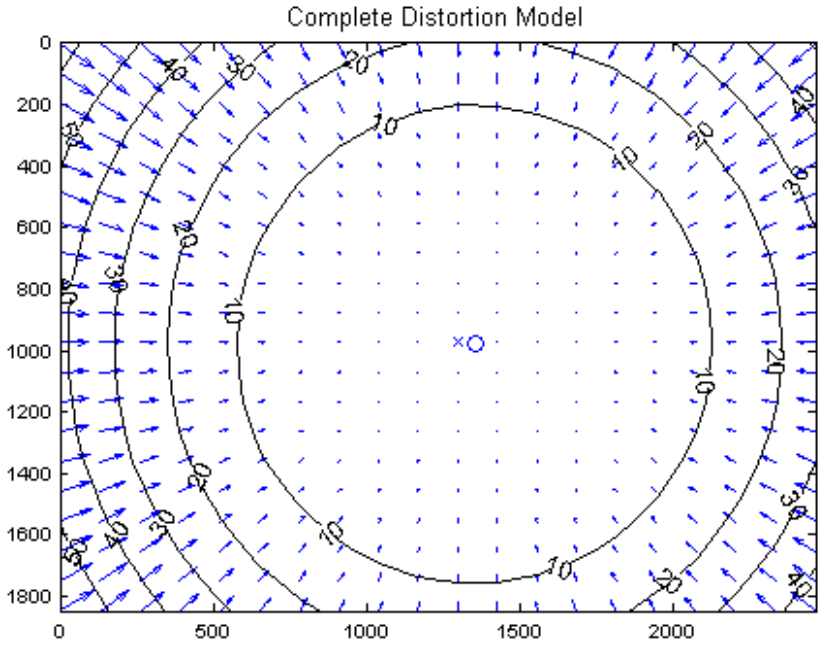

Figure 4.17 Aptina 12.5mm checkerboard overall distortion effects showing displacement in pixels

Figure 4.16 and Figure 4.17 show the reprojection error and overall lens distortion effects for the 12.5mm grid calibration, respectively. The reprojection error consists of a uniform statistical cluster of points with few outliers. This type of point cluster represents a proper calibration and shows uniformity between all input calibration images. The lens distortion effects consist of a uniform radial component which grows increasingly from the lens center. It is also seen that the principal point is slightly off-center yielding more distortion on the left half of the image.

4. Aptina 6mm Grid Calibration Results

Table 4.5 represents the final camera calibration results for the Aptina camera using a 6mm grid. As shown the overall pixel error lies within 0.3826 pixels of resolution which for the Aptina is equivalent to 0.4362 arc-minutes of accuracy. It is also shown that the distortion of this calibration is primarily due to radial effects as the tangential components are small. It can also be seen that the last radial coefficient component was manually set to zero since its error was larger than its value.

Table 4.5 Aptina camera calibration parameters resulting from a 6mm grid

Parameter Name	Value	Standard Error (1σ)
Overall Pixel Error	(0, 0) pixels	+/- (0.3826, 0.2871) pixels
Focal Length	(2753.82, 2756.29) pixels	+/- (0.6626, 0.6408) pixels
Principal Point	(1310.99, 962.647) pixels	+/- (0.5312, 0.6024) pixels
Skew	0	+/- 0
Radial Coefficients	(-0.1711, 0.1835, 0)	+/- (0.0005875, 0.002797, 0)
Tangential Coefficients	(-0.000183, -0.0006036)	+/- (0.00003766, 0.00004156)

Figure 4.18 and Figure 4.19 show the reprojection error and overall lens distortion effects for the 6mm grid calibration, respectively. The reprojection error consists of a uniform statistical cluster of points with few outliers. Many of the images in this calibration consistently yielded cross-axial error in three directions forming triangular distribution patterns.

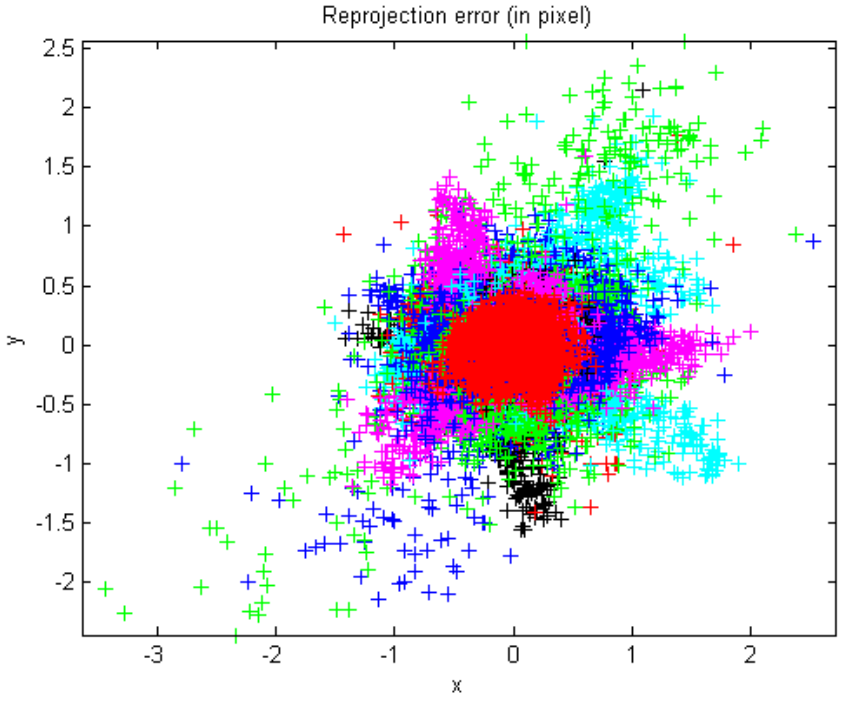

Figure 4.18 Aptina 6mm checkerboard reprojection error

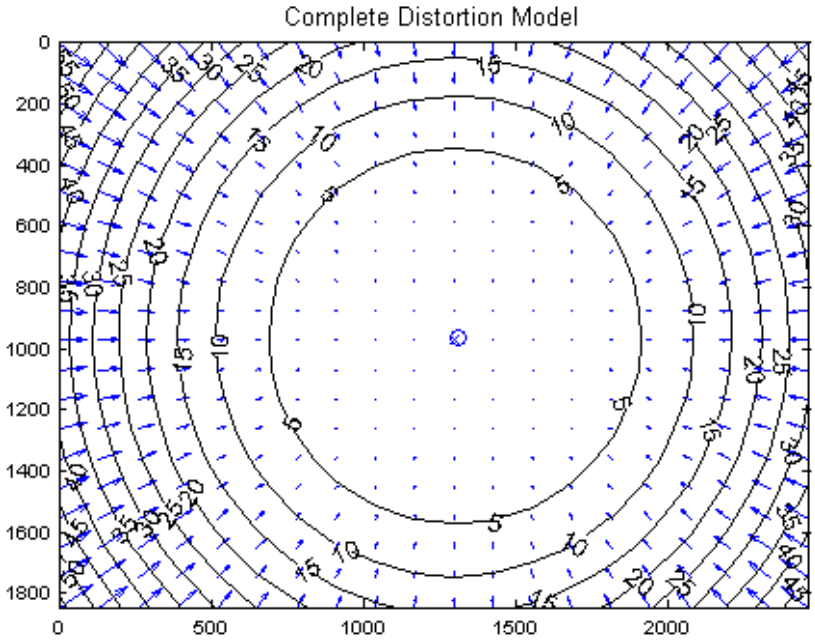

Figure 4.19 Aptina 6mm checkerboard overall distortion effects showing displacement in pixels

Error patterns of this type are important and should not be thrown out as they are introduced by the complexity of the distortion pattern unlike the banded cross-axial patterns which are due to improper grid corner extraction which can occur due to out of focus grids. The lens distortion effects consist of a uniform radial component which grows increasingly from the lens center. It is also seen that the principal point is slightly off-center yielding more distortion on the left half of the image.

V. Summary

With the calibrated camera parameters and the 6^{th} order distortion camera model the calculations of three-dimensional camera vectors from image sensor spot coordinates can be performed. These vectors will also have estimates of their error from the camera calibration results which can be used to better assess the capabilities of the studied cameras.

Comparing the calibration results for each camera yields a surprising outcome. For the Droid X2 camera the 6mm grid gave higher pixel accuracy whereas for the Aptina camera the 12.5mm grid was better. Looking into this further by manually analyzing the calibration images showed accuracy correlating with image focus and resolution, which affects the ability to properly detect the grid corners, instead of correlating with grid size. The smaller the grid size the fewer contrasting pixels there are to interpolate between to find the grid corner and therefore the accuracy of the extracted corner is lower. Therefore, because the Droid X2 and Aptina cameras have pixel resolutions of 0.91 and 1.14 arc-minutes, respectively the Droid X2 has a 19% increase in pixel contrast data available at each grid corner assuming that the camera contrast capabilities are similar (which is a good assumption given that the checkerboard corners represent areas of extreme contrast). It is also shown that between the two calibrations performed for each imager the standard errors of the focal lengths and principal points do not agree when extended out to 99% confidence intervals. In general this would yield improperly converged calibration data however the standard errors shown are those calculated by the convergence processes of the calibration method and therefore are based on calibration data and are only as accurate as the extracted grid corner locations which are again dependent on resolving capabilities. Given the knowledge of contrast differences and the fact that the focus is variable across all images helps explain the differences seen in the dataset accuracies.

Therefore, for image processing the datasets of higher accuracy will be used; the Droid X2 will use its 6mm grid dataset and the Aptina will use its 12.5mm grid dataset.

It is also important to note that the Aptina camera has a removable lens, which if moved will require the camera to be recalibrated as both the 12.5mm and 6mm grid calibration results show slightly off-center distortion. The Droid X2 does not have this limitation as its lens system is not manually adjustable.

CHAPTER 5

ATMOSPHERIC PROCESSING

I. Introduction

The purpose of atmospheric processing is to correct three-dimensional camera vectors for atmospheric perturbation effects resulting in corrected three-dimensional camera vectors. Star cameras orbiting the earth on a satellite do not require atmospheric processing as they are outside the atmosphere whereas star cameras on the surface of the earth are inside the atmosphere and therefore require atmospheric processing. This chapter describes the various techniques of correcting data which has been collected inside the earth's atmosphere as well as the methods used for correcting the acquired starlight spot vectors.

II. Background

A. Atmospheric Refraction

Just as a light ray bends as it passes through a transparent material, such as a lens, at an angle the same ray bends as it passes through the Earth's atmosphere. Earth's atmosphere acts as a giant lens where the angle at which a ray of light passes through the atmosphere determines the amount it will be bent, otherwise known as refraction. This factor is dependent on conditions such as humidity, temperature, pressure and relative atmospheric movement. Visually, atmospheric refraction causes an object to appear higher in the sky than it actually is.

For imaging purposes atmospheric refraction is more of a concern on large FOV cameras (FOV greater than 15 degrees) because the refraction angle will have a significant difference from one side of the image to the other. Since most smartphone based optics have large FOVs this effect must be taken into account when doing ground based testing.

Uniform atmospheric refraction can be accounted for in various ways. Bennett developed an empirical formula accurate to within 4.2 arc-seconds based off from Garfinkel's work in the late 1960's [42]. Equation (5.1) gives the refraction angle (ζ_{alt}) in arc-minutes given the angular altitude (h_a) in degrees, the surface temperature (T) in degrees Celsius and the surface pressure (P) in kilopascals.

$$\zeta_{alt} = \cot\left(h_a + \frac{7.31}{h_a + 4.4}\right)\left(\frac{P}{101}\right)\left(\frac{283}{273 + T}\right) \tag{5.1}$$

Bennett's method of calculating the refraction angle is a reliable approach as long as the atmospheric parameters are known (i.e. temperature and pressure) and accuracy greater than 4.2 arc-seconds is not desired. For cases where higher accuracy is required the more robust method created by Auer et al. is more appropriate [43]. Auer's method uses a more accurate representation of where on the Earth the viewing is taking place and also is developed as a piecewise integral divided across the various atmospheric layers (i.e. troposphere, stratosphere). His model, though more numerically complex, incorporates the fewest number of assumptions providing a more accurate representation of the refractive effects.

Also a concern when calculating atmospheric refraction effects is the reliability of the temperature and pressure data. Refraction increases and decreases by approximately 1% for every 0.9 kPa increase or decrease in pressure respectively. Likewise, refraction increases and decreases by approximately 1% for every 3 °C increase or decrease in temperature respectively [44].

B. Atmospheric Seeing / Scintillation

Atmospheric seeing, an adverse effect of high altitude turbulence, can largely effect what is seen through a camera viewing the night sky. The high altitude turbulence caused by a mixing of warm and cool air causes a visual twinkling of stars in the sky which not only affects the apparent magnitude of a star but also its apparent position. Seeing causes a star to appear to both move slightly and vary in brightness. This can have a huge effect on star identification from the ground as both the main characteristics of the star vary. Because seeing is caused by turbulence it can also be looked at as being weather dependent. A Canadian meteorological group has developed well known models [45] which aim at predicting both seeing and other weather based effects for astronomers. Many other models including those of Coulman [46] and Buscher [47] which get into advanced topics such as adaptive optics are very well performing as well. Overall, in order to account for seeing an understanding of the night in which the image is being taken in has to known.

III. Methodology

This section will describe the methods and steps used to correct the three-dimensional spot camera vectors due to the atmosphere. The chosen atmospheric model implemented for this task will be described and a discussion will follow as to why and how this model was selected. Figure 5.1 shows the stages required to arrive at the required corrected camera vectors.

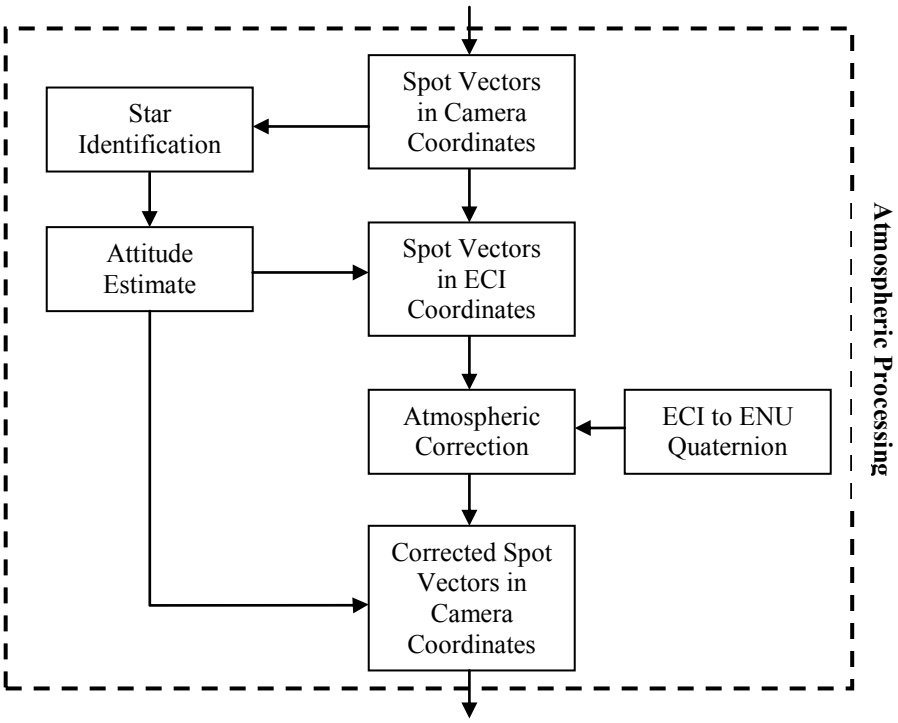

Figure 5.1 Flow diagram of the atmospheric processing stages

A. Obtain Preliminary Attitude

The first stage upon receiving spot camera vectors is to obtain a preliminary attitude solution from them before any atmospheric processing is performed. This is required for two reasons: (1) the atmospheric correction stage requires vectors in the earth-centered-inertial (ECI) coordinate system which can only be determined by a known attitude and (2) after the atmospheric correction stage is complete the vectors must be transformed back into camera coordinates. This preliminary attitude solution implements the methods discussed in detail in Chapter 7 where instead of passing in atmospheric corrected vectors, pure uncorrected

camera vectors are used. The attitude quaternion ($^{ECI}_{C}\hat{\bar{q}}$) is defined as the rotation from camera-to-ECI coordinates.

B. Define the ECI-to-ENU Rotation

The ECI-to-ENU (east-north-up) rotation is dependant only on the latitude and longitude of the taken images and the date and time at which they were taken. The date and time are used to calculate the Julian date and Greenwich sidereal time which is then added to the east longitude of the site to define the local sidereal time [48]. From the local sidereal time and the site latitude a rotation matrix is defined as

$$^{ENU}_{ECI}R = \begin{bmatrix} -\sin(\theta) & \cos(\theta) & 0 \\ -\sin(\phi)\cos(\theta) & -\sin(\phi)\sin(\theta) & \cos(\phi) \\ \cos(\phi)\cos(\theta) & \cos(\phi)\sin(\theta) & \sin(\phi) \end{bmatrix} \tag{5.2}$$

where ϕ is the site latitude and θ is the sidereal time. For implementation this rotation matrix is converted into a quaternion as of Appendix A and defined as $^{ENU}_{ECI}\bar{q}$.

C. Apply Atmospheric Correction

The atmospheric correction applied to camera spot vectors implements the atmospheric refraction equation Eq. (5.1). This equation requires knowledge of the altitude angle (h_a) of the camera vector as well as the temperature (T) and pressure (P) of the site when the image was taken.

1. Define the Altitude Angle

Using the preliminary attitude solution ($^{ECI}_{C}\hat{\bar{q}}$) and the $^{ENU}_{ECI}\bar{q}$ quaternion defined in the previous step the camera spot vectors are transformed into ENU coordinates using a series of quaternion rotations as defined in Appendix A. The altitude angle is then defined as the angle between these ENU vectors and the ENU horizontal plane. Equation (5.3) gives a method of calculating the altitude angle value where $^{ENU}\bar{V}$ is an ENU spot unit vector and $^{ENU}\bar{Z}$ represents the ENU coordinate systems up vector.

$$h_a = 90 - \cos^{-1}\left(\frac{^{ENU}\bar{V} \cdot {}^{ENU}\bar{Z}}{\left\|^{ENU}\bar{V}\right\|}\right), \qquad ^{ENU}\bar{Z} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{5.3}$$

2. Define the Temperature and Pressure

The temperature and pressure required for the atmospheric refraction equation is not measured during acquisition. Instead weather almanac data is acquired from nearby weather stations which are recorded at resolutions of thirty minutes [49],[50]. The data is extracted from the online weather service, wunderground, using their data extraction API (application programming interface). Connections made to the wunderground service requesting temperature and pressure data are automated as part of the image processing tool discussed in Chapter 8. These requests acquire the data closest to the times and locations at which the images were taken. Image date stamps and the known site latitude and longitude are used as inputs to the service.

3. Calculate Corrected Vectors

Having all the required inputs the atmospheric refraction equation, Eq. (5.1), can now be implemented yielding refraction angles. The calculated refraction angles (ζ_{alt}) are then used to rotate the ENU vectors toward the ENU horizontal plane since refraction causes an object to appear higher in the sky than it actually is. The following series of equations details this process. Equation (5.4) defines the rotation axis ($\bar{V}_r$) and its unit vector ($\bar{\imath}_r$) which are used to rotate the ENU spot vectors ($^{ENU}\bar{V}$) from the ENU frame to the atmospheric corrected ENU frame.

$$\bar{V}_r = {}^{ENU}\bar{V} \times {}^{ENU}\bar{Z}, \qquad \bar{\imath}_r = \frac{\bar{V}_r}{\|\bar{V}_r\|} \tag{5.4}$$

Equation (5.5) performs the rotation of the ENU spot vectors ($^{ENU}\bar{V}$) using a form of eigen-axis rotation where $\bar{\imath}_r$ represents the eigen-axis defined in Eq. (5.4). The result is the atmospheric corrected vectors ($^{ENU_c}\bar{V}$) in ENU coordinates.

$${}^{ENU_c}\bar{V} = {}^{ENU}\bar{V} \cdot \cos(h_a) + \left(\bar{\imath}_r \times {}^{ENU}\bar{V}\right)\sin(h_a) + \bar{\imath}_r\left(\bar{\imath}_r \cdot {}^{ENU}\bar{V}\right)(1 - \cos(h_a)) \tag{5.5}$$

The final step in the process is to transform the atmospheric corrected vectors back into camera coordinates ($^{C}\bar{V}$). The preliminary attitude solution ($^{ECI}_{C}\hat{\bar{q}}$) and the ECI-to-ENU quaternion ($^{ENU}_{ECI}\bar{q}$) are multiplied (in the

quaternion sense) to form the camera-to-ENU quaternion ($^{ENU}_{C}\hat{q}$) whose inverse is used to perform the transformation (Appendix A).

IV. Summary

Using the refractive model defined in this section and following the procedures to properly implement this model it is possible to obtain camera vectors which have been corrected for atmospheric refraction. It is also important to understand that this methodology does not account for the random atmospheric seeing effects or intensity losses and therefore further errors due to the atmosphere are possible. However, these errors are believed to be much less than the errors which are considered.

The applied atmospheric correction method will provide accuracies to within 4.2 arc-seconds. The model follows an exponentially decaying pattern with increasing altitude angle. At a 5 degree apparent altitude there is an ~10 arc-minute refractive change in position whereas at a 40 degree apparent altitude there is an ~1.25 arc-minute refractive change in position.

CHAPTER 6

STAR IDENTIFICATION

I. Introduction

The purpose of star identification is to map measured camera-frame spot vectors to known star vectors. Star identification is the principle behind star cameras in that without identification an attitude cannot be calculated. This chapter discusses the various methods used for star identification and how they are implemented.

II. Background

There are numerous star identification algorithms which have been developed. All of these methods can suffer from the following two types of errors: (1) misidentification error; and (2) errors originating from the chosen star catalog. Since star identification is a large problem in and of its own the methods and associated errors analyzed are those discussed by Brätt [51].

The key to successful star identification is having more than two stars in a given image. Given an imager's FOV there is a limit to the number of possible stars which can be viewed. As seen in Figure 9.16 for imagers having a FOV smaller than 40 degrees and whom can only capture stars having a magnitude less than 3 there will be areas of the sky which do not have enough stars for attitude determination. A camera's limiting star magnitude is directly correlated to its FOV; this is a key part in determining the ability to identify stars.

III. Methodology

Though various star identification algorithms exist [51] a method which completely matches every starlight spot to its proper star is desired such that a full working knowledge of each camera can be ascertained. This is not achievable when using classic methods. If a spot is not identified but did originate from a star then it will be thought of as noise, making the camera appear noisier than it is in reality. To best model a camera every detail must be discernible as either star data or camera error.

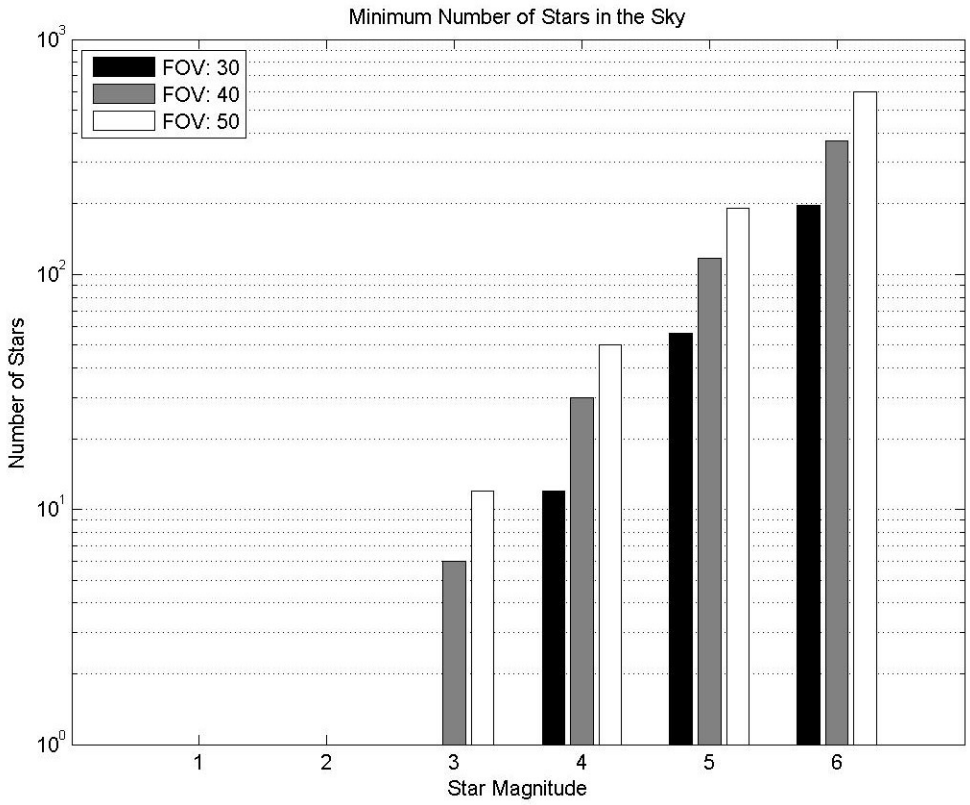

Figure 6.1 Minimum number of stars in the sky for a given imager FOV

A. Star Catalog

Star identification methods require a catalog of stars containing identifier codes and their position vectors for each star. The catalog vectors are what identification methods search through to find matches to spot vectors. The catalog utilized was the Hipparcos catalog whose data is listed in the 1991 epoch. This particular catalog was chosen due to the reasons discussed in [51]. The catalog vectors are represented in the ECI coordinate frame. These vectors were also date corrected using their known proper motions to eliminate as much error as possible. The dates used to correct the catalog were the dates the image datasets were taken. The remaining errors in the vectors are orders of magnitude smaller than the Aptina or Droid X2 cameras could measure and thereby will not influence results.

B. Star Overlay Method

To achieve complete star identification a manual technique was developed which allows an overlay of the stars in the sky to be placed on top of camera spots. The overlay can be manually moved and rotated by

a user until the stars align with the spots. Once aligned the overlapping stars and spots define the possible matches for the given image yielding the identification process complete.

This technique was developed as a MATLAB GUI (graphical user interface) and is integrated into the star processing tool discussed in Chapter 8. An example overlay is shown in Figure 6.2 where the green and blue stars represent true stars in the sky (the blue are constellation stars) and the white circles represent image spots. A full description of the overlay GUI is given in Chapter 8. This overlay method will be the basis of star identification as it guarantees all possible starlight spots receive a match.

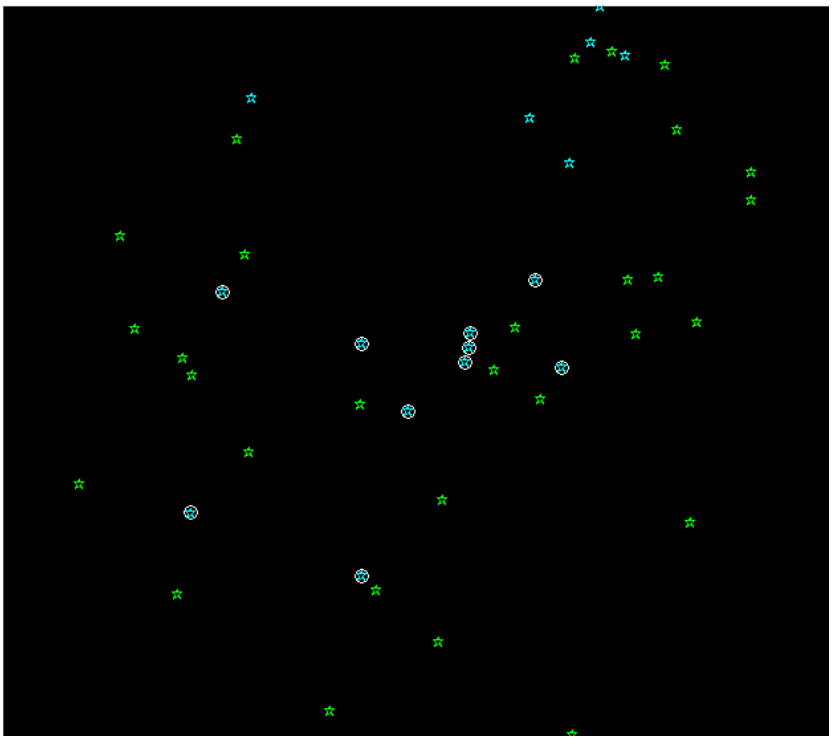

Figure 6.2 Example star identification overlay

IV. Summary

The star overlay GUI along with the Hipparcos catalog will allow stars to be matched to the spot vectors extracted from the images. This matching process provides the link between the camera frame and the ECI frame and leads to an attitude solution. The steps required to obtain this attitude solution are discussed in the following chapter.

CHAPTER 7

ATTITUDE DETERMINATION

I. Introduction

The purpose of attitude determination is to determine the best-fit rotation which aligns a series of measured body-frame vectors to their corresponding known reference-frame vectors. In the case of star cameras the body-frame is the star camera frame and the reference-frame is the ECI frame. This chapter first describes the possible approaches to solving this problem and then details the methods implemented for this star camera research.

II. Background

Five methods of calculating an attitude solution will be discussed. They include: (1) the TRIAD method, (2) Davenport's q-method, (3) the QUEST method, (4) mathematically averaging quaternions, and (5) attitude propagation methods.

A. TRIAD

The TRIAD algorithm is the simplest solution to the spacecraft attitude determination problem and was the first solution to be implemented. The algorithm obtains the direction cosine matrix relating two sets of vectors one taken from a reference frame ($\bar{r}_1, \bar{r}_2$) and the other measured in a body frame ($\bar{m}_1, \bar{m}_2$). The direction cosine matrix (${}^{ECI}_{C}R$) is defined by Eq. (7.1)

$$ {}^{ECI}_{C}R = [\bar{u}_{r1} \quad \bar{R} \quad \bar{u}_{r1} \times \bar{R}][\bar{u}_{m1} \quad \bar{M} \quad \bar{u}_{m1} \times \bar{M}]^T \tag{7.1} $$

where

$$ \bar{u}_{r1} = \frac{r_1}{\|\bar{r}_1\|}, \qquad \bar{u}_{m1} = \frac{\bar{m}_1}{\|\bar{m}_1\|} \tag{7.2} $$

$$\bar{R} = \frac{\bar{r}_1 \times \bar{r}_2}{\|r_1 \times \bar{r}_2\|}, \qquad \bar{M} = \frac{\bar{m}_1 \times \bar{m}_2}{\|\bar{m}_1 \times \bar{m}_2\|} \tag{7.3}$$

There are also covariance methods which can be used alongside this TRIAD method and allow error to be traced. For star cameras, pairs of stars and their corresponding camera vectors are passed through this algorithm which can yield more than one solution per image. From this stage there are a few possible directions to go. One possibility is to pass each rotation and its covariance into a Kalman filter and let it average out the values. Another possibility is to create an average rotation matrix by weighting them with respect to their covariances.

The major disadvantage of this algorithm is it is computationally inefficient and therefore is slower than other equivalent algorithms. It is also limited to two measurements at a time which for star cameras is not optimal because there are always more than two identifiable stars per image. For a more thorough view of the TRIAD algorithm's derivation see [52].

B. Davenport's q-Method

Davenport's q-method is a solution to Wahba's problem. Wahba's problem involves finding the orthonormal matrix A that minimizes the cost function

$$J(A) = \frac{1}{2}\sum_{i=1}^{n} w_i \|\bar{m}_i - A(\bar{r}_i)\|^2 \tag{7.4}$$

where w_i are non-negative weights, $\bar{m}_i$ are the unit measurement vectors, $\bar{r}_i$ are the unit reference vectors and n is the total number of measurement/reference vector pairs. The loss function can be rewritten as

$$J(A) = \lambda_0 - trace(AB^T) \tag{7.5}$$

where

$$\lambda_0 = \sum_{i=1}^{n} w_i \tag{7.6}$$

and

$$B = \sum_{i=1}^{n} w_i \bar{m}_i \bar{r}_i^T \tag{7.7}$$

Defined in this way it is seen that $J(A)$ is minimized when the $trace(AB^T)$ is maximized. Davenport parameterized the attitude matrix (defined in Appendix A) by a unit quaternion yielding

$$trace(AB^T) = \bar{q}^T K \bar{q} \tag{7.8}$$

where K is a symmetric traceless matrix defined by

$$K \equiv \begin{bmatrix} B + B^T - trace(B) I_{3\times 3} & \sum_{i=1}^{n} w_i \bar{m}_i \times \bar{r}_i \\ \left(\sum_{i=1}^{n} w_i \bar{m}_i \times \bar{r}_i\right)^T & trace(B) \end{bmatrix} = -\sum_{i=1}^{n} w_i \Omega(\bar{m}_i) \Gamma(\bar{r}_i) \tag{7.9}$$

Using the quaternion identities defined in Eq. (A.12) and Eq. (A.14) the right-hand-side of Eq. (7.9) can be derived. To optimize the attitude solution a quaternion must be found which maximizes the right-hand-side of Eq. (7.8) while it remains a unit quaternion ($\|\bar{q}\| = 1$). This optimal quaternion is equal to the normalized eigenvector of K having the largest eigenvalue and follows the solution of

$$K \bar{q}_{opt} = \lambda_{\max} \bar{q}_{opt} \tag{7.10}$$

The quaternion resulting from this eigenvalue problem of Eq. (7.10) represents the best fit quaternion for the given set of measurement vectors [53]–[56].

C. QUEST

The QUEST algorithm is an extension of Davenport's findings with the exception that it must use the traceless form of the K matrix. Following Davenport's q-method solution, the optimized loss function can be written as

$$J(A_{opt}) = \lambda_0 - \lambda_{max} \tag{7.11}$$

From this it was observed that $\lambda_{\max}$ can be found by a Newton-Raphson iteration scheme initialized at λ_0 since $\lambda_{\max}$ is very close to λ_0 when the optimized loss function is small. Normally, only a single iteration is necessary and is therefore more computationally efficient than the Davenport q-Method. This efficiency comes at the cost of being less robust; however, this method had proven itself reliable in flight [53]–[56].

D. Averaging Attitude Quaternions

Finding an average of given attitude quaternions is useful in that an optimal solution can be determined from numerous individual sensor attitude solutions. In another case, various fixed attitude estimates from a single sensor may require averaging to determine the quaternion which best represents all the data. This problem again uses the Wahba problem and is solved by a simplified version of Davenport's q-method [57]. The solution is

$$\tilde{\bar{q}} = \max(\bar{q}^T M \bar{q}) \tag{7.12}$$

where $\tilde{\bar{q}}$ is the average quaternion and M is the 4 x 4 matrix

$$M \triangleq \sum_{i=1}^{n} w_i \bar{q}_i \bar{q}_i^T \tag{7.13}$$

The solution to this maximization problem is the eigenvector of M which corresponds to the maximum eigenvalue.

E. Attitude Propagation

Attitude propagation is a technique used to estimate a future attitude based on both a known attitude and the rotation rates. The rotation rates can originate either from rate sensors mounted on the rotating body or if the body is rotating about an inertially fixed axis (such as the case for bodies on the surface of the earth). Equation (7.14) defines the propagation equation for rotations about an inertially fixed axis.

$$\bar{q}(t) = \bar{q}(t_0)e^{\int_{t_0}^{t}\left(\frac{1}{2}\Omega\right)\frac{d}{dt}} \tag{7.14}$$

where Ω is the quaternion form of the angular rate defined in Appendix A. This is the simplest form of attitude propagation as most other methods included additional terms to correct for conic motion [58].

III. Methodology

This section will define the methods and steps used to solve for both the individual camera image attitude solutions and their corresponding attitude errors. Figure 7.1 shows the stages necessary to arrive at attitude solutions for individual star camera images.

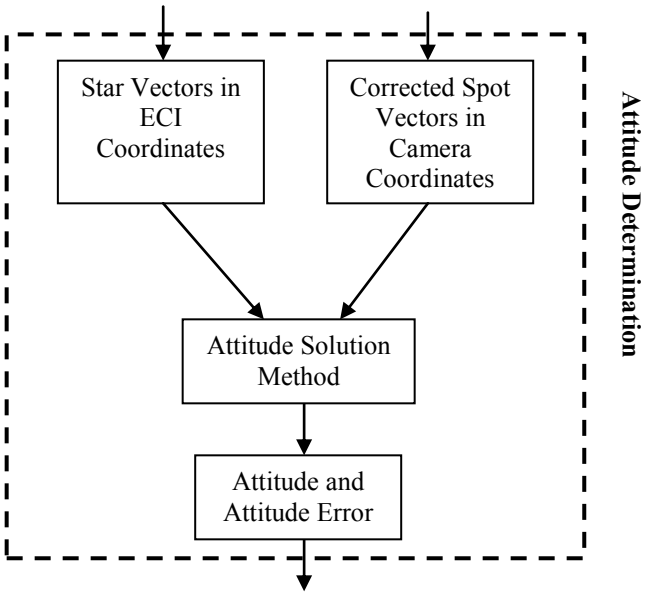

Figure 7.1 Flow diagram of the attitude determination stages

A. Attitude Determination

The attitude determination method chosen to process starlight spot camera vectors is Davenport's q-method. Even though the QUEST algorithm is more computationally efficient the q-method is more robust and yields solutions even if the data has excessive error. Also, since all data is post-processed computational efficiency is not required.

For the attitude solutions the corrected spot vectors in camera coordinates along with their respective star vectors in ECI coordinates will be passed to Eq. (7.10) and then solved using an eigensolver. MATLAB and its built-in eigensolver will be used as the implementation platform.

B. Attitude Error

Individual attitude solutions at the image level still contain error which cannot be totally defined by their covariances. Equation (7.15) defines the error quaternion ($\bar{q}_{error}$) as the truth quaternion ($\bar{q}_T$) multiplied (in the quaternion sense) by the estimated quaternion ($\hat{\bar{q}}$). Equation (7.16) defines the quaternion error (q_e) as a scalar angle calculated from the scalar component (q_s) of the error quaternion and is used to represent a camera's total error.

$$\hat{\bar{q}} = \bar{q}_T \otimes \bar{q}_{error} \rightarrow \bar{q}_{error} = \bar{q}_T \otimes \hat{\bar{q}} \tag{7.15}$$

$$q_e = 2\cos^{-1}(q_s) \tag{7.16}$$

The truth quaternion used in this formulation is both unavailable and unknown; however, a truth quaternion having a minimum error can be calculated given multiple estimates. The accuracy of this truth quaternion grows with the number of estimates. Implementation of this method only works on image datasets which have a fixed ENU orientation (fixed mounting position). Equations (7.17)-(7.19) define the governing equations for implementing this error technique. The constant camera-to-ENU mounting rotation (${}^{ENU}_{C}\hat{\bar{q}}$) is estimated for each image by Eq. (7.17) where ${}^{ENU}_{ECI}\bar{q}$ is defined using the method described in Chapter 5.

$${}^{ENU}_{C}\hat{\bar{q}} = {}^{ENU}_{ECI}\bar{q} \otimes {}^{ECI}_{C}\hat{\bar{q}} \tag{7.17}$$

Equation (7.18) defines an average camera-to-ENU quaternion (${}^{ENU}_{C}\tilde{\bar{q}}$) where the $qAvg$ operator represents the average quaternion of the set of quaternions and n is the number of images in the dataset. The technique used to average quaternions is the technique described in the previous section.

$${}^{ENU}_{C}\tilde{\bar{q}} = qAvg({}^{ENU}_{ECI}\bar{q}_i \otimes {}^{ECI}_{C}\hat{\bar{q}}_i)|_{i=1}^{n} \tag{7.18}$$

By quaternion multiplication a truth camera-to-ECI quaternion (${}^{ECI}_{C}\bar{q}_{T}$) is defined and shown in Eq. (7.19). The error quaternion for the camera-to-ECI rotation can now be defined by Eq. (7.15).

$$ {}^{ECI}_{C}\bar{q}_{T} = {}^{ENU}_{ECI}\bar{q}^{-1} \otimes {}^{ENU}_{C}\bar{q}_{avg} \tag{7.19} $$

This error method is coded in the MATLAB programming language and is used to define the overall repeatable error of each camera through analysis of the various image datasets.

IV. Summary

Using Davenport's q-Method to solve for individual image attitude quaternions allows for the fulfillment of the basic underlying principle of a star camera: obtain the camera's orientation. By calculating the error of each attitude solution the overall repeatable characteristics of each camera are definable. Furthermore, the error will allow in-depth studies of the processes involved in starlight passing through to the cameras.

CHAPTER 8

STAR IMAGE PROCESSING TOOL

I. Introduction

Thus far the individual stages of the star camera process have been discussed. This chapter describes the software tool developed to implement and combine each of these processing stages starting from an image and resulting in an attitude solution. This tool allows images to be processed quickly while also allowing quick iterative image investigations to be performed.

II. Methodology

The software tool was developed to incorporate all the methodology steps of each of the processing stages shown in Figure 1.3. These include: (1) image sensor processing, (2) camera lens processing, (3) atmospheric processing, and (4) star identification and attitude determination. The tool was required to have a robust ability to handle all types of star image processing such that a user can control all aspects of the processing. The individual abilities and how they link to the aforementioned processing stages is shown in the next section, design.

III. Design

Three main points define the tools design. They include: (1) code style, (2) GUI organization, and (3) code maintainability.

A. Code Style

The processing code is written as a series of MATLAB classes and packages which mesh together using an object-oriented-programming style to allow modular control over each sub-process while maintaining a clean code flow. Built on top of this processing code is a GUI which gives a user quick access to control variables. The GUI also provides access to post-processing and analysis functions to aid in result visualization. A script is also available and provides most of the same functionality as the GUI whereby instead of user selectable fields the user manually changes variables within the script. The GUI

was developed with the idea that a user unfamiliar with the MATLAB programming language would be able to both processes and visualize star images with ease.

B. GUI Organization

The GUI is divided into two main sections: (1) the user input pane and (2) the analysis pane. The user input pane is where all user input controls are housed. It contains a series of menus and options and is divided into two sub-sections: image processing and image post-processing. The analysis pane is where all the results generated during post-processing analysis are held. The results are organized in a tabular, hierarchical manner. This section is also where image processing user prompts are located such as the star identification overlay. An image of these sections is shown in Figure 8.1.

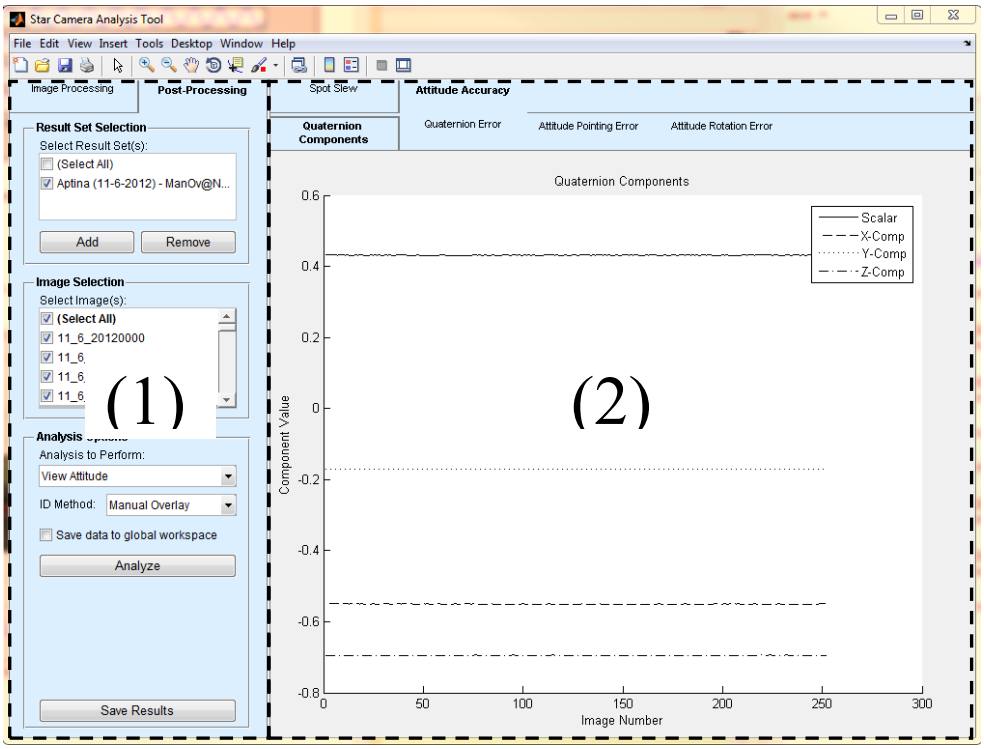

Figure 8.1 Star processing tool GUI organization

1. Image Processing

The image processing pane is divided into four sub-sections: (1) Image Parameters, (2) Star Camera Parameters, (3) Identification Parameters, and (4) Save Options. Each sub-section is accessible through either a tab GUI control or within a control frame as seen in Figure 8.2 and contain fields pertaining to their respective names.

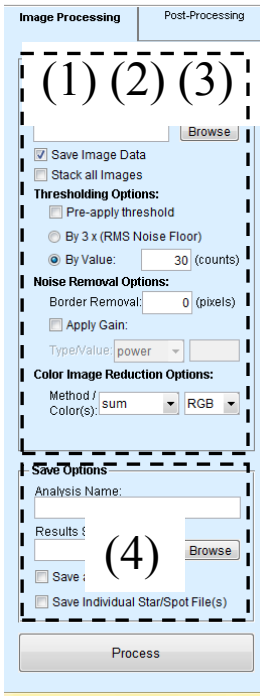

Figure 8.2 High-level sub-sections of the image processing pane

Figure 8.3 identifies all possible input controls available to the user within the image processing pane. The input controls are numbered. These numbers are used to identify each control in the descriptions given below. Each description will detail what the control is used for, if it is required and explain relations to other controls or processes which are specific to certain tasks.

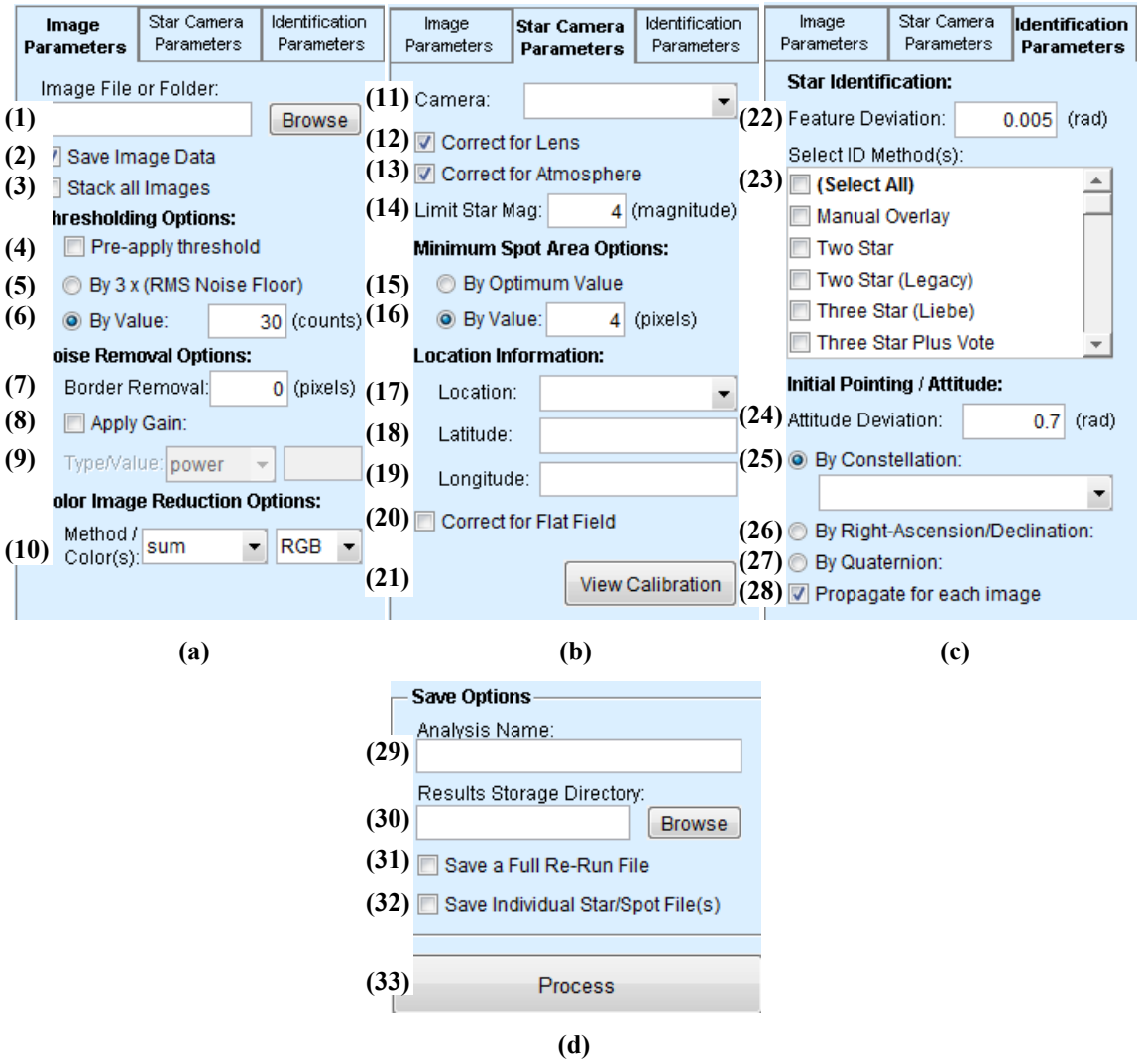

Figure 8.3 Image-Processing pane user controls: (a) Image Parameters (b) Star Camera Parameters (c) Identification Parameters (d) Save Options

1. Text-Field / Button (Image File or Folder): Holds the system file path to the image(s) to be processed. If the file path is a directory, every image file within that directory is processed in series as part of the same result data object. If the file path is an image file the single image is processed. This field is required. The browse button provides quick access to the file system of the computer alleviating the need to copy and paste full paths into the field directly.
2. Checkbox (Save Image Data): By default the tool will hold image array data in memory after processing is complete. It is then also saved to the result file. Unchecking this box will prevent

both of these processes form happening. Also, if a directory of images are processed containing ten or more images the tool will also remove image array data to save memory.

3. Checkbox (Stack all Images): When selected all images in the input directory will be processed as a single image by stacking (averaging) all the images together. This can aid in processing images with low SNR values.
4. Checkbox (Pre-apply threshold): This option is only applicable to color images and is ignored for all non-color images. When checked the threshold value applied to an image is applied to each color first before the image is convert to a grayscale image otherwise when unchecked grayscale conversion is performed first.
5. Radio-Button (By 3 x (RMS Noise Floor)): When selected the image threshold value is set to three times the RMS (root-mean-square) noise floor.
6. Radio-Button / Number-Field (By Value): When select the number entered is the value in intensity counts used to threshold the image(s). If this value equals zero no threshold is applied and the image will pass through the processing stages unchanged such that the original image is viewable for post-processing.
7. Number-Field (Border Removal): Defines the number of pixels to remove from all borders of the image(s). Value must be greater than or equal to zero.
8. Checkbox (Apply Gain): When checked the image gain options (9) become available.
9. Selection / Number-Field (Type/Value): The method and value of applying a gain to the image(s). The methods available are: power and percent. When power is selected the image pixels intensities are raised by the input value and then scaled back down to the range of the images original bit-depth. When percent is selected the value represents the percent of the maximum intensity specified by the image bit-depth which is added to the original image. Any resulting values which are greater than the maximum possible intensity specified by the image bit-depth are set back to the maximum value (they are cropped).
10. Selection / Selection (Method/Color(s)): This option is only applicable to color images and is ignored for all non-color images. The method selection details the method used to combine the

selected color(s). Possible methods are: sum, average, weighted average and RGB_to_YCbCr. Each method acts on the color arrays defined by the color selection (i.e. RGB, RG, GB, RB, R, G, B). The sum and average methods sum and average the selected color arrays respectively to form single grayscale arrays. The sum method also rescales the intensities back down to the original image bit-depth per color. The weighted average method uses intensity weights for the RGB array combination which match the patterns of the human eye. The RGB_to_YCbCr convert the RGB colorspace into the intensity based YCbCr colorspace. The Y intensity array is then used as the grayscale image after proper scaling is performed.

11. Selection (Camera): Defines the camera and the calibration model to use to process the image(s).
12. Checkbox (Correct for Lens): When checked the geometric calibration model associated with the selected camera (11) is used to correct the image, otherwise, the basic distortionless pin-hole camera model is implemented using only the focal length and principal point of the selected camera.
13. Checkbox (Correct for Atmosphere): When selected the image is corrected for atmospheric effects.
14. Number-Field (Limit Star Mag.): Represents the lower star intensity limit. Star's of magnitudes greater than this value will not be included for star identification.
15. Radio-Button (By Optimum Value): When selected an optimal minimum spot area is determined and used for noise removal.
16. Radio-Button / Number-Field (By Value): When selected the value represents the minimum number of pixels an image spot must contain to be treated as a starlight spot.
17. Selection (Location): When a location is selected its latitude and longitude automatically populate their respective fields. This field is not required; it is instead provided for quick access.
18. Number-Field (Latitude): The latitude of the location the image(s) were taken. This field is required if atmospheric correction is enabled.
19. Number-Field (Longitude): The longitude of the location the image(s) were taken. This field is required if atmospheric correction is enabled.

20. Checkbox (Correct for Flat Field): When selected the image(s) are intensity corrected using the flat field data defined for the selected camera (11).
21. Button (View Calibration): When pressed a figure is shown which details the pixel error defined during the geometric calibration of the selected camera (11).
22. Number-Field (Feature Deviation): For automated star identification algorithms it represents the allowable angular difference between star identification catalog features and the corresponding measured patterns. For the Manual Overlay it represents the angular difference allowed for a match to be made between an overlay star and an image spot.
23. Multiple-Selection (Select ID Method(s)): Represents the available star identification methods. Each selected method will individually process the image data allowing for post-process comparisons to be made. If no methods are selected star identification will not be performed.
24. Number-Field (Attitude Deviation): Represents the allowable deviation from the given initial attitude. This field is required for the Manual Overlay and Two Star (Legacy) star identification methods and is optional for all others. This value is used to crop the star catalog and respective feature-lists radially from the defined initial attitude.
25. Radio-Button / Selection (By Constellation): The center of the selected constellation is used as the camera's initial attitude.
26. Radio-Button / Number-Field(s) (By Right-Ascension/Declination): The input right-ascension and declination in ECI coordinates is used as the camera's initial attitude.
27. Radio-Button / Number-Field(s) (By Quaternion): The input four-variable quaternion is used as the camera's initial attitude.
28. Checkbox (Propagate for each image): When selected the attitude solution calculated for each image is propagated (using the difference in image times) and used as the following image's initial attitude. When using the Manual Overlay to process a directory of images setting this to true saves time.

29. Text-Field (Analysis Name): A user given name to define the set of data. This name will appear when loaded into the post-processing section of the tool. If left blank a generic incrementing name whose prefix, “Analysis_” is applied.
30. Text-Field / Button (Results Storage Directory): A file system path to a directory in which to store the results from the image processing stage. The browse button provides quick access to the file system of the computer alleviating the need to copy and paste full paths into the field directly. If left blank the current active directory defined by the MATLAB pwd command is used.
31. Checkbox (Save a Full Re-Run File): When checked a full re-run file is created and stored as a mat file. This file can then be loaded back into the post-processing section of the tool at a later date.
32. Checkbox (Save Individual Star/Spot File(s)): When checked individual mat files for each processed image will be made. These files will contain an array of StarIdentification objects, one for each star identification method used. This is useful for quick access to identification data of individual images.
33. Button (Process): When pressed the image(s) will be processed using the parameters defined by all the user controls in the image processing pane. When processing completes the resulting data is automatically available for post-processing.

2. Post-Processing

The post-processing pane contains three sub-sections: (1) Result Set Selection, (2) Image Selection and (3) Analysis Options. Figure 9.16 identifies all possible input controls available to the user within the post processing pane. The input controls are numbered. These numbers are used to identify each control in the descriptions given below. Each description will detail what the control is used for, if it is required and explain relations to other controls or processes which are specific to certain tasks.

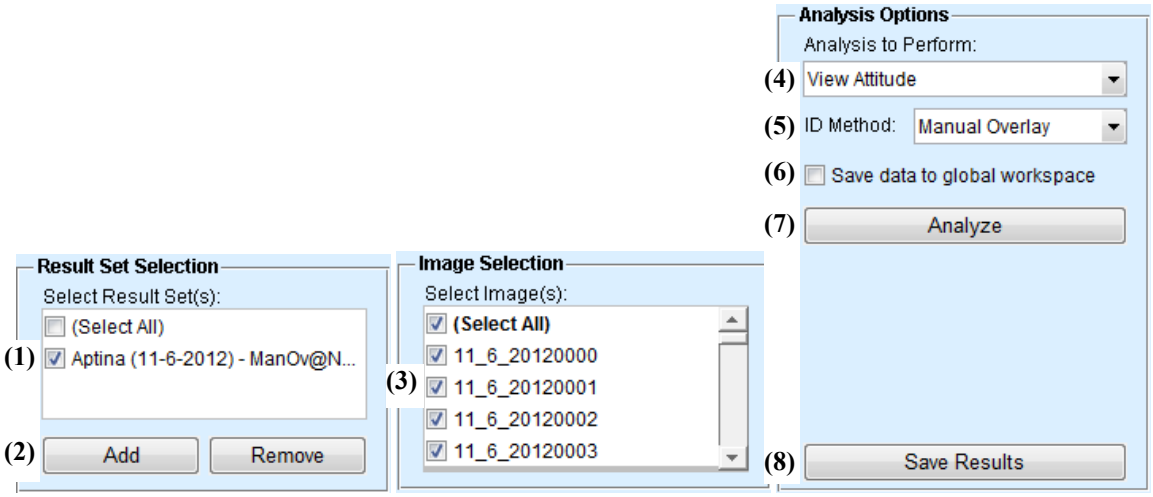

Figure 8.4 Post-Processing pane user controls: (a) Result Set Selection (b) Image Selection (c) Analysis Options

1. Multiple-Selection (Select Result Set(s)): Loaded result sets appear here. Every dataset available here is an option is held in memory. Those checked will be used in the selected analysis (4).
2. Button / Button (Add / Remove): The add button when pressed opens a file system browser which you can used to navigate and select a result set mat file to load and make available for selection. The remove button clears any selected datasets from memory freeing up system resources. This is useful when many test processes are run which do not need to be analyzed. Resources can be freed to continue running additional tests without having to close and re-open the tool.
3. Multiple-Selection (Select Image(s)): When a single result set is selected (1) all images within that set will appear here and can be selected for analysis. When more than one result set is select all images from all datasets will be included in the analysis.
4. Selection (Analysis to Perform): Provides a list of available analyses which can be performed on image(s). Depending on the analysis selected it may be required that the images had star identification performed.
5. Selection (ID Method): For analyses requiring star identification data the star identification data of the selected method will be used.

6. Checkbox (Save data to global workspace): An option for all analyses is to have data passed back to the global workspace which a user can use for unique tests not included in the tool. When checked the data defined for the selected analysis method (4) will be copied to the user's global MATLAB workspace.
7. Button (Analyze): When pressed the selected analysis (4) is performed.
8. Button (Save Results): When pressed all axes, plots and data available in the analysis pane is saved to a user defined directory. Each are saved as fig, jpeg and eps files.

C. Maintainability

To make the code both maintainable and flexible many of the options available in the GUI have easy modification points allowing for quick changes to be made. These modification points are all detailed within the "Dependencies" directory of the software package. After performing any of the following updates all running versions of the GUI must be restarted before the changes will become available.

1. Adding New Cameras

New cameras can be added by appending camera structures to the setCameras.m file located in the "Dependencies\Cameras\" directory. The fields required in the structure are: name, chipFormat, geometricCalibrationFile and intensityCalibrationFile. The name field is a user definable name where the current standard is to append the calibration grid size to the end of the name. The chipFormat field is a string defining the camera sensors format. The geometricCalibrationFile is the file name, including extension, to the calibration file (returned from the camera calibration toolbox). This calibration file must be stored in the "Dependencies\Cameras\Calibrations\Geometric\" directory. The intensityCalibrationFile is the file name, including extension, to the intensity calibration file. This calibration file must be stored in the "Dependencies\Cameras\Calibrations\Intensity\" directory.

When the setCameras.m file is edited it must be executed in MATLAB from the top-level software directory while selecting "Add to Path" at the prompt. This re-saves the camera structure to the cameras.mat file found in the same directory.

2. Adding New Locations

New site locations can be added by appending site structures to the setImageLocations.m file located in the "Dependencies\ImageSites\" directory. The fields required in the structure are: name, latitude and longitude allow of which pertain to the site(s) being added.

3. Adding New Star Identification Methods

New star identification methods can be added by appending method structures to the setStarID_Methods.m file located in the "Dependencies\StarID\" directory. The fields required in the structure are: name, methodName and featureType. The name is the visible name which will appear in the GUI but must also be unique. The methodName corresponds to the name of the StarIdentification class method name used to perform the identification and is the name of the new methods m-file without the extension. The featureType corresponds to the suffix of the StarCatalog class's feature extraction method where the prefix is "getFeatureList_".

The new method's m-file must be placed into the StarIdentification class's folder. One last edit must also be made to the StarIdentification class definition file StarIdentificaiton.m. The new methods static function definition must be added to this file. If a new feature type is also defined a function accepting a StarCatalog object must be created which returns the featureList. This function must be placed in the StarCatalog class's folder and its naming convention must follow the prefix and suffix notation described above.

4. Adding New Analyses

New analysis methods can be added by appending method structures to the setAnalysisFns.m file location in the "Dependencies\AnalysisFns\" directory. The fields required in the structure are: name, fnName, analysisName, usesStars, plotNames and data. The name field represents the name which appears in the GUI selection box. The fnName is the name of the new analysis function's m-file which needs to be placed in the "Dependencies\AnalysisFns\" directory. The analysisName field is the text displaced in the high-level analysis tab. If left blank only a single plot is allowed whose name will be used in its place. The usesStars field is a Boolean (true or false) value. When true the tool will pass indexes and method names to

the analysis function (the particulars of which are discussed below). The plotNames field is a cell array of plot names where one name is given to each plot axis made within the function. These names will be used as the sub-tab headers. The number of names give also defines the number of MATLAB axes passed to the function. The data field defines the type of data the function requires. The options are: image, sensor and stars. Using this technique prevents having to loop through the main data object extracting the desired data within each analysis function. Instead the data pertaining to the analysis is passed directly to it making the code much cleaner. When the data field is image the tool passes an array of CameraImage objects pertaining to the selected images. When the data field is sensor the tool passes an array of axes, an array of sensor objects (in the case of the GUI StarCamera objects), an array of image indexes corresponding to the selected images and an array of indexes (the length of which is equal to the number of selected sensors) defining the index of the selected star identification method for each sensor. When the data field is stars the tool passes an array of axes, an array of CameraImage objects pertaining to the selected images and the name of the star identification method to use for data extraction.

5. Adding New Controls

Adding new GUI controls has three steps which need to be followed. The first is to locate the code section where the new control will be placed, position the control and define the new control within the CameraAnalysisTool.m file. The second is to push the output of the new control to either the CameraImage object or the StarCamera object by adding a value extraction line of code to the processImage sub-function of the CameraAnalysisTool.m file (located at the top of the file). The third is to add a new class property to the class definition file of the class the variable is being passed to (either CameraImage.m or StarCamera.m). These three steps allow passing GUI control variables to the underlying code where it can then be used in other stages of processing.

IV. Usage and Limitations

- Directories can only contain images from a single camera. Directories containing mixed camera image will cause the code to crash (unless the images have the same specifications).

- The optimal minimum spot area option only works well for high SNR cameras. This option will not work for cameras containing large amounts of noise.
- The object oriented programming style in MATLAB is slower than traditional modular code. However, each new release of MATLAB is improving this deficiency making this less of an issue.
- The result set data files are MATLAB mat files containing large objects and arrays of objects. Reading and writing from these files can take time and is significantly slower than manipulating mat files containing normal variables.
- Result sets loaded into a computer having a different directory structure will not be able to reload image data as the paths saved are full system paths (not relative). Therefore, analyses requiring raw image data not stored within other data objects will not work.
- Propagation of image quaternions is only as accurate as the timestamp of the image. Images with lower timestamp resolution will have additional error added to the propagated quaternion.

V. Summary

The image processing tool described in the section is meant to be a time saving tool which both helps both process and analyze star data. The flexibility built into both the GUI and the underlying code structure allows newly required options to be easily added. The tool was mainly built due to the need of identifying all of the spots on an image, something star identification algorithms cannot achieve. It started out as just the star overlay identification method and progressed into the tool it is after realizing that the Droid X2 camera created other issues, mainly not being able to distinguish stars from noise. From this realization and the fact that many images would need to be processed in both a reliable and repeatable manor the decision to build the tool was made.

CHAPTER 9

STAR IMAGE ANALYSES

I. Introduction

This chapter focuses on analyzing the processed image data from the acquired images of each camera. The aim of the analysis is to determine the underlying limitations of each camera. The ability of each camera to capture starlight on images will be determined; if it is able then repeatability will also be determined. The determined attitude solutions will provide insight into the accuracies of each imager and allow for a breakdown of the error into the primary three components: error due to the sensor, error due to the lens and error due to the atmosphere.

II. Signal-to-Noise Ratio

The SNR value is important in determining the ability to extract starlight spots. The image noise is comprised of both sensor dark currents as defined in Chapter 3 and imaging environmental effects. Table 9.1 defines the RMS sensor noise of images taken with the camera's lens cap on. The three different values given for the Droid X2 are the red, green, and blue color channels defined individually. These values represent the base noise which is applicable to all images.

Table 9.1 RMS noise floor with the lens cap on

Camera	RMS Noise
Aptina	35.21
Droid X2	5.25, 1.51, 2.45

Table 9.2 provides individual values for the Aptina and Droid X2 camera SNR values. Again, the three values given for the Droid X2 are the corresponding SNR values associated with the red, green, and blue color arrays, respectively. The SNR value for the Aptina was calculated with a star having a magnitude equal to 2.07 whereas the SNR value of the Droid X2 was calculated with a star having a magnitude of 1.64. These values were each obtained from a single image.

Table 9.2 Signal-to-noise ratio of each camera

Camera	SNR value
Aptina	6.17
Droid X2	5.7, 10.53, 9.31

III. Image Processing

This section provides a table (Table 9.2) of result sets used for the analyses. Each result set represents processed data from the datasets described in Table 2.1 where the processing parameters are varied. These variations in processing parameters allow for the components of error to be discerned. Each result set is given a number (Result Set) which is used to identify it within the various analyses. The varied parameters include: the minimum star magnitude (Star Mag.), the noise threshold (Threshold), the minimum allowable spot size (Spot Size), the included colors (Colors), atmospheric correction whether it is on or off (Atmos.), and lens correction whether it is on or off (Lens). These parameters were direct inputs into the analysis tool.

Table 9.3 Image result set processing parameters

Result Set	Camera	Dataset	Star Mag.	Threshold	Spot Size	Colors	Atmos.	Lens
1	Aptina	10-5-2012	5	1024	6	N/A	Yes	Yes
2	Aptina	10-5-2012	5	1024	6	N/A	No	Yes
3	Aptina	10-5-2012	6	3 x RMS	4	N/A	Yes	Yes
4	Aptina	11-6-2012	5	1024	6	N/A	Yes	Yes
5	Aptina	11-6-2012	5	1024	6	N/A	Yes	No
6	Aptina	11-6-2012	6	3 x RMS	4	N/A	Yes	Yes
7	Droid X2	11-6-2012	2.8	18	4	G	Yes	Yes
8	Droid X2	3-30-2012	2.8	18	15	G	No	Yes

IV. Image Quality

The basic definition of image quality for star images is the ability to capture stars and distinguish them from the various noise sources. Though other quality aspects can be considered such as pixel saturation

levels and spot eccentricity, the main desire is to make sure starlight is definable on the image sensor. This section provides image and image processing histogram analyses and basic noise trends for each camera.

Figure 9.1 and Figure 9.2 show example image intensity histograms after thresholding from the Aptina and Droid X2 cameras, respectively. The two figures reveal both digital resolution and range differences between the two cameras. The Aptina has greater digital resolution due to its 12-bit imaging capabilities compared to the Droid X2's 8-bit. The Aptina also has a greater range compared to the Droid X2 which is shown by the difference in the intensity spreads of the two cameras. For the Aptina, after thresholding three-quarters of the intensity bins are noticeably active whereas for the Droid X2 only approximately one-ninth of the bins are noticeably active. This larger range aids in noise removal. Due to the green color channel of the Droid X2 having the highest SNR value it was chosen as the color channel to base all processing analyses on.

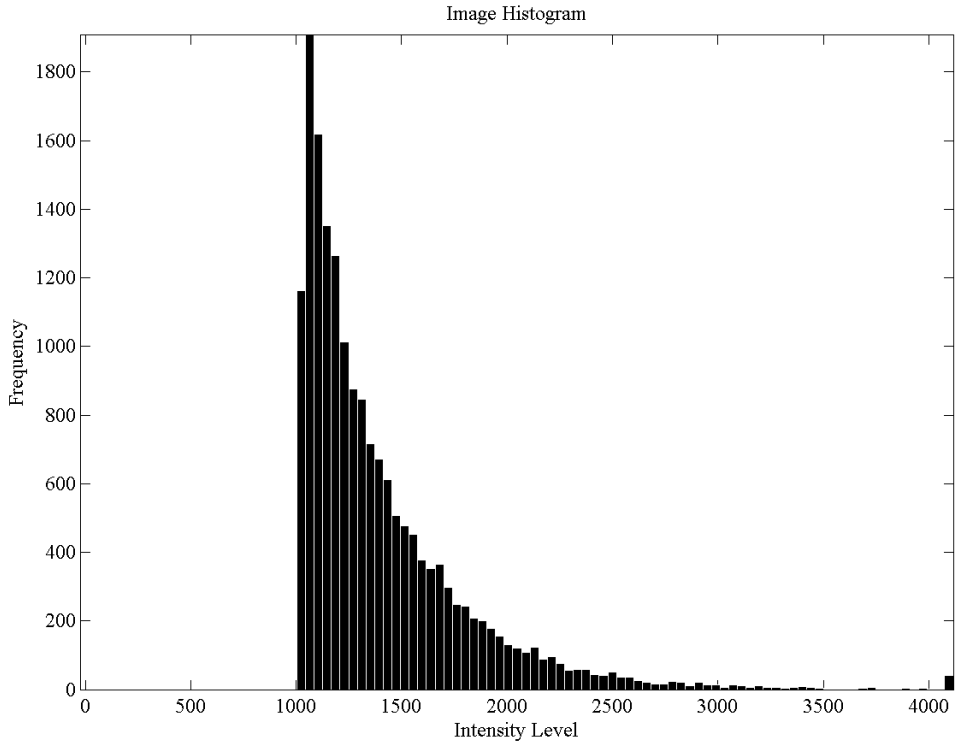

Figure 9.1 Aptina - image intensity histogram after thresholding (12-bit image) (result set 1)

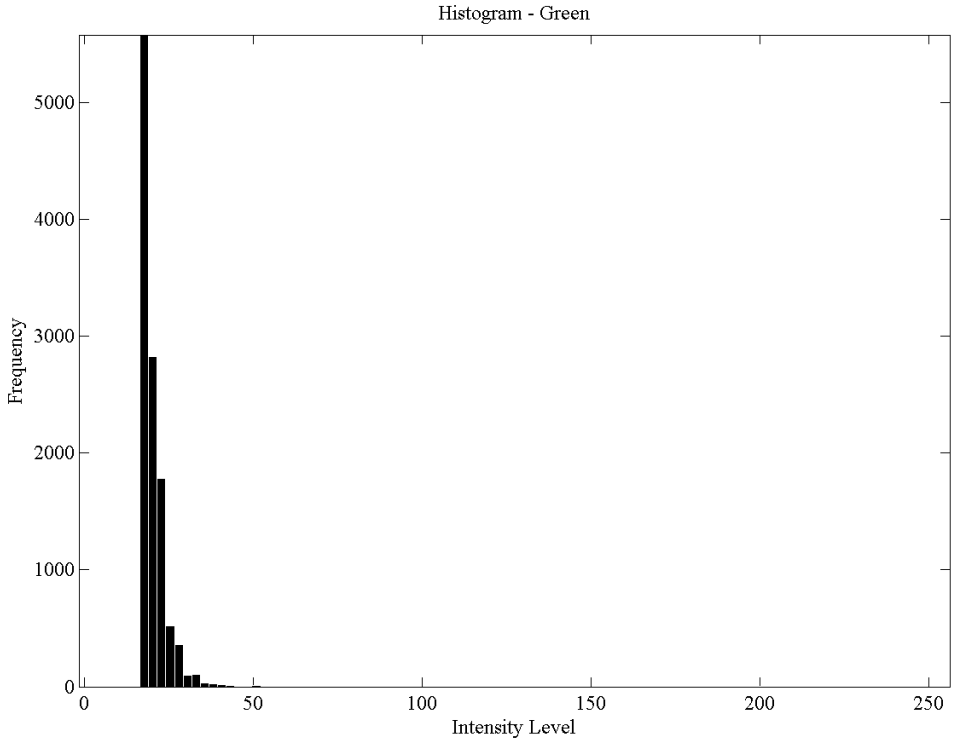

Figure 9.2 Droid X2 - image intensity histogram after thresholding (8-bit image) (result set 8)

After image sensor processing (which includes the thresholding and spot size reduction, see Table 9.3) the remaining spots were analyzed. Figure 9.3 and Figure 9.4 shows histograms of spot intensities (the sum of the pixel intensitites making up a spot), spot areas and spot flatness factors (a normalized variance of spot pixel intensities) for each camera. The frequencies shown are normalized based on the number of images and therefore represent estimates to the probability of an occurrence at each bin location.

The Aptina results yield excellent capabilities in noise reduction as at the settings used for this result set (set 1) the noise is unnoticeable. In the same regard it shows highly predictable trends in the respects of spot characteristics. A clear definable minimum spot area can be seen; though set at 4 pixels in this result set the area histogram reveals that settings outward of 20 or higher should be reliable as well. As for intensities, they are important for proper spot centroiding. The flatness factor is a measure of the shape of each spot where the percentage is physically the percentage of a spot's one standard deviation intensity level to the spot's maximum intensity. The shape of Aptina spots in regards to this flatness factor show a

near Gaussian trend with a mean around 20%. This could be used as a part of another spot extraction technique not focused on spot area and threshold values.

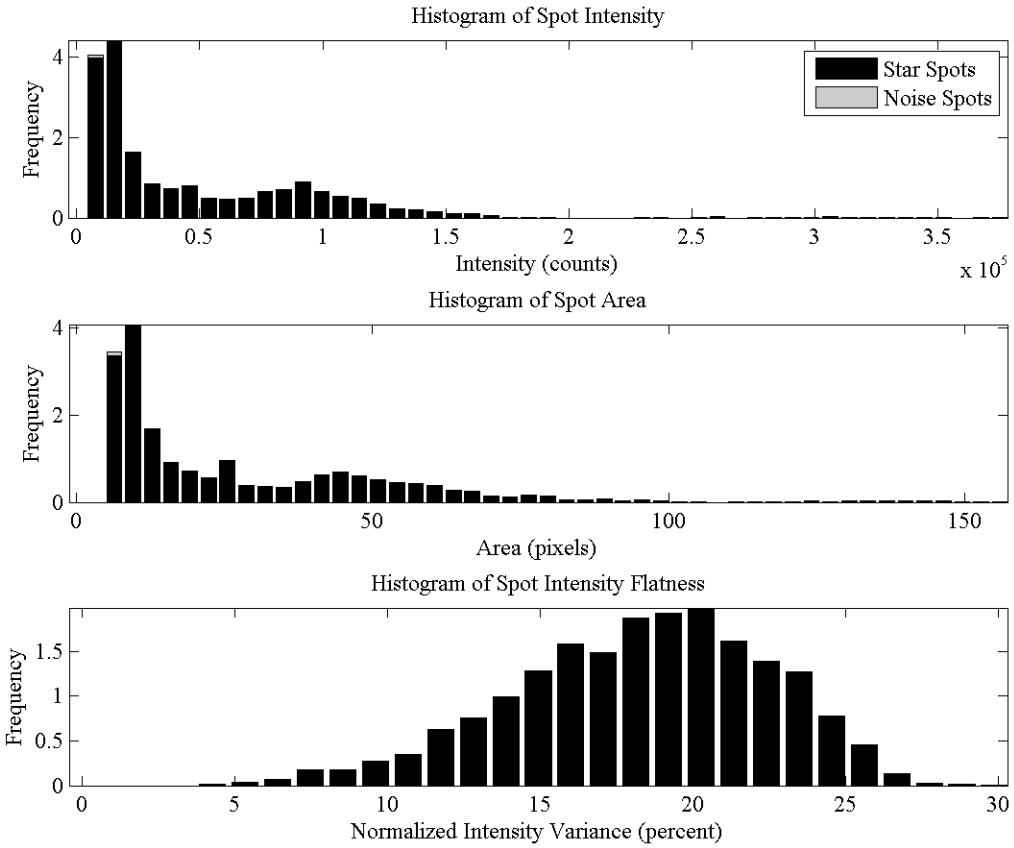

Figure 9.3 Aptina – single image histograms after thresholding of (top) spot intensity volume, (middle) spot area and (bottom) spot flatness factor (result set 1)

The Droid X2 results reveal noise dominance. No distinguishing patterns are followed only by starlight spots; noise spots follow most all the same patterns. Noise spot areas do tend to be heavily weighted below 5 pixels however additional processing of the images did not show any improvement in removing this high noise dominance. Spot flatness factors do not lead to a well behaved pattern and therefore also cannot be used to distinguish starlight spots from noise spots. Without the ability to separate a majority of the noise from the stars an attitude is not achievable.

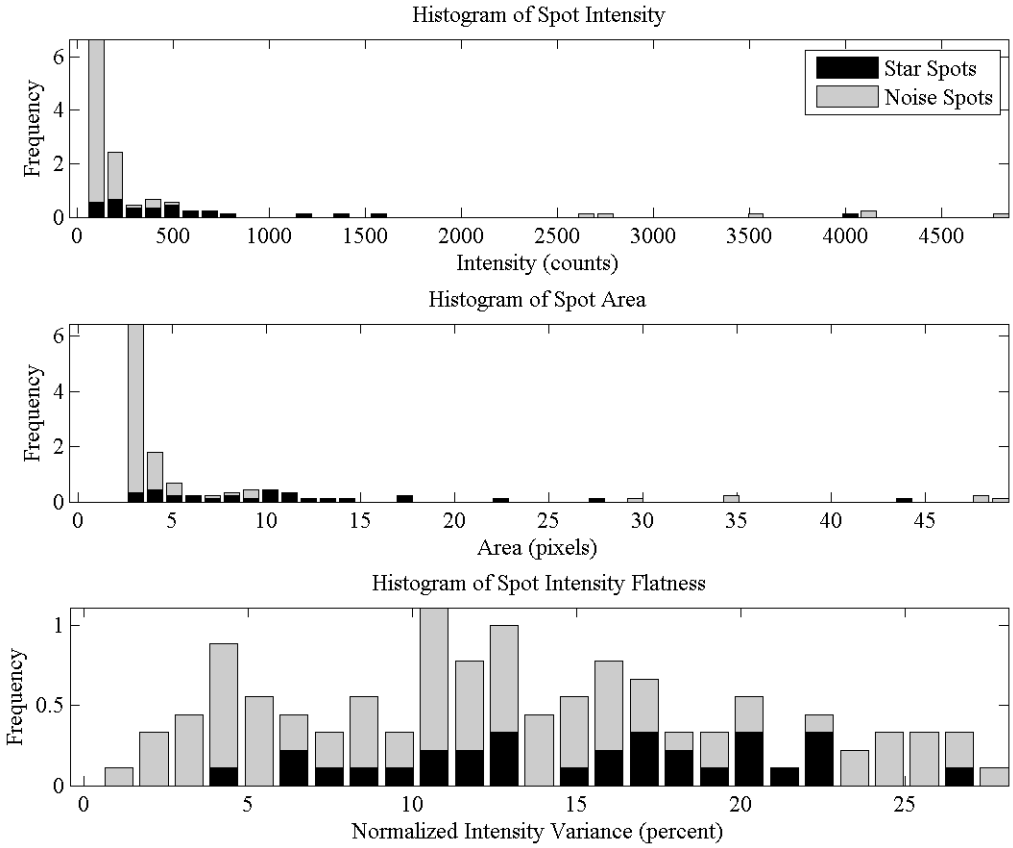

Figure 9.4 Droid X2 – single image histograms after thresholding of (top) spot intensity volume, (middle) spot area and (bottom) spot flatness factor (result set 8)

Table 9.2 show the number of starlight and noise spots on average per image for each camera. It is seen that the Aptina has a maximum of one noise spot in images where noise did exist and that over 75% of images had no noise. Also, though not distinguished in the table, is that all the noise spots that are defined in the table are stars of a higher magnitude than that used for the result set.

In order to obtain results on noise for the Droid X2 extensive iterative testing was done to arrive at the processing parameters used for the result sets. The SNR values alone did not provide a reliable means of defining the threshold and minimum spot area. The identified processing parameters where used with the overlay star identification method and allowed many images to be processed. The results again show Droid X2 as being completely noise dominated with over 90% of the spots per image being noise. This corresponds to over 80 noise spots per image or 1 to 3 starlight spots per image. It should also be noted that some images are complete noise not having a single starlight spot.

Table 9.4 Starlight and noise spot statistics

Camera	Average Number of Noise Spots per Image	Average Number of Starlight Spots per Image
Aptina	0.09	16.23
Droid X2	91.15	1.85

V. Spot Characteristics

The following section attempts to describe and classify the spots imaged by both cameras. Both noise and starlight spots are detailed and distinguishing features such as spot area and intensity are looked at further. Analyses are also shown as to the variability of a single starlight spot over time. These analyses are meant to show the variability of spot features which effect spot centroiding and in turn accuracy. It is also important to note that the figures showing spot pixel distributions have weighted intensity values to make the spot shape and profile more visible. This means the grey-scale values fluctuate image to image and therefore cannot be used to compare overall intensity.

A. Pixel Distributions of Spots

Figure 9.5 and Figure 9.6 show 20 random starlight spot images from the Aptina and the Droid X2 cameras respectively. These spots were extracted at random among the available starlight spots spread throughout the result sets.

Aptina starlight spots vary greatly in their areas ranging from 9 to over 100. For the most part, the spots' pixel distributions represent ovals of low eccentricity. Though the occasional small and linearly oriented spots appear they are linked to stars having magnitudes near the threshold border. Therefore after thresholding few pixels remain.

The Droid X2 images spots having areas as low as 2 to over 50. The distributions seen below have equal trends of oval shaped and linearly shaped starlight spots. Where the spots are oval in shape the intensities are well behaved (Gaussian in nature); however smaller spots and those that are linear tend to have a lower flatness factor and therefore there intensities are more uniform.

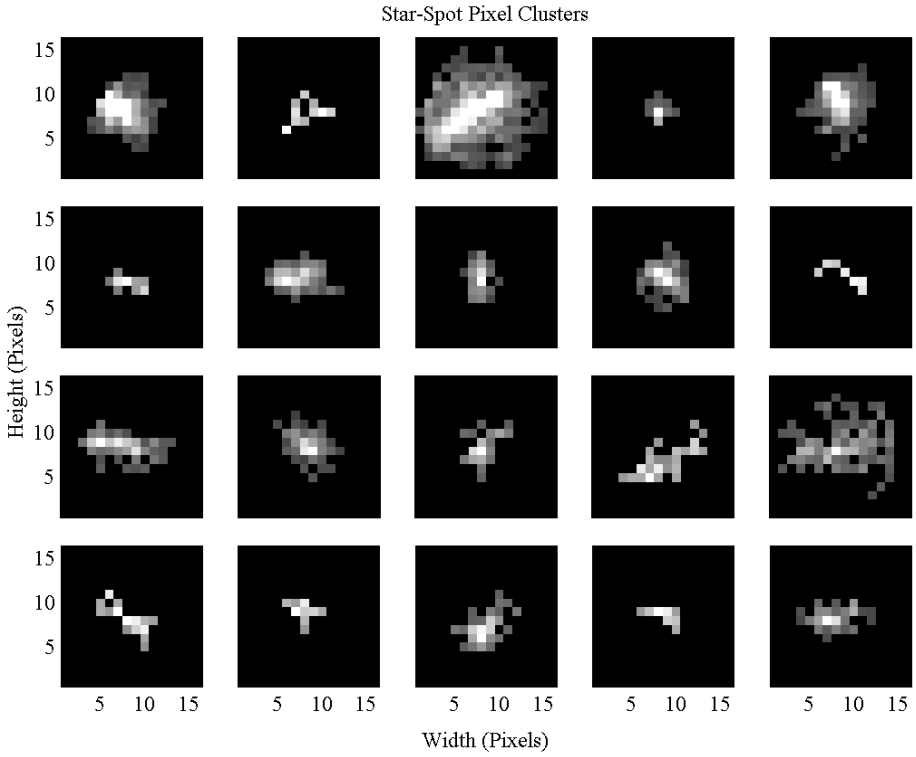

Figure 9.5 Aptina - random starlight spot pixel clusters (result set 1)

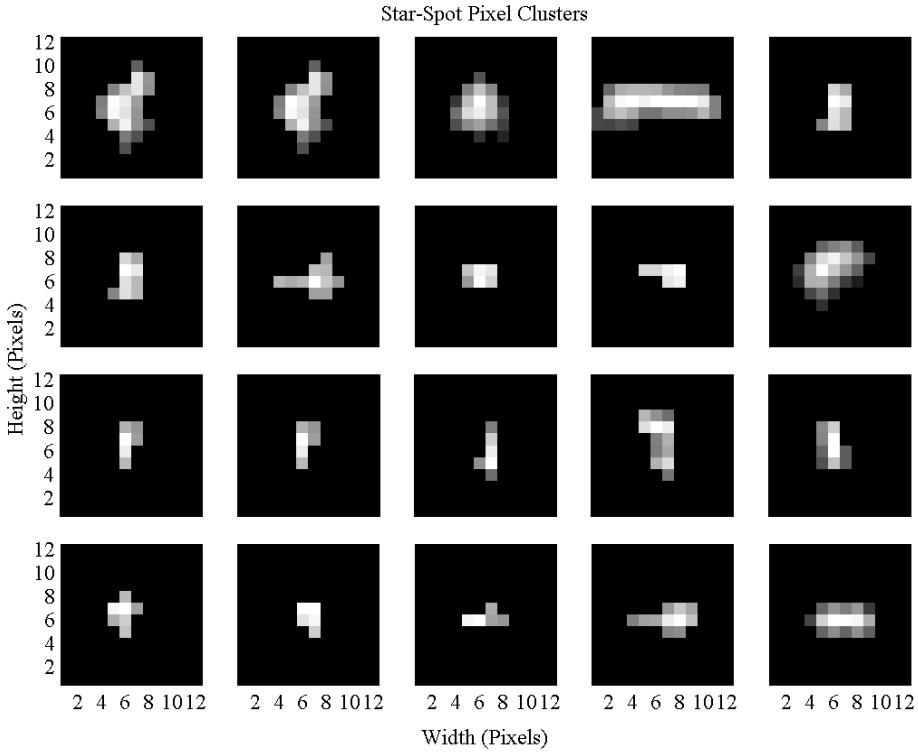

Figure 9.6 Droid X2 - random starlight spot pixel clusters (result set 8)

This can be caused by two possible events: first, the pixels could be very near the threshold value and therefore thresholding leaves only the peak of the spot or second, the sensor noise could be smoothing out the starlight signal since the signal has similar intensity peaks to noise.

Figure 9.7 and Figure 9.8 represent random noise spots from both the Aptina and Droid X2 cameras, respectively. These noise spots were extracted at random among the available noise spots spread throughout the result sets.

The Aptina noise spots were extracted from a result set having a threshold at the noise floor. The noise floor results allowed a small percentage of noise spots to appear in the post image processed spot data; however, just like the non-noise floor results, a high-percentage of these noise spots are true stars at a higher magnitude than that used to run the analysis. Though many of these noise spots are truly starlight spots, they still would be treated as noise if the star identification algorithm is not using a high enough magnitude. Therefore, it is important to use image processing parameters like those used in result sets 1, 2, 4 and 5 such that only starlight spots of equal or less magnitude to that used for identification are extracted.

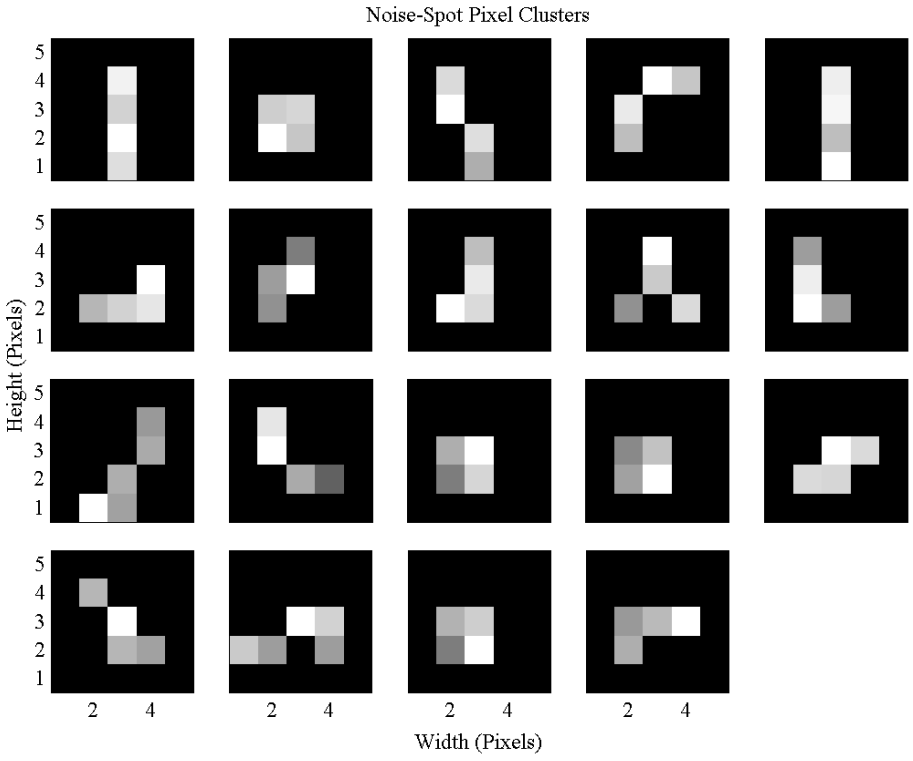

Figure 9.7 Aptina - random noise spot pixel clusters (result set 3)

The Droid X2 noise spots are those typically found in a normal star image. The unfortunate fact that some of the spots are very similar in both area and profile to those of starlight spots reinforces the fact that most of the noise is inseparable from the starlight. There are however distinguishing noise spots which do differ from starlight and have properties such as high eccentricities (long single pixel wide spots) and low flatness factors (near uniform spot pixel intensities). These properties could be used to aid in noise reduction though preliminary tests revealed large amounts of noise still existed after any attempts were made.

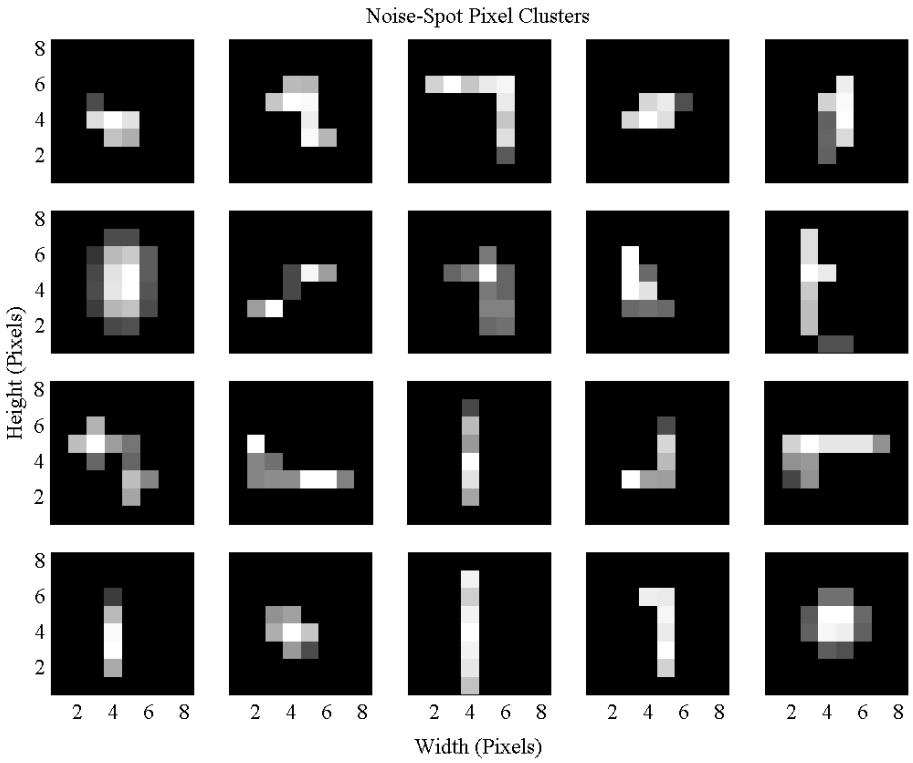

Figure 9.8 Droid X2 - random noise spot pixel clusters (result set 8)

Figure 9.9 and Figure 9.10 track individual starlight spots through time for both the Aptina and Droid X2 cameras respectively. When available the spots were selected at an even time distribution starting from when the star first appeared in the dataset and ending with the stars last appearance in the dataset. The time progression starts with the top left image and ends with the bottom right image.

The Aptina progression shows a general increasing spot area trend as time progresses. This increasing trend can be a result of two factors: (1) the starlight spot's movement across the lens reveals

uncompensated geometric distortion where the smaller spot clusters are a reflection of a smaller amount of aberrational light movement and (2) the starlight spot's movement across the lens reveals uncompensated intensity disturbances. Since intensities were not calibrated in this study, the final outcome will assume a combination of both the above stated factors.

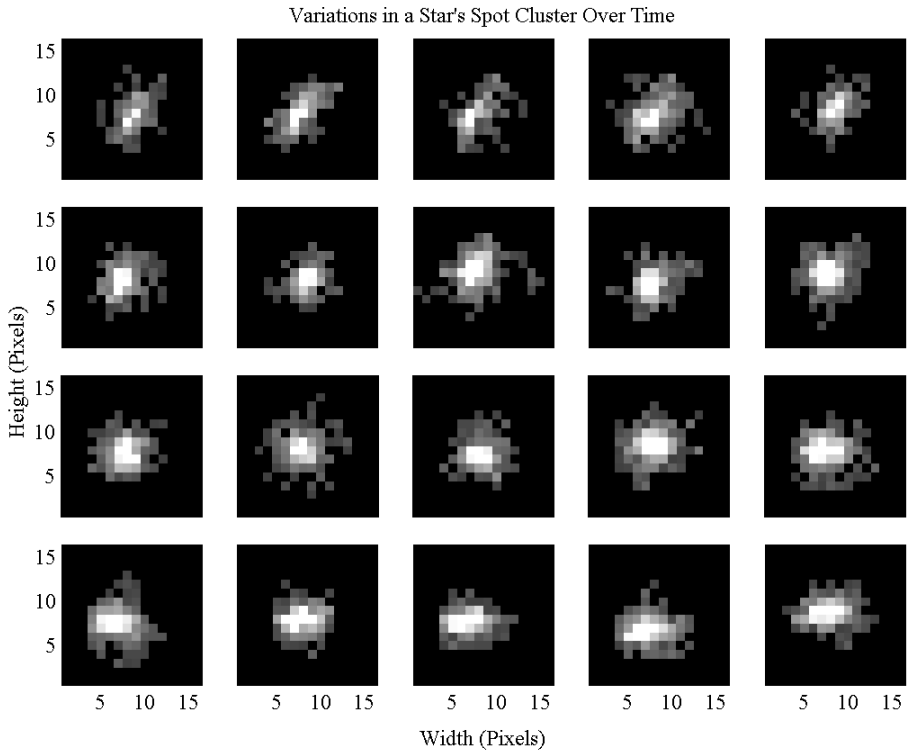

Figure 9.9 Aptina – change in a magnitude 2.07 star's spot pixel clusters over a 180 minute duration (result set 1)

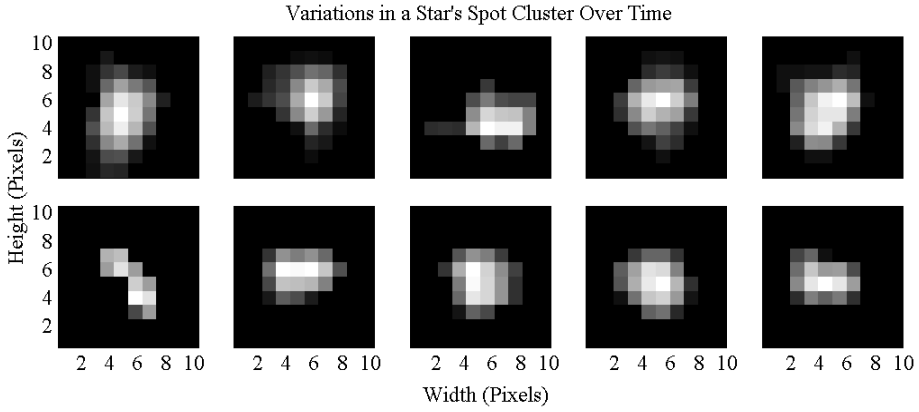

Figure 9.10 Droid X2 – change in a magnitude 0.17 star's spot pixel clusters over a 63 minute duration (result set 8)

The Droid X2 progression shows a nearly uniform propagation spot pixel pattern. The slight changes that do exist may be caused by uncompensated environmental effects such as atmospheric haze (thin clouds) or also changes in the automatic camera parameters such as focus. An issue from the start with the Droid X2 camera was the locked camera settings. Camera settings combined with uncontrollable image processing could cause the slight changes noticed.

B. Shape of Spots

The characteristics of a spot which defines the shape are defined by the following: (1) the intensity distribution of the spot pixels, and (2) the overall area of the spot pixels.

1. Spot Intensity

Figure 9.11 and Figure 9.12 show the spot intensity (sum of all spot pixel intensities) compared to their identified stars' magnitudes. Each vertical scatter pattern corresponds to a unique star and therefore unique starlight spots over time. The figures have a logarithmic volume scale to match the logarithmic nature of the star magnitude definition.

The Aptina camera having a high number of spots identified for the result set shows a highly logarithmic trend for the maximum possible intensity volume for each star. This is shown by the linear trend of the maximum starlight spot volume values when viewed in the logarithmic scale. Also notable is the variation of each star's volume spread across the dataset. The anomaly at magnitude 4.2 is a variable magnitude star which at the time of imaging was brighter than its nominal value.

The Droid X2 camera has a small number of data points however the logarithmic trend is also viewable. The variation of each star's volume spread across the dataset is also notable.

The factors mentioned previously when discussing variations in pixel distributions apply here as well; however, one other factor can also have an effect and that is the variability in star magnitude. Star's have a recorded general magnitude which is the magnitude shown in the comparisons; however, this magnitude varies and the degree of variability and frequency at which is varies is star dependant. The variable nature of a star's magnitude combined with all other uncompensated effects results in the variability in spot intensity.

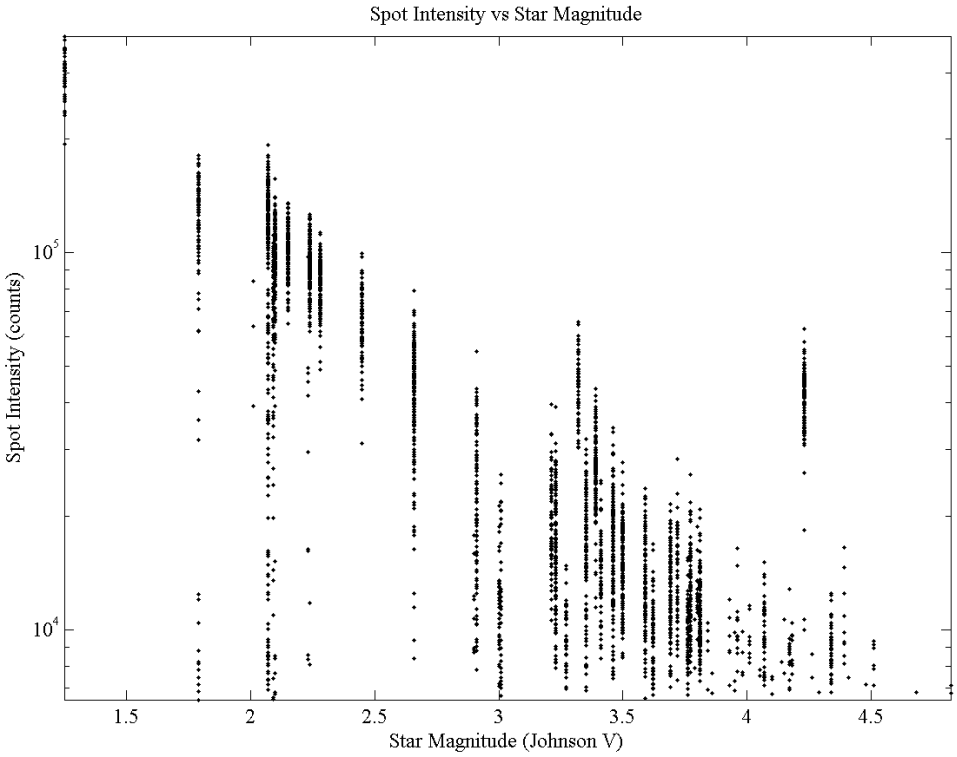

Figure 9.11 Aptina - starlight spot intensity volumes (result set 4)

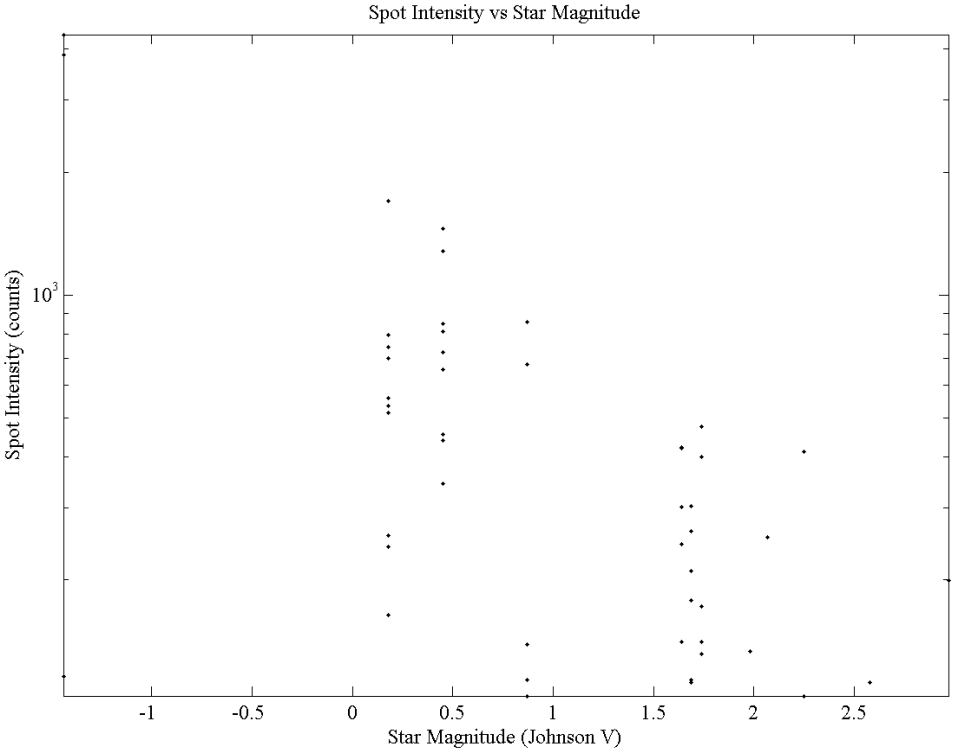

Figure 9.12 Droid X2 - starlight spot intensity volumes (result set 8)

2. Spot Area

Figure 9.13 and Figure 9.14 show the spot areas (number of individual pixels making up each spot) compared to their identified stars' magnitudes. Each vertical scatter pattern corresponds to a unique star and therefore unique starlight spots over time. The figures have a logarithmic area scale to match the logarithmic nature of the star magnitude definition.

Both the Aptina and Droid X2 cameras show the same patterns and trends seen in the starlight spot intensity volume comparisons. Spot area follows the same logarithmic nature as overall intensity volume. The full scale variations from maximum area down to the minimum spot area are clearly visible and are a result of all the previously mentioned factors.

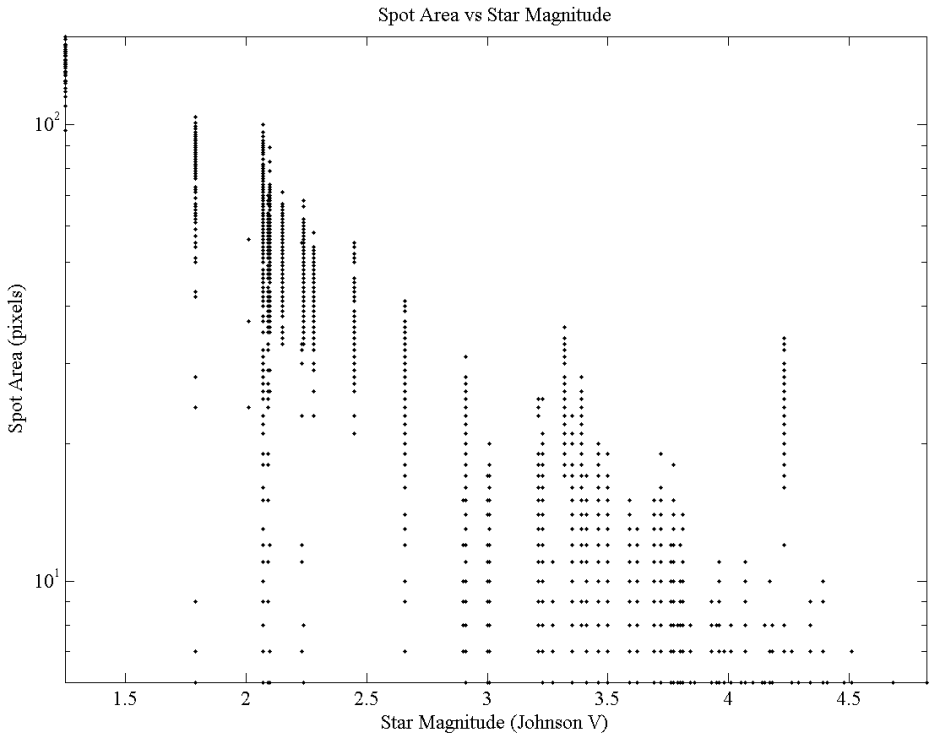

Figure 9.13 Aptina - starlight spot pixel areas (result set 4)

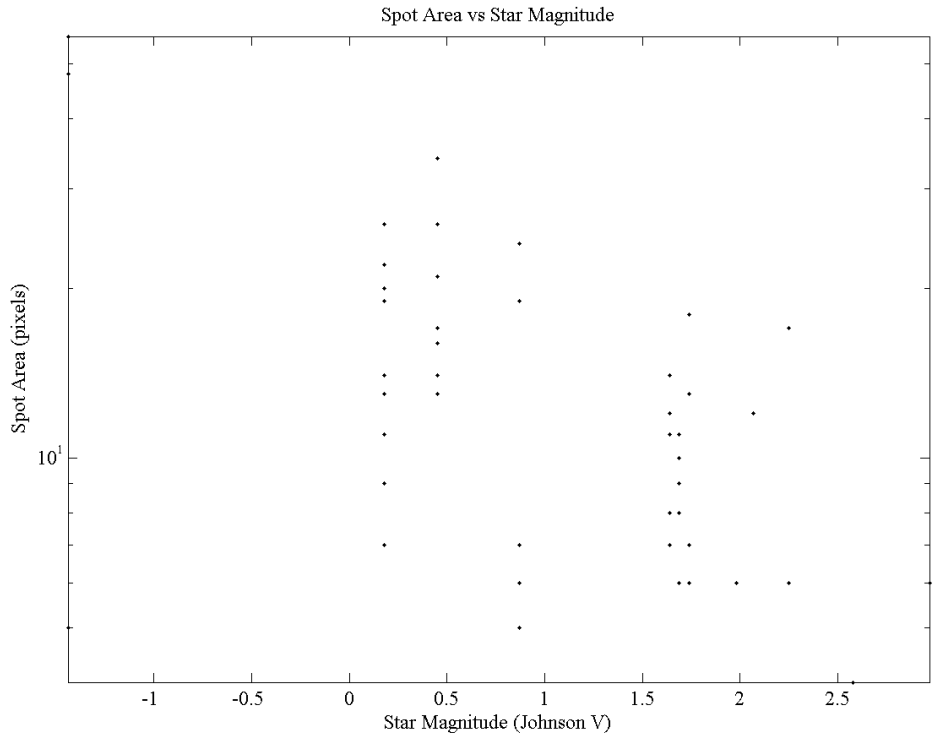

Figure 9.14 Droid X2 - starlight spot pixel areas (result set 8)

C. Summary

Overall many spot characteristic trends are noticeable in both cameras. Appendix C provides additional figures supporting these trends. The Aptina shows a dominant presence in its ability to capture many stars in an image and do it consistently. This allows for clearly distinct trends such as the variability in spot area and shape to be seen. The Droid X2 on the other hand lacks the ability to capture many stars and therefore the data collected, though showing trends, does not have enough statistical backing to provide a thorough understanding of how it captures starlight as the magnitudes of stars' captured are all of high intensity.

VI. Attitude Error

A. Mounting Quaternion Error

The mounting quaternion as defined in Chapter 7 is an average of an entire dataset's individually determined mounting quaternions. A standard error of this average quaternion can also be defined and is a function of the number of images in the dataset.

To define the standard error of the calculated mounting quaternions the Euler angles are first extracted from both the average and individual mounting quaternions. The difference of these Euler angles is then taken which defines the deviation of the individual rotations from the average rotation. The standard error for each Euler angle deviation is then taken and is defined by Eq. (A.9) where σ_a is the standard deviation of the angle deviations and n is the number of data points.

$$S_e = \frac{\sigma_a}{\sqrt{n}} \tag{9.1}$$

The RMS value of the three Euler angle standard errors is then taken. This RMS value defines the overall standard error for the mounting quaternion.

Figure 9.15 and Figure 9.16 show the Euler angle deviations defined in the ECI coordinate frame for the Aptina and Droid X2, respectively.

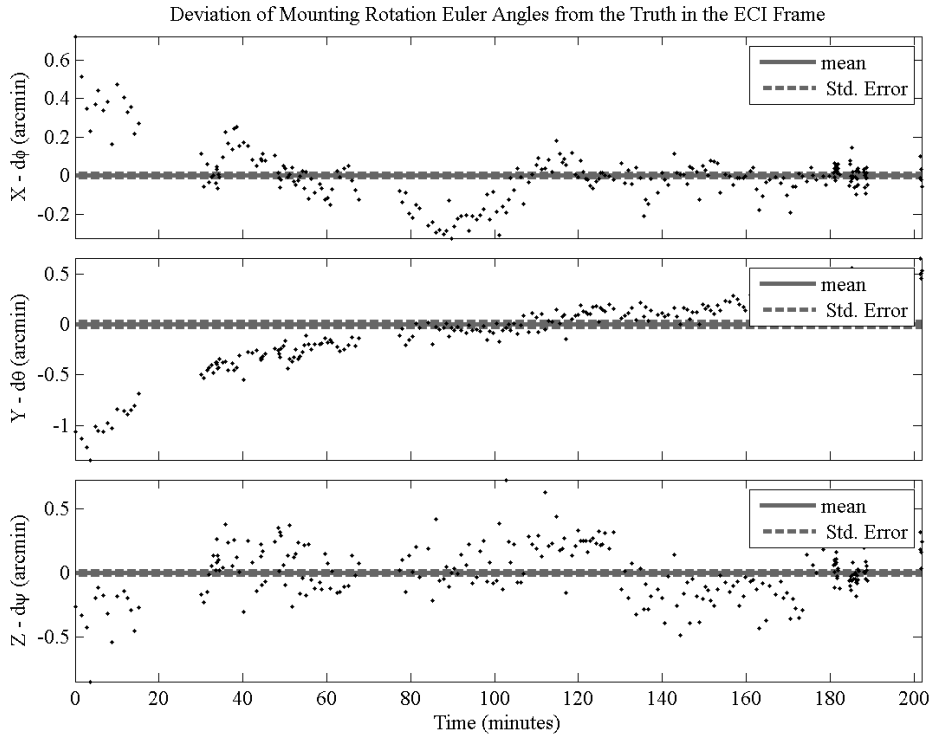

Figure 9.15 Aptina – Euler angle differences between the average and individual mounting rotations (result set 4)

Two observations can be made by comparing these two figures: (1) on average the Aptina has half the angular deviation per Euler angle of the Droid X2 and (2) the standard error of the Aptina is nearly an order of magnitude smaller than for the Droid X2.

It is seen that the standard error of the Aptina mounting quaternion is 0.016 arc-minutes and the standard error of the Droid X2 mounting quaternion is 0.131 arc-minutes. The near order of magnitude difference in mounting error between the Aptina and Droid X2 is primarily due to the increase in valid data points by nearly a factor of 5 between the two result sets. By definition the standard error decreases with increases data points.

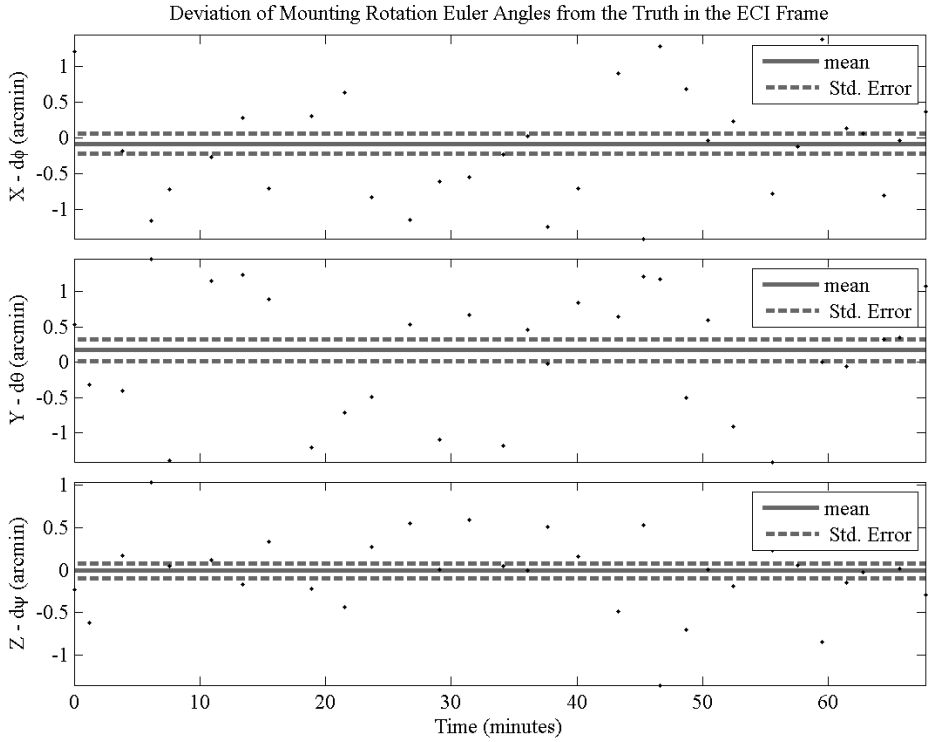

Figure 9.16 Droid X2 – Euler angle differences between the average and individual mounting rotations (result set 8)

B. Total Attitude Error

Figure 9.17 and Figure 9.18 show the total attitude error as well as the pointing and boresight roll components of error for the Aptina and Droid X2, respectively. The total error is the angle defined by the error quaternion of each image in the datasets. The error quaternion is a reference to the error quaternion

defined in Chapter 7. The total attitude error is a culmination of all accounted and unaccounted error sources. Errors which are accounted for include: spot centroiding errors, calibrated radial and tangential lens distortion errors, and atmospheric refraction errors. Error which may be present and are unaccounted for include: ground seeing errors, atmospheric seeing errors, atmospheric scintillation errors and lens vignetting. Though previously defined as having negligible effects on accuracy when combined they could have an impact.

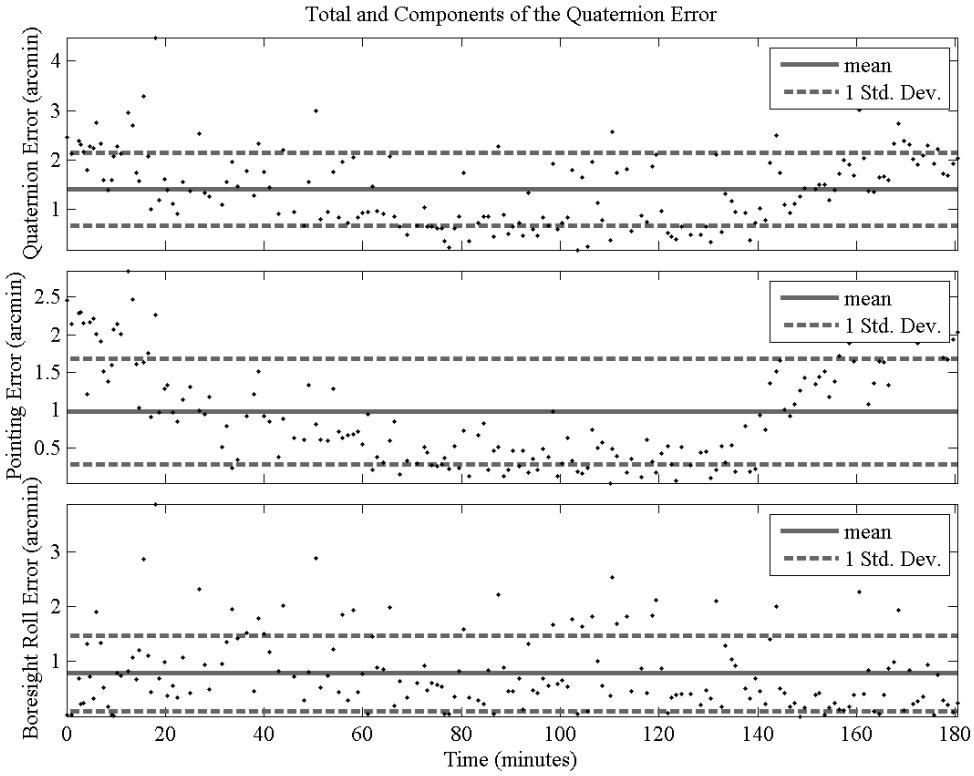

Figure 9.17 Aptina – total and components of the attitude error (result set 1)

The Aptina camera has an average total attitude error of ~1.5 arc-minutes with a standard deviation of ~0.6 arc-minutes. The average pointing error is ~1 arc-minute with a standard deviation of ~0.7 arc-minutes. The average boresight roll error is ~0.75 arc-minutes with a standard-deviation of ~0.7 arc-minutes. There is a drift in the quaternion most noticeable in the pointing error and has a slight parabolic nature. The trend steadily decreases from around 2 arc-minutes to 0.4 arc-minutes levels out for a bit and then increases back toward 2 arc-minutes. Possible explanations for this trend are discussed in the

following sections; however, if this drift is systematic in nature (meaning it is not a part of the camera but instead the environmental factors with which the images were taken) then a tighter bound on the pointing error is evident. This tighter bound is defined by the clustering of error around the drift trend and could be viewed to have an average pointing error of 0.4 arc-minutes and a standard deviation of 0.15 arc-minutes.

The Droid X2 camera has an average total attitude error of ~5.4 arc-minutes with a standard deviation of ~2.1 arc-minutes. The average pointing error is ~2.5 arc-minute with a standard deviation of ~1.4 arc-minutes. The average botesight roll error is ~4.9 arc-minutes with a standard-deviation of ~2 arc-minutes. The error is very scattered and random in nature and thereby should represent a decent estimation of the camera's accuracy. These attitudes, attained by heavy image processing and noise removal, are only as reliable as the processing and number of stars per image. Since most images had only 3 or less stars visible, some of which were represented by very few pixels, it should be noted that these factors may have a high impact on the results.

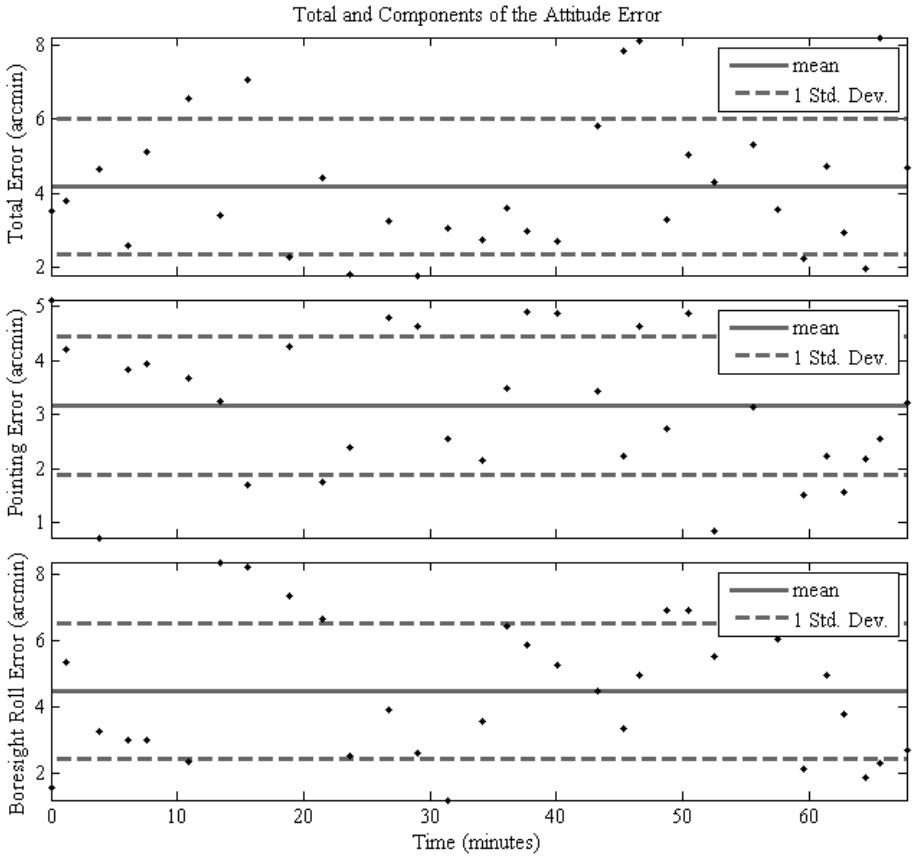

Figure 9.18 Droid X2 – total and components of the attitude error (result set 8)

VII. Focal Plane Influence on Attitude

This section discusses the possible effects the focal plane has on the attitude solution. Specifically it looks at the effects of thresholding and how the threshold changes the calculated spot centroids. Since the spot centroids are used to create camera vectors any changes to these spots will affect the comparisons of these vectors to the identified star vectors which directly affects the attitude error. Since the Aptina results have more starlight spot data than the Droid X2 results it will be used for this comparison.

Figure 9.19 shows the change in spot characteristics (area and intensity volume) as well as the change in individual starlight spot to star vector angular differences. As expected a decrease in the threshold value both increased spot area and intensity volume. The decrease in threshold also caused a decrease in angular deviation on average by ~0.15 arc-minutes. This decrease corresponds to an overall attitude accuracy having slightly less error. This deviation shift is very inconsistent as the figure also shows a large number of spots whose deviation actually increased. It is therefore believed that although thresholding may affect a solution it is only due to other errors which exist in the system such as unaccounted geometric distortion. Physically, when a point source maps its light on a focal plane it should be Gaussian in nature and therefore a centroid should not be affected by a threshold; however, with the geometric errors present in the camera systems, that is not the case and hence the slight changes seen in angular deviation.

VIII. Lens Influence on Attitude

This section discusses the possible effects of lens on the attitude solution. Specifically how calibration affects attitude and whether distortion is still visible in the results after calibration. This section uses only data from the Aptina camera result sets as the number of data point far exceeds that of the Droid X2.

A. Resolution Limitation

As discussed in Chapter 4 optical resolution can either be aberration or diffraction limited. Since stars represent point sources of light the FWHM taken from their spot profile distributions equals the lens optical resolution. Since the FWHM is not star intensity dependent any starlight spot which is properly focused can be used to determine the optical resolution.

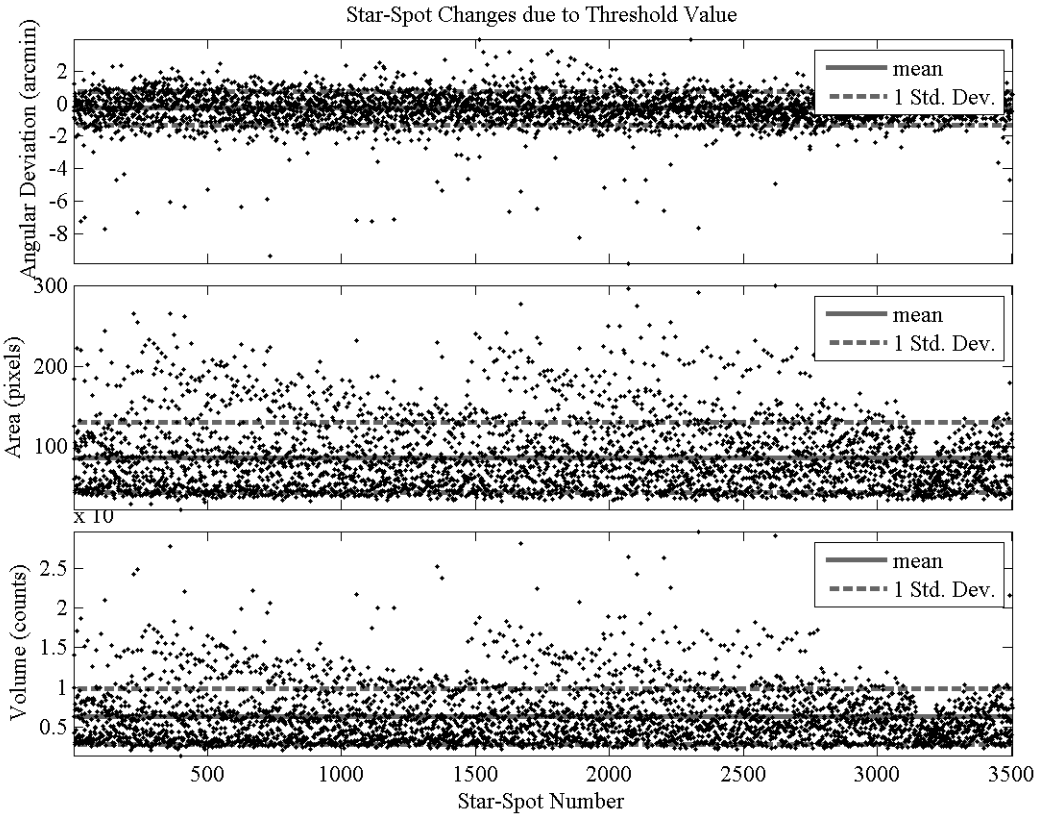

Figure 9.19 Aptina – change in properties due to threshold value (top) mapped angular deviations (middle) spot area (bottom) spot volume (result set 1)

It should be noted that since the Droid X2 images are stored as compressed jpeg files the FWHM is biased. This bias combined with the high threshold applied to the image makes the measured FWHM an underestimate and therefore the true optical resolution is higher.

Figure 9.20 and Figure 9.16 show spot profile distributions for the Aptina and Droid X2 cameras, respectively. As seen the FWHM of the Aptina is 7.6151 pixels which is equal to 8.6812 arc-minutes and the FWHM of the Droid X2 is 3.0054 pixels which is equal to 2.7223 arc-minutes.

To calculate the theoretical diffraction limit Eq. (4.1) is utilized with the light wavelength spread throughout the visible spectrum (445nm to 658nm). The Aptina yields a diffraction limit range of 0.7776 – 1.1499 arc-minutes when its lens is set at the f/2.5 aperture setting (the aperture used data collection). The Droid X2 yields a diffraction limit range of 0.9705 – 1.4350 arc-minutes. Comparing these results to the measured FWHM of both the Aptina and Droid X2 cameras shows that both cameras are aberration limited since the FWHM values are greater than their respective theoretical diffraction limits.

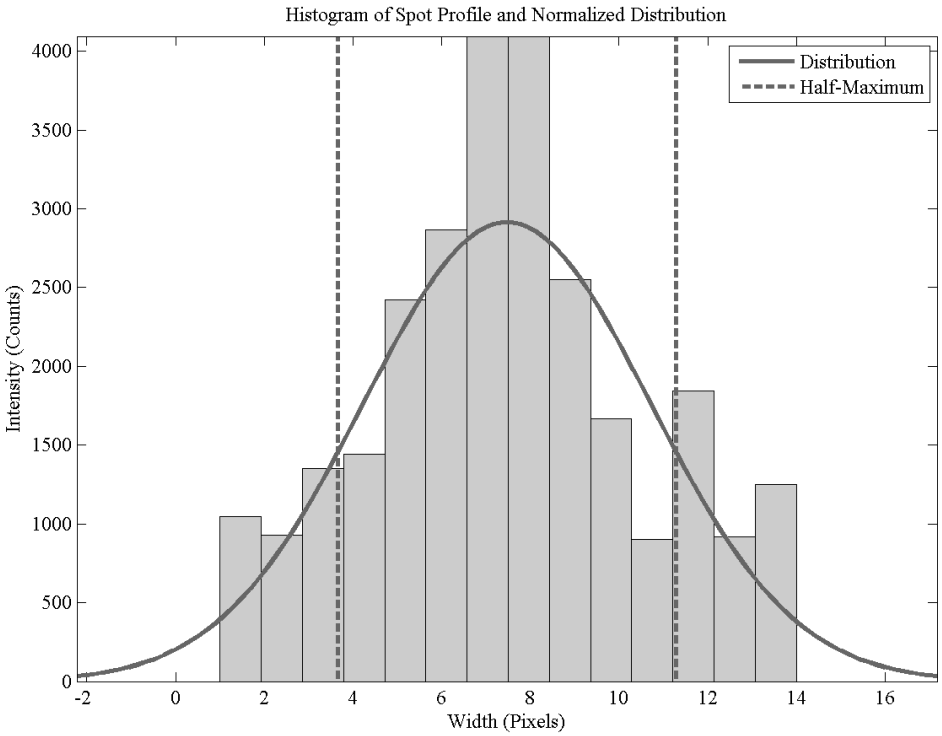

Figure 9.20 Aptina – spot intensity profile detailing the spot distribution and FWHM

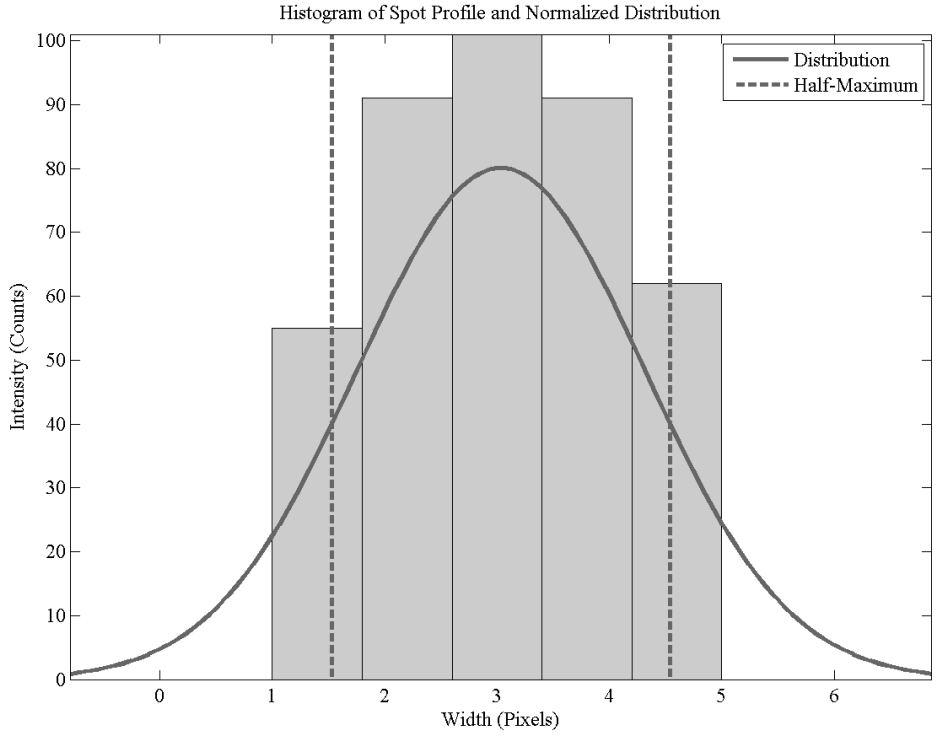

Figure 9.21 Droid X2 – spot intensity profile detailing the spot distribution and FWHM

B. Effects of Lens Correction

Figure 9.22 shows the difference in total attitude error between using a calibrated lens and not using a calibrated lens. It can be seen that lens calibration increases the attitude accuracy by ~3 arc-minutes. This trend also reveals the drift in the quaternion discussed previously. It should be noted that the result sets without lens calibration identify far fewer stars per image and therefore this figure does not show direct comparisons. The decrease in stars per image identified also automatically decreases the attitude error and therefore if both the result sets with and without lens correction identified the same stars the error found would be significantly higher on the order of 15-20 arc-minutes are found in other tests.

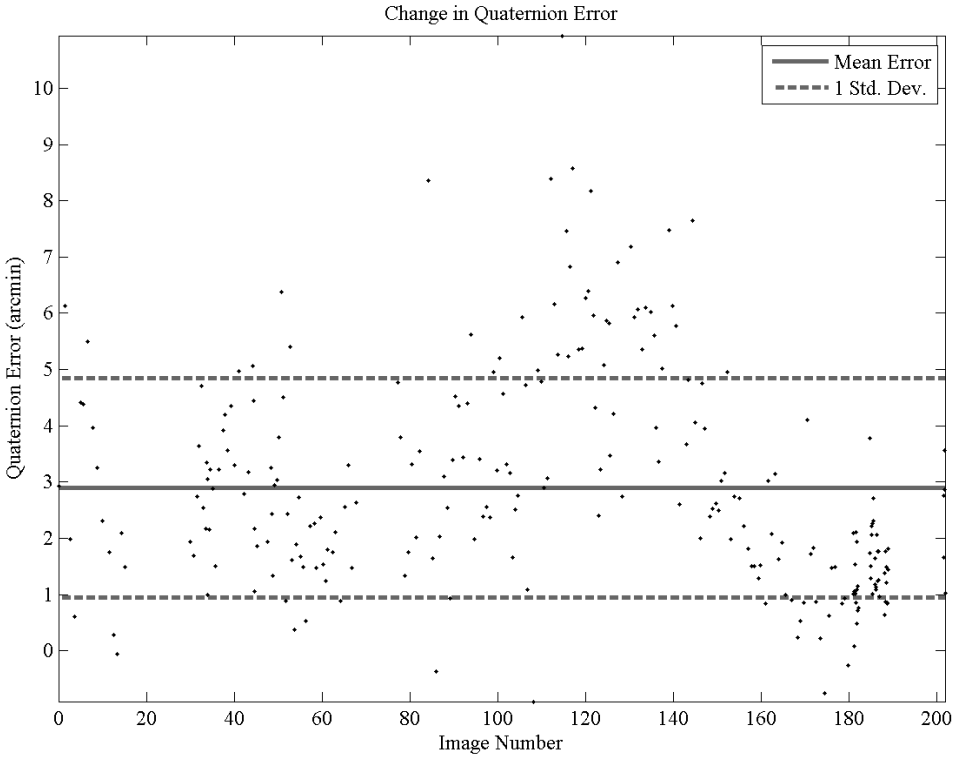

Figure 9.22 Aptina – change in total attitude error without lens correction (result sets 4 and 5)

C. Effects of Radial Distortion

Figure 9.23 shows the average radial distance of starlight spots per image from the camera's bore axis. There is a steady decreasing progression to this value which could provide some insight into the quaternion drift discovered previously. As the radial distance decreases the quaternion error also decreases. This could

show that for the Aptina the quaternion error is dependent on where starlight has passed through the lens. Stars passing closer to the center cause less error than those passing through an edge.

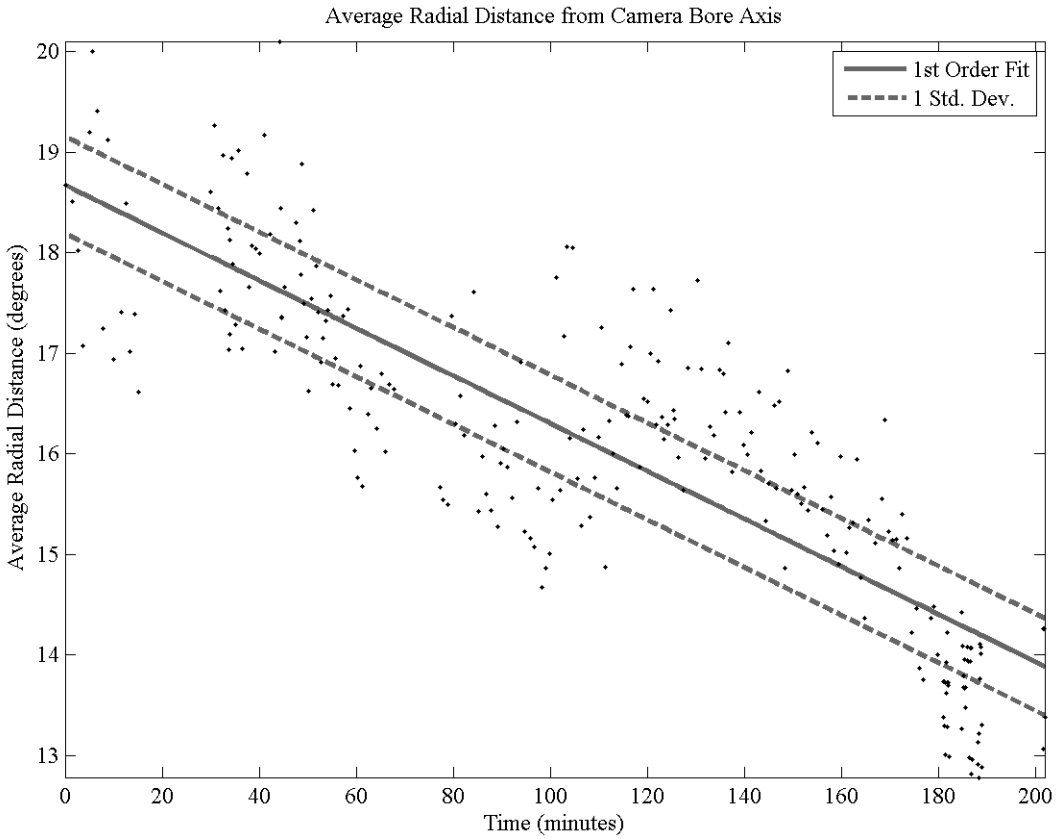

Figure 9.23 Aptina – average spot radial distance from the camera bore axis (result set 1)

Figure 9.24 shows the angular deviation between starlight spot vectors and their mapped star vectors. This trend shows a radially increasing pattern of angular deviation and reinforces the previous idea that stars passing through the lens edge have more error associated with their spots than those passing closer to the center. A second order polynomial fit is shown to define the even order nature of radial distortion and how the error follows the normal physics of radial distortion. This figure shows that radial distortion is still present after calibration and that it does affect the accuracy results.

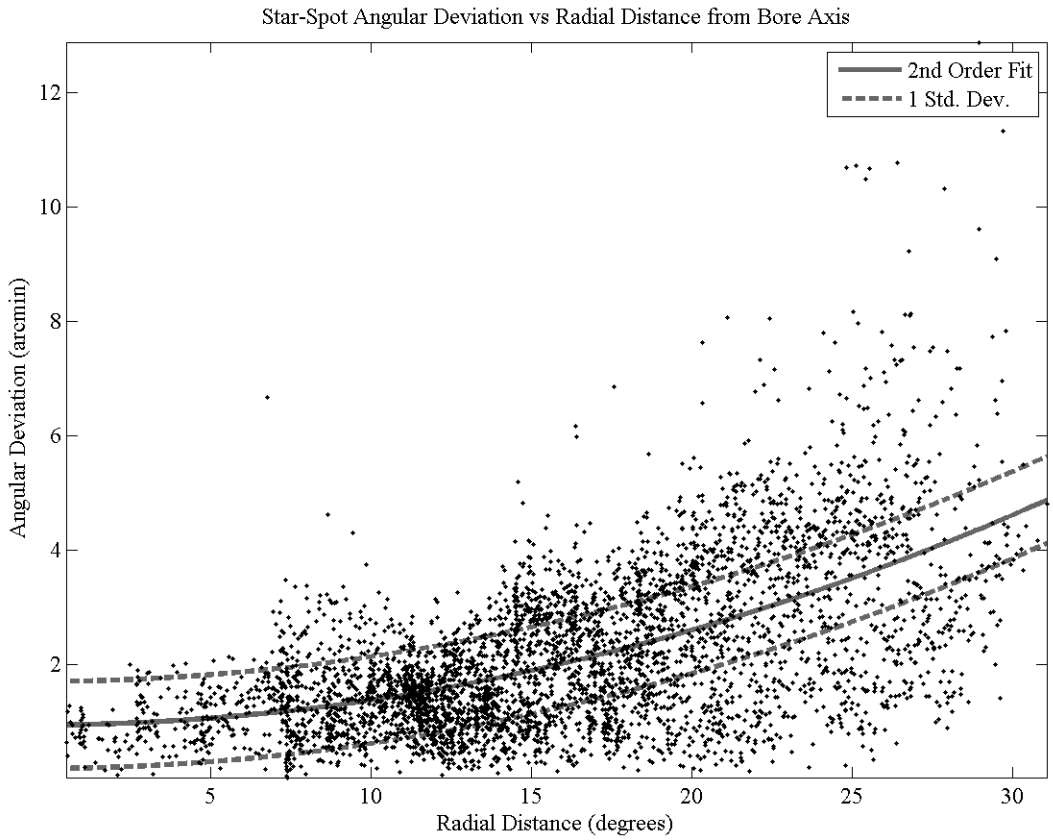

Figure 9.24 Aptina – variations in star to spot angular deviations as a function of radial distance from the camera bore axis (result set 1)

IX. Atmospheric Influence on Attitude

This section discusses the possible effects the atmosphere has on the attitude solution. Specifically how the atmospheric model used for correction effects the attitude accuracy. This section uses only data from the Aptina camera result sets as the number of data point far exceeds that of the Droid X2. Changes due to star altitude angle were also investigated though the results yielded no correlation to attitude accuracy. These investigations can be found in Appendix C.

Figure 9.25 shows the change in quaternion error between a result set with atmospheric correction and one without. It is seen that the attitude accuracy actually increases on average by ~0.02 arc-minutes without atmospheric correction. However, these results are highly variable having a standard deviation of ~0.3 and showing the same drift trends discussed previously. It is therefore believe these results are inconclusive by the fact that they are overshadowed by a larger source of error.

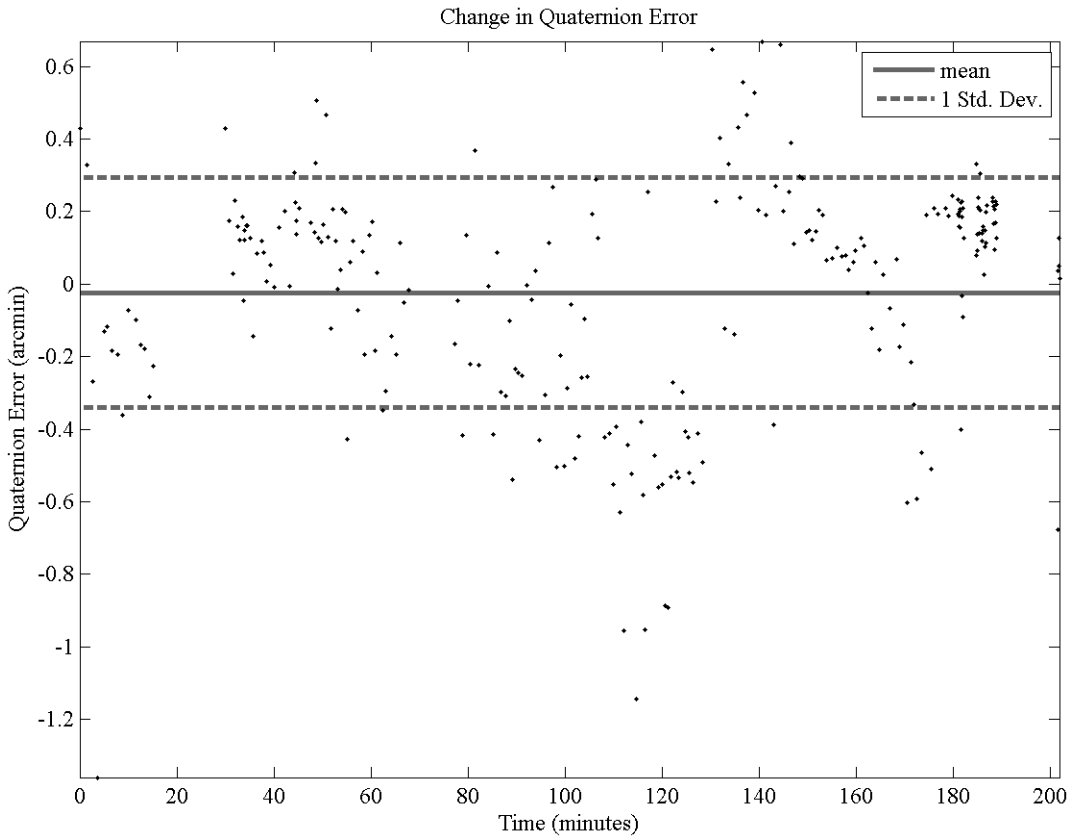

Figure 9.25 Aptina - change in total attitude error without atmospheric correction (result sets 3 and 4)

X. Summary

Both the Aptina and Droid X2 cameras are able to view starlight as spots on their focal planes. The Aptina results show distinctive differences between starlight spot areas and flatness factors and those of noise spots. They also show low concentrations of noise spots. The Droid X2 does not reveal any distinctive trends and is also dominated by noise spots.

Attitude accuracy is ~ 4 arc-minutes more accurate for the Aptina than for the Droid X2. Furthermore, the Droid X2 also requires a high amount of image processing and manual noise removal techniques. For these reasons it would make it impossible to use the Droid X2 at the tested camera settings in a real-time envronoment.

A breakdown of the components of error described in this chapter is shown in Table 9.5. The table shows both the total error as well as the four primary error components: (1) error in the truth mounting quaternion, (2) error due to thresholding, (3) error due to the lens and (4) error due to the atmosphere.

Table 9.5 Total camera error breakdown given in arc-minutes

Camera	Total Error	Truth Error	Threshold Error	Lens Error	Atmospheric Error
Aptina	1.5 +/- 0.6	0.0163	0.15	0.29	0.07
Droid X2	5.4 +/- 2.1	0.1309	0.15	0.58	0.07

It is shown that the summation of the defined components of error for each camera is within two standard deviations of their respective total errors. This verifies that the calculated total errors agree with the culmination of the analytical model errors, the calibration errors and the errors defined in the analysis.

It is shown that the dominant sources of error influencing the attitude accuracies are the lenses of the cameras. Atmospheric and thresholding compensations do not have significant components in the error. Unaccounted error may also be involved in the error. Other possible error sources include: atmospheric turbulence, ground heat seeing effects and stray clouds all of which could be affecting the solution.

CHAPTER 10

CONCLUSIONS

I. Calibration and Correction Methods

Camera lens calibration was accomplished by passing checkerboard images through a 6^{th} order lens distortion model. The model included both radial and tangential distortion components. It was discovered that the higher-order distortion terms on both cameras were negligible and therefore excluded from their final calibration results. The calibration process concluded with the Aptina imager having an accuracy of ~0.25 pixels (~0.28 arc-minutes) and the Droid X2 having an accuracy of ~0.65 pixels (~0.58 arc-minutes).

Atmospheric correction was based on a pure refraction model and did not include any variability due to turbulent fluids. The refraction model yield inconclusive results as to its ability to correct images. Some images experienced an increase in accuracy while others experienced the opposite. Due to the complexity of modeling the atmosphere it is believed that a pure refractive model does not provide adequate correction for images.

For future work, lens calibrations should be performed with large grid sizes due to their distinguished high-contrast near grid corners. This higher and broader contrast level allows the corner extraction method to find the corner with a higher degree of accuracy. The decrease in the number of corners due to the larger grid size can be made up for by increasing the number of overall calibration images yielding a similar number of individual calibration points to the smaller grid size. For atmospheric correction it is advised to either incorporate a more advanced correction model or take images when atmospheric effects are at a minimum.

II. Image Processing Software

The software to both process and analyze star images was developed in MATLAB as a GUI. This presentation style gives users of all backgrounds the ability to process star data. Though many similar tools exist, some of which have exceedingly more options, none of them work well or at all for wide FOV

cameras with significant amounts of noise. The developed tool provided the means of extracting the desired accuracy data out of a series of star images.

III. Smartphone Camera Abilities

The abilities of smartphone cameras were shown to be insufficient for spacecraft applications using the acquisition settings attainable in this research. Noise dominated over most all the starlight's signal. Where stars were found they did not exceed a 2.6 magnitude. This alone is enough to prevent this class of imager from being reliable star cameras as a 2.6 magnitude limit combined with a 50 degree FOV results in areas of the sky where there will not be enough star to obtain an attitude.

Though the results in this research are conclusive to the inabilities of smartphone imagers there is one key factor which mandated this result and that is the exposure time. Limited by an exposure time of 66 milli-seconds this class imager was shown to fail at accurately obtaining usable star data. However, the actual imager specifications permit exposure times up to 1 second and therefore when correlated to data from the Aptina it is very realistic to assume that a longer exposure time would make these imagers valuable. For future work, obtaining the ability to capture longer exposures and collecting new star images based on this new exposure would provide the results needed to make any final conclusions on smartphone imagers.

IV. Accuracy of Low-Cost Imagers

The attitude accuracy of the Aptina camera is 1.5 arc-minutes using the procedures of this research. The pointing and rotational accuracies are 1 arc-minute and 0.75 arc-minutes, respectively. Though these accuracies are decent for this class imager it is felt that with higher quality calibrations even better attitude accuracy could be obtained.

The attitude accuracy of the Droid X2 camera is 5.5 arc-minutes using the low starlight spot count images obtained in this research. The pointing and rotational accuracies are 2.5 arc-minutes and 4.8 arc-minutes respectively. In the future the tests should be repeated with a similar camera where exposure time

is fully controllable. The exposure time tests would allow for a more thorough understanding of the imager's capabilities as both more spots and a higher intensity count would be available.

V. Components of Error

The components of the attitude error were only analyzed for the Aptina camera as the Droid X2 did not provide enough starlight spot data per image to provide reliable results. The results show dominance in lens error effects. The lens displays radial distortion after calibration and this distortion is shown to have a direct influence on the accuracy of the calculated starlight spot vectors. Error due to both atmospheric refraction and image thresholding are negligible and had results yielding indecisive behavior.

Other sources of error may exist which have not be studied in this research. They include ground heat radiant seeing, atmospheric seeing and lens intensity mappings among others. These may have roles in the total error found in this study.

REFERENCES

[1] NASA, *Spacecraft Star Trackers*, Houston, TX, 1970.

[2] Lee, S., Ortiz, G., and Alexander, J., "Star Tracker-Based Acquisition, Tracking, and Pointing Technology for Deep-Space Optical Communications," *Interplanetary Network Progress Report*, Vol. 42-161, 2005, pp. 1–18.

[3] Ball Aerospace, *CT-602 Star Tracker*, Boulder, CO, 2013.

[4] Ball Aerospace, *CT-601 High Accuracy Star Tracker*, Boulder, CO, 1995.

[5] TERMA, *Terma HE-5AS Star Tracker*.

[6] Percival, J. W., and Nordsieck, K. H., "The ST5000: An Attitude Determination System with Low-Bandwidth Digital Imaging," *Astronomical Data Analysis Software and Systems X*, Madison, WI, 2001, p. 463.

[7] Ball Aerospace, *CT-601 High Accuracy Star Tracker*, Boulder, CO, 1995.

[8] Ball Aerospace, *CT-633 Stellar Attitude Sensor*, Boulder, CO, 2013.

[9] Blue Canyon Technologies, *Blue Canyon Technologies XACT Datasheet*, Boulder, CO, 2012.

[10] Berlin Space Technologies, *Star Tracker ST-200*, 2012.

[11] Space Dynamics Laboratory, *Star Camera - Compact, Light-Weight, & Low Power Star Imaging for Nano- & Pico-Satellites*, Logan, UT, 2012.

[12] Singla, P., Griffith, D. T., Crassidis, J. L., Junkins, J. L., and Antonio, S., "Attitude Determination and Autonomous On-Orbit Calibration of Star Tracker for the GIFTS Mission," *AAS/AIAA Space Flight Mechanics Meeting*, 2002, pp. 1–20.

[13] OmniVision, *Ov8810/12 8-Megapixel Product Brief*, Santa Clara, CA, 2008.

[14] Micron Technology, *MT9P031: 1/2.5-inch 5MP CMOS Digital Image Sensor Features*, 2006.

[15] Micron Technology, *Image Sensor Demo System Kits*, 2006.

[16] Danko, A., "ClearDarkSky," 2012.

[17] dev.at.a...@gmail.com, "androidscreencast."

[18] Truesense Imaging Inc., *Image Sensor ISO Measurement Reference Document*, 2012.

[19] Kodak, *CCD Image Sensor Noise Sources*, 2005.

[20] Goodrich SUI, *The Sensitivity of Focal Plane Arrays and Cameras*, 2006.

[21] Truesense Imaging Inc., *Conversion of Light to Electronic Charge Reference Document*, 2012.

[22] Cao, Y., and Zhang, X., "An on-chip hot pixel identification and correction approach in CMOS imagers," *IEEE*, 2011, pp. 408–411.

[23] Babic, Z. V, and Mandic, D. P., "An Efficient Noise Removal and Edge Preserving Convolution Filter," *TELSIKS*, 2003, pp. 538–541.

[24] Perona, P., and Malik, J., "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 12, Jul. 1990, pp. 629–639.

[25] Liebe, C., "Accuracy Performance of Star Trackers-A Tutorial," *IEEE Transactions on and Electronic Systems*, Vol. 38, 2002, pp. 587–599.

[26] Samaan, M. A., Mortari, D., Pollock, T. C., and Junkins, J. L., *Predictive Centroiding for Single and Multiple FOVs Star Trackers*, College Station, TX, 2002.

[27] CVI Melles Griot, *Fundamental Optics*, Albuquerque, NM, 2009.

[28] Powell, J., and Southall, C., *Mirrors Prisms and Lens*, The MacMillan Company, New York, NY, 1918.

[29] ElfQrin, "Image: Lenses en.svg," 2011.

[30] Chao, M., Chengt, S., and Hsing, C., "Aspheric Lens Design," *IEEE Ultrasonics Symposium*, Vol. 2, 2000, pp. 1025–1028.

[31] Fryer, J. G., "Lens Distortion for Close Range Photogrammetry," *Photometric Engineering*, Vol. 37, 1971, pp. 855–866.

[32] Pers, J., and Kovacic, S., "Nonparametric, Model-Based Radial Lens Distortion Correction Using Tilted Camera Assumption," *Computer Vision*, Bad Aussee, Austria, 2002, pp. 286–295.

[33] Heikkila, J., and Silven, O., "A Four-Step Camera Calibration Procedure with Implicit Image Correction," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Comput. Soc, 1997, pp. 1106–1112.

[34] Tsai, R., "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses," *IEEE Journal on Robotics and Automation*, Vol. 3, Aug. 1987, pp. 323–344.

[35] Zhang, Z., and Way, O. M., "Flexible Camera Calibration by Viewing a Plane from Unknown Orientations," *IEEE*, Vol. 1, 1999, pp. 666–673.

[36] Zhang, Z., and Member, S., "A Flexible New Technique for Camera Calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 22, 2000, pp. 1330–1334.

[37] Zhang, Z., "A Flexible New Technique for Camera Calibration," *Constraints*, Vol. 1998, 2008.

[38] Shen, J., Zhang, G., and Wei, X., "Star Sensor On-Orbit Calibration Using Extended Kalman Filter," *Star*, 2010, pp. 958–962.

[39] University of Oulu, "Calibration Toolbox for Matlab," University of Oulu, Oulu, Finland, 2000.

[40] Bouguet, J.-Y., "Camera Calibration Toolbox for Matlab," Intel Corporation Microprocessor Research Labs, Santa Clara, CA, 2003.

[41] Bouguet, J.-Y., “First Calibration Example - Corner Extraction, Calibration, Additional Tools” Available: http://robots.stanford.edu/cs223b04/JeanYvesCalib/htmls/example.html.

[42] Garfinkel, B., “Astronomical Refraction in a Polytropic Atmosphere,” *Astronomical Journal*, Vol. 72, 1967, pp. 235–254.

[43] Night, F., “Seeing Forecast For Astronomical Purposes,” *Astronomy*, 2011, pp. 14–16.

[44] Meeus, J., *Astronomical Algorithms*, Willmann-Bell Incorporated, Richmond, Virginia, 1991.

[45] Government of Canada, “Weather forecast for astronomy” Available: http://weather.gc.ca/astro/index_e.html.

[46] Coulman, C. E., “Fundamental and Applied Aspects of Astronomical Seeing,” *Ann. Rev. Astron. Astrophys.*, Vol. 23, 1985, pp. 19–57.

[47] Buscher, D. F., Armstrong, J. T., Hummel, C. a, Quirrenbach, A., Mozurkewich, D., Johnston, K. J., Denison, C. S., Colavita, M. M., and Shao, M., “Interferometric Seeing Measurements on Mt. Wilson: Power Spectra and Outer Scales,” *Applied Optics*, Vol. 34, Feb. 1995, pp. 1081–96.

[48] Curtis, H. D., *Orbital Mechanics for Engineering Students*, Elsevier Ltd., Burlington, MA, 2009.

[49] Utah State University, “Utah Climate Center” Available: http://climate.usurf.usu.edu/index.php.

[50] Weather Underground, “Weather Underground” Available: http://www.wunderground.com/.

[51] Brätt, S. P., “Error Modeling and Analysis of Star Identification Algorithms for Low Quality Star Cameras,” Utah State University, 2013.

[52] Black, H., “A Passive System for Determining the Attitude of a Satellite,” *AIAA Journal*, Vol. 2, 1964, pp. 1350–1351.

[53] Crassidis, J. L., Markley, F. L., and Cheng, Y., “Survey of Nonlinear Attitude Estimation Methods,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, Jan. 2007, pp. 12–28.

[54] Markley, F., and Mortari, D., “Quaternion attitude estimation using vector observations.,” *Journal of the Astronautical Sciences*, Vol. 48, 2000, pp. 359–380.

[55] SHUSTER, M., “Maximum likelihood estimation of spacecraft attitude,” *Journal of the Astronautical Sciences*, Vol. 37, 1989, pp. 79–88.

[56] Kim, S.-G., Crassidis, J. L., Cheng, Y., Fosbury, A. M., and Junkins, J. L., “Kalman Filtering for Relative Spacecraft Attitude and Position Estimation,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, Jan. 2007, pp. 133–143.

[57] Markley, F. L., Cheng, Y., Crassidis, J. L., and Oshman, Y., “Averaging Quaternions,” *Journal of Guidance, Control, and Dynamics*, Vol. 30, Jul. 2007, pp. 1193–1197.

[58] Gupta, S., “Linear Quaternion Equations with Application to Spacecraft Attitude Propagation,” *Aerospace Conference, 1998 IEEE*, 1998, pp. 69–76.

[59] Shuster, M., and Natanson, G., “Quaternion Computation from a Geometric Point of View,” *Journal of the Astronautical Sciences*, Vol. 41, 1993, pp. 545–556.

APPENDICES

APPENDIX A

QUATERNIONS

I. Quaternion Identities

A. Quaternion

A quaternion is a four-dimensional vector which is defined as

$$\bar{q} \equiv \begin{bmatrix} \bar{\rho} \\ q_s \end{bmatrix} \tag{A.1}$$

where

$$\bar{\rho} \equiv [q_x \quad q_y \quad q_z]^T = \bar{e} \sin\left(\frac{\nu}{2}\right), \qquad q_s = \cos\left(\frac{\nu}{2}\right) \tag{A.2}$$

and where e is the unit Euler axis and ν is the rotation angle.

B. Quaternion Norm

The quaternion norm is defined as

$$\|\bar{q}\| \equiv \sqrt{q_x^2 + q_y^2 + q_z^2 + q_s^2} \tag{A.3}$$

C. Quaternion Conjugate

The quaternion conjugate is defined as

$$\bar{q}^* \equiv \begin{bmatrix} -\bar{\rho} \\ q_s \end{bmatrix} \tag{A.4}$$

D. Inverse Quaternion

The inverse quaternion is defined as

$$\bar{q}^{-1} \equiv \frac{q^*}{\|\bar{q}\|} \tag{A.5}$$

E. Cross-Product Matrix

$[\bar{\rho}\times]$ is the cross-product matrix defined by

$$[\bar{\rho}\times] \equiv \begin{bmatrix} 0 & -q_z & q_y \\ q_z & 0 & -q_x \\ -q_y & q_x & 0 \end{bmatrix} \tag{A.6}$$

F. Attitude Matrix

The attitude matrix is related to the quaternion by

$$A(\bar{q}) = (q_s^2 - \|\bar{\rho}\|^2)I_{3\times3} + 2\bar{\rho}\bar{\rho}^T - 2q_s[\rho\times] = \Xi^T(\bar{q})\Psi(\bar{q}) \tag{A.7}$$

where

$$\Xi^T(\bar{q}) \equiv \begin{bmatrix} q_s I_{3\times3} + [\bar{\rho}\times] \\ -\bar{\rho}^T \end{bmatrix}, \qquad \Psi(\bar{q}) \equiv \begin{bmatrix} q_s I_{3\times3} - [\bar{\rho}\times] \\ -\bar{\rho}^T \end{bmatrix} \tag{A.8}$$

and $I_{3\times3}$ is the identity matrix.

G. Quaternion Multiplication

Quaternion multiplication is defined by

$$\bar{p} \otimes \bar{q} = [\Xi^T(\bar{q}) \quad \bar{q}]\bar{p} = [\Psi(\bar{p}) \quad \bar{p}]\bar{q} = \begin{bmatrix} p_x q_s + p_s q_x + p_y q_z - p_z q_y \\ p_y q_s + p_s q_y + p_z q_x - p_x q_z \\ p_z q_s + p_s q_z + p_x q_y - p_y q_x \\ p_s q_s - p_x q_x - p_y q_y - p_z q_z \end{bmatrix} \tag{A.9}$$

H. Quaternion Exponential

Quaternion exponentials are defined by

$$e^{\bar{q}} = e^{q_s}\begin{bmatrix} \cos(\|\bar{\rho}\|) \\ \dfrac{\bar{\rho}}{\|\bar{\rho}\|}\sin(\|\bar{\rho}\|) \end{bmatrix} \tag{A.10}$$

I. Quaternion Rotation

Written with respect to rotation matrices (A) it is seen that sequential rotations are defined by performing quaternion multiplication between the rotation quaternions.

$$A(\bar{p})A(\bar{q}) = A(\bar{p} \otimes \bar{q}) \tag{A.11}$$

J. Quaternion Kinematics Equation

The quaternion kinematics equation is defined by

$$\dot{\bar{q}} = \frac{1}{2}\,\Xi(\bar{\mathrm{q}})\bar{\omega} = \frac{1}{2}\Omega(\bar{\omega})\bar{\mathrm{q}} \tag{A.12}$$

where $\bar{\omega}$ is the angular rate vector ($[\omega_x \quad \omega_y \quad \omega_z]^T$) and

$$\Omega(\bar{\omega}) \equiv \begin{bmatrix} -[\bar{\omega}\times] & \bar{\omega} \\ -\bar{\omega}^{\mathrm{T}} & 0 \end{bmatrix} = [\omega_x \quad \omega_y \quad \omega_z \quad 0]^T \tag{A.13}$$

K. Functional Operations

The following is a mathematical identity of quaternions

$$\Psi(\bar{q})\bar{\omega} = \Gamma(\bar{\omega})\bar{q} \tag{A.14}$$

II. Direction Cosine Matrix Conversion

Various algorithms exist to convert a direction cosine matrix to a quaternion some of which are defined in [59].

The following equation converts a quaternion to a direction cosine matrix

$$A(\bar{q}) = \begin{bmatrix} q_s^2 + q_x^2 - q_y^2 - q_z^2 & 2q_xq_y - 2q_sq_z & 2q_xq_z + 2q_sq_y \\ 2q_xq_y + 2q_sq_z & q_s^2 - q_x^2 + q_y^2 - q_z^2 & 2q_yq_z - 2q_sq_x \\ 2q_xq_z - 2q_sq_y & 2q_yq_z + 2q_sq_x & q_s^2 - q_x^2 - q_y^2 + q_z^2 \end{bmatrix} \tag{A.15}$$

APPENDIX B

IMAGE DATASETS

Table B.1 Star image datasets

Date	# of Images	Camera	Sequential	Latitude	Longitude	Directory Name
5/20/2010	11	Aptina	No	41.742597	-111.842483	2010_05_20
5/31/2010	45	Aptina	No	41.742597	-111.842483	2010_05_31
1/18/2011	6	Aptina	No	41.742597	-111.842483	2011_01_18
2/2/2011	33	Aptina	No	41.742597	-111.842483	2011_02_02
2/8/2011	27	Aptina	No	41.742597	-111.842483	2011_02_08
5/14/2011	27	Aptina	No	41.742597	-111.842483	2011_05_14
6/20/2012	81	Aptina	No	41.742597	-111.842483	2012_06_20
10/5/2012	170	Aptina	Yes	41.771948	-111.749832	2012_10_05
11/6/2012	252	Aptina	Yes	41.771948	-111.749832	2012_11_06
9/2/2011	12	Droid X2	No	41.742597	-111.842483	2011_9_2
9/17/2011	30	Droid X2	No	41.742597	-111.842483	2011_9_17
9/23/2011	17	Droid X2	No	41.742597	-111.842483	2011_9_23
9/29/2011	22	Droid X2	No	41.742597	-111.842483	2011_9_29
9/30/2011	14	Droid X2	No	41.742597	-111.842483	2011_9_30
10/14/2011	12	Droid X2	No	41.742597	-111.842483	2011_10_14
10/17/2011	18	Droid X2	No	41.742597	-111.842483	2011_10_17
10/29/2011	9	Droid X2	No	41.742597	-111.842483	2011_10_29
12/7/2011	62	Droid X2	No	41.742597	-111.842483	2011_12_07
1/16/2012	5	Droid X2	No	41.742597	-111.842483	2012_1_16
3/22/2012	381	Droid X2	No	41.742597	-111.842483	2012_3_22
3/30/2012	18	Droid X2	No	38.77356	-109.58103	2012_3_30
5/10/2012	21	Droid X2	No	41.742597	-111.842483	2012_5_10
5/14/2012	67	Droid X2	No	41.742597	-111.842483	2012_5_14
5/20/2012	4	Droid X2	No	41.742597	-111.842483	2012_5_20
6/20/2012	81	Droid X2	No	41.771948	-111.749832	2012_6_20
9/3/2012	2	Droid X2	No	41.742597	-111.842483	2012_9_3
10/5/2012	213	Droid X2	Yes	41.771948	-111.749832	2012_10_05
11/6/2012	238	Droid X2	Yes	41.771948	-111.749832	2012_11_06

APPENDIX C

ADDITIONAL RESULTS

This section contains additional results not shown in the image analysis chapter. These results provide reinforcement to the ideas and conclusions drawn on the tested cameras.

I. Aptina – Result Set 4

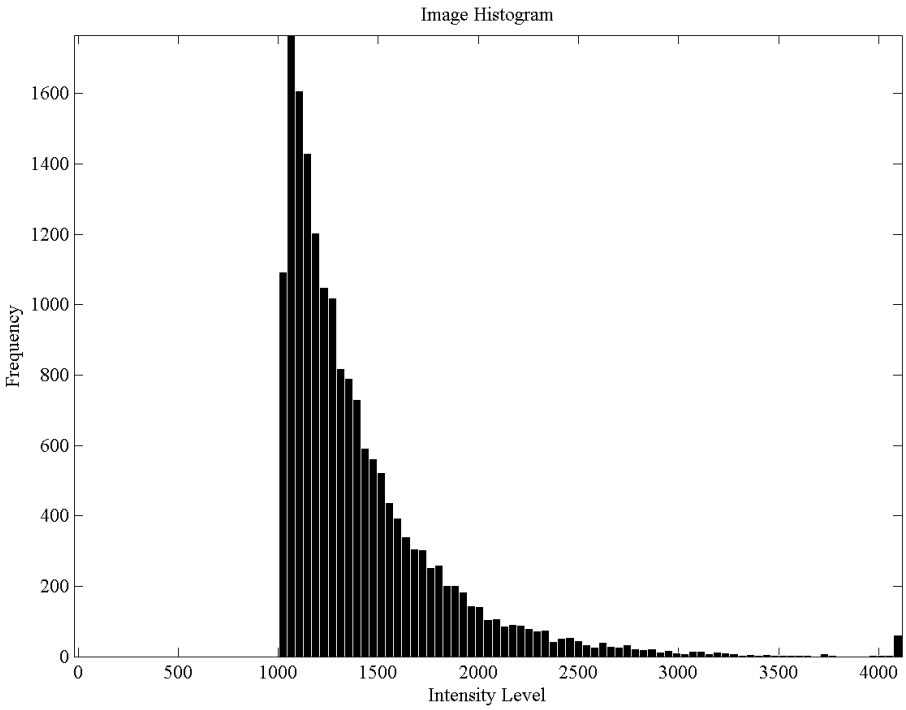

Figure C.1 Aptina – image histogram (result set 4)

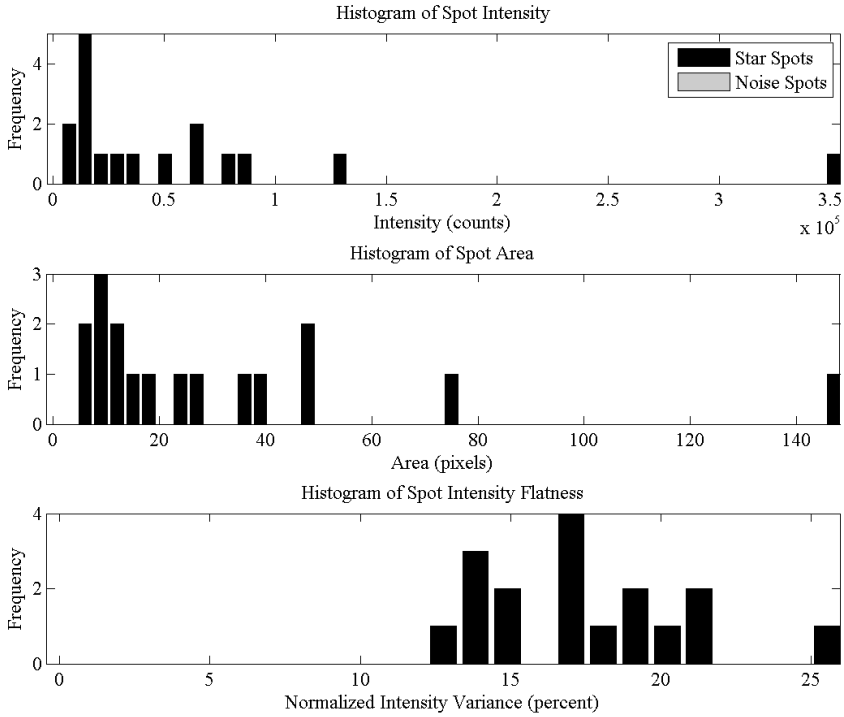

Figure C.2 Aptina – histograms of spot characteristics (top) spot intensity (middle) spot area (bottom) spot flatness factor (result set 4)

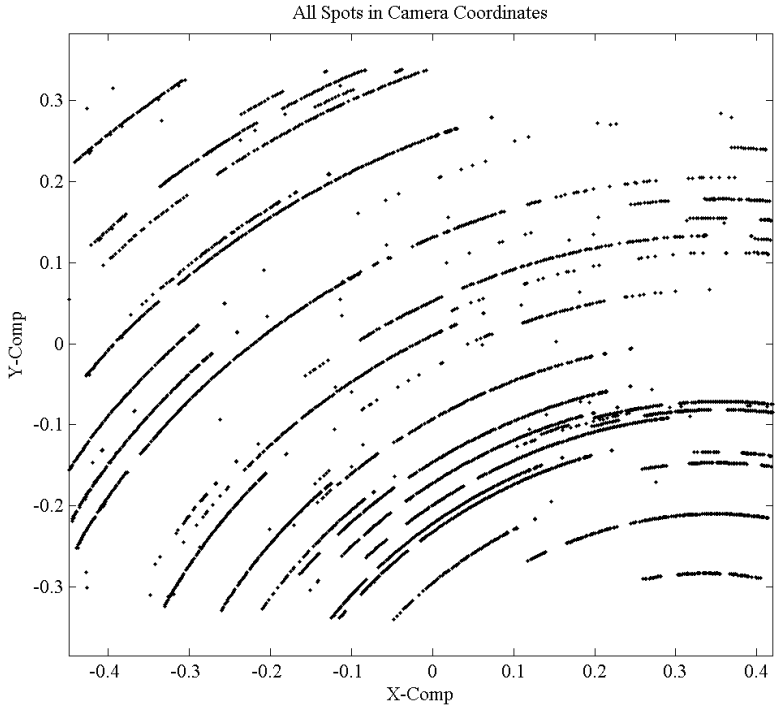

Figure C.3 Aptina – dataset spot slew (result set 4)

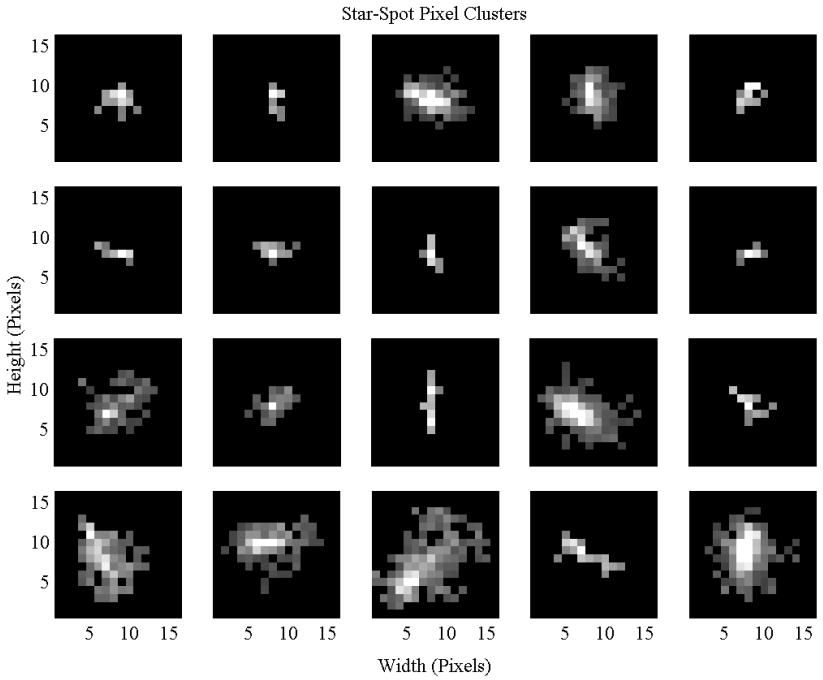

Figure C.4 Aptina – random starlight spot pixel clusters (result set 4)

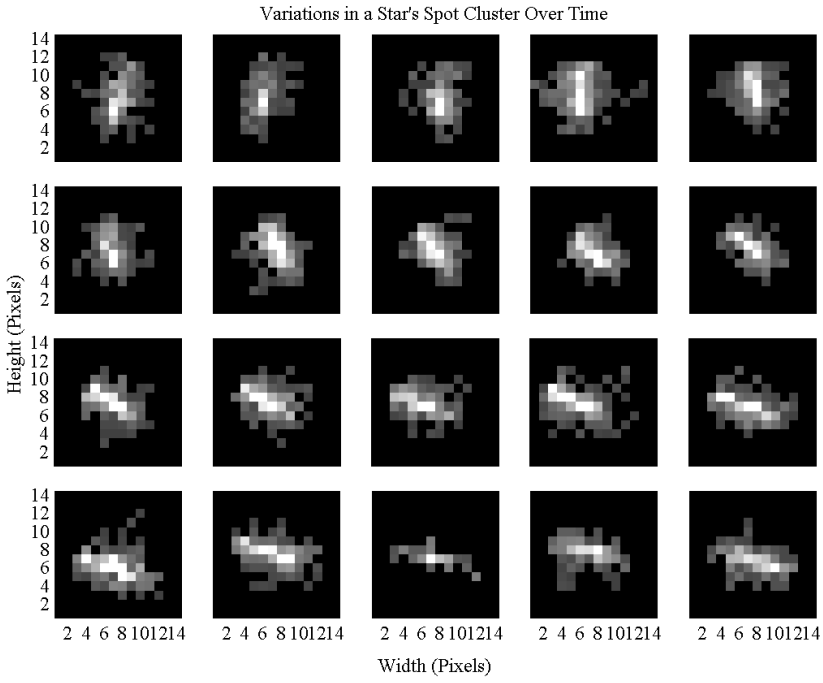

Figure C.5 Aptina – variation of a single starlight spot over time (result set 4)

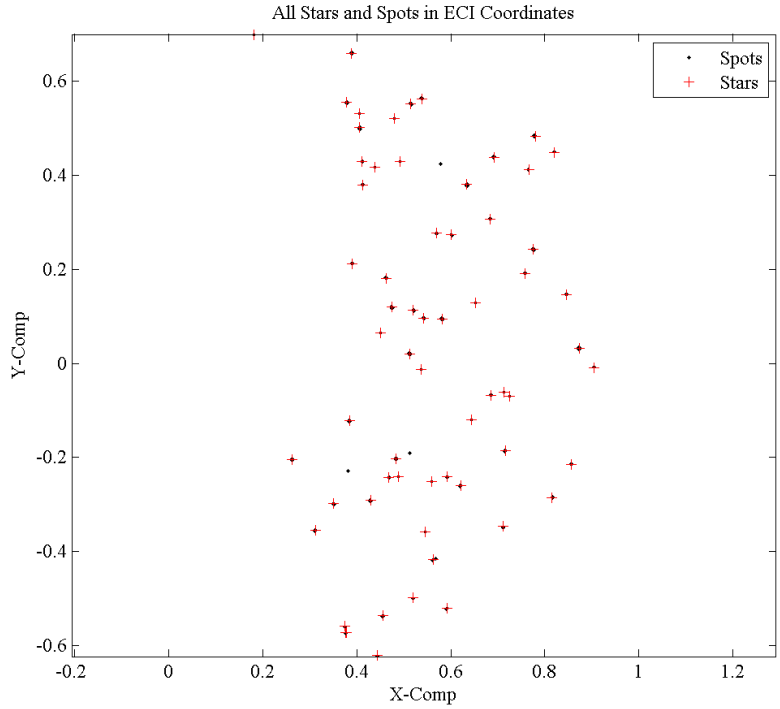

Figure C.6 Aptina – spots and stars mapped into the ECI reference frame (result set 4)

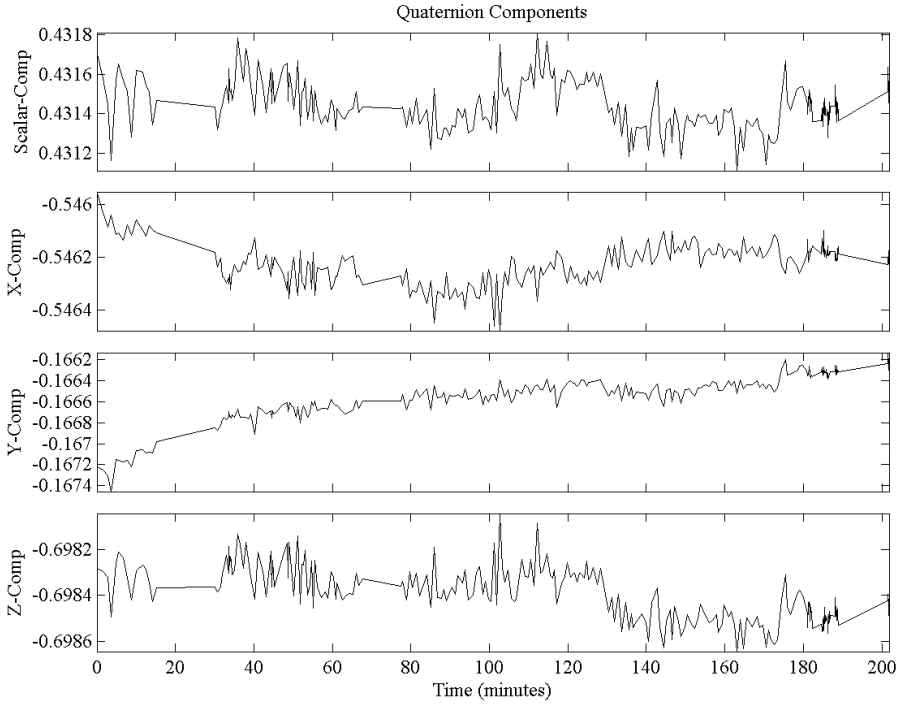

Figure C.7 Aptina – variation of the components of the camera mounting quaternion (camera to ENU) (result set 4)

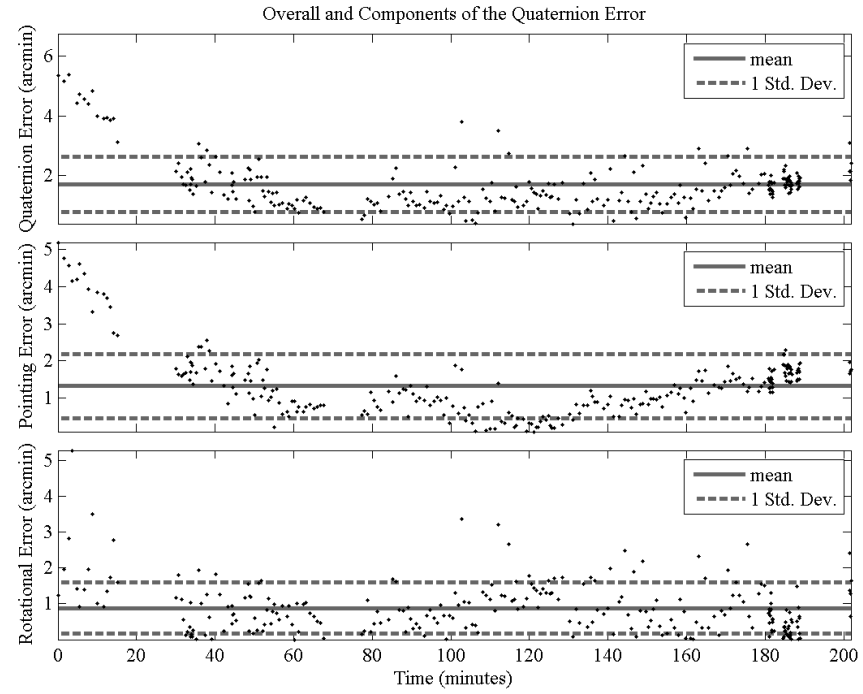

Figure C.8 Aptina – quaternion error and the pointing and rotational components (result set 4)

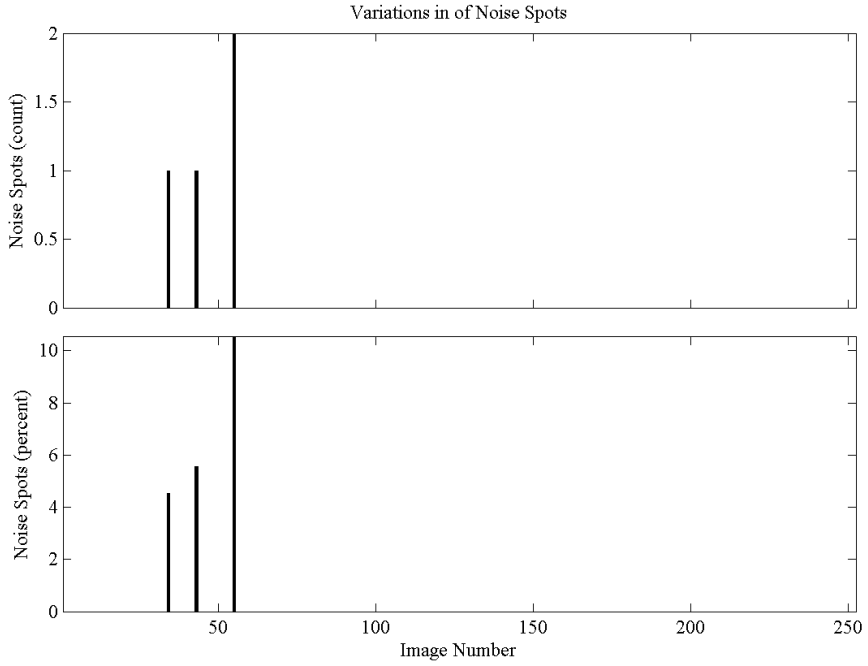

Figure C.9 Aptina – noise spots per image over the dataset (result set 4)

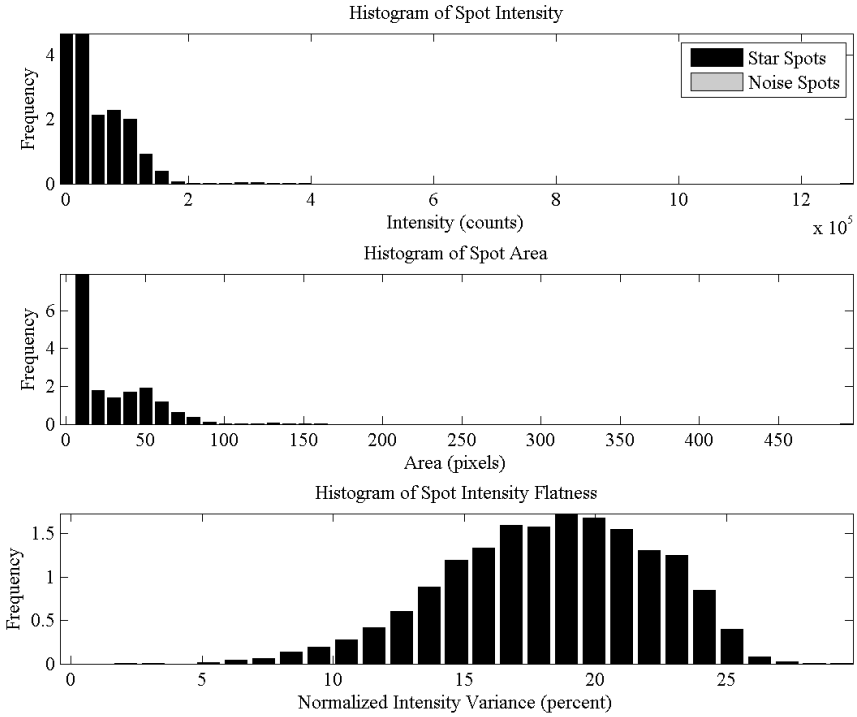

Figure C.10 Aptina – average histograms over the dataset (result set 4)

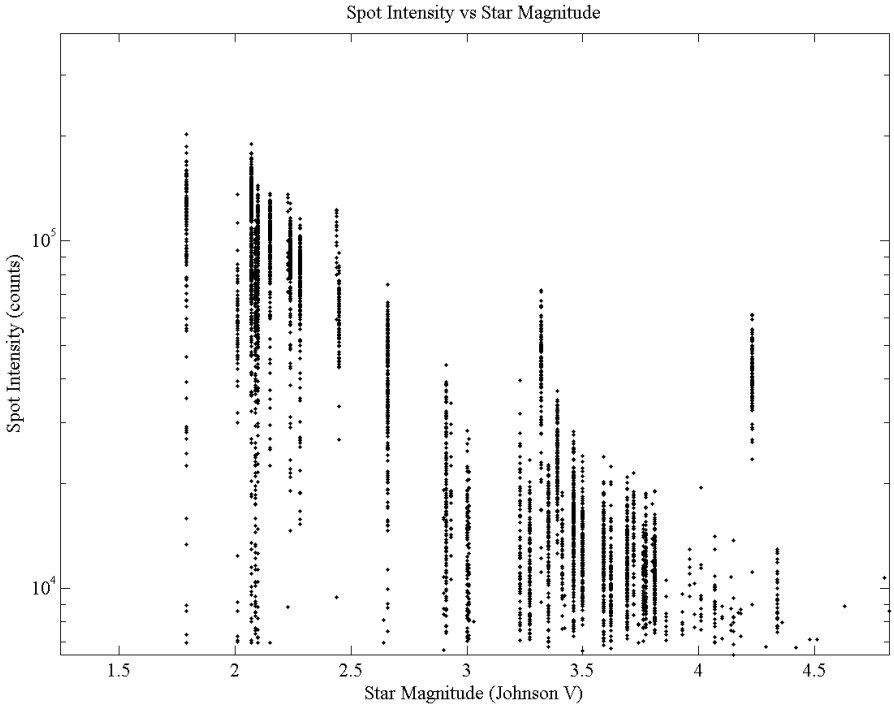

Figure C.11 Aptina – spot intensity compared to its identified star's magnitude (result set 4)

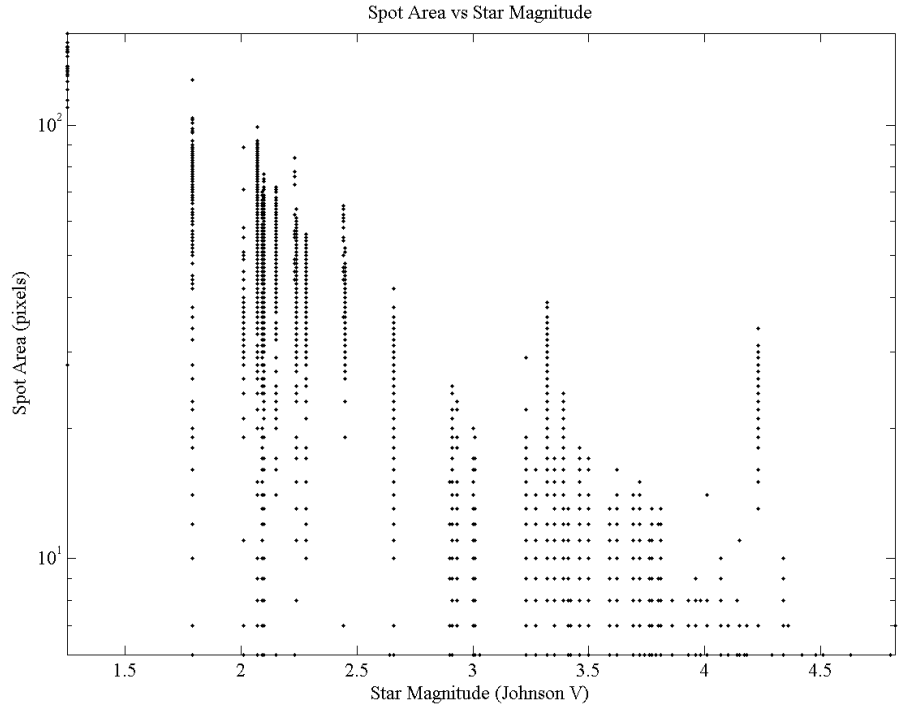

Figure C.12 Aptina – spot area compared to its identified star's magnitude (result set 4)

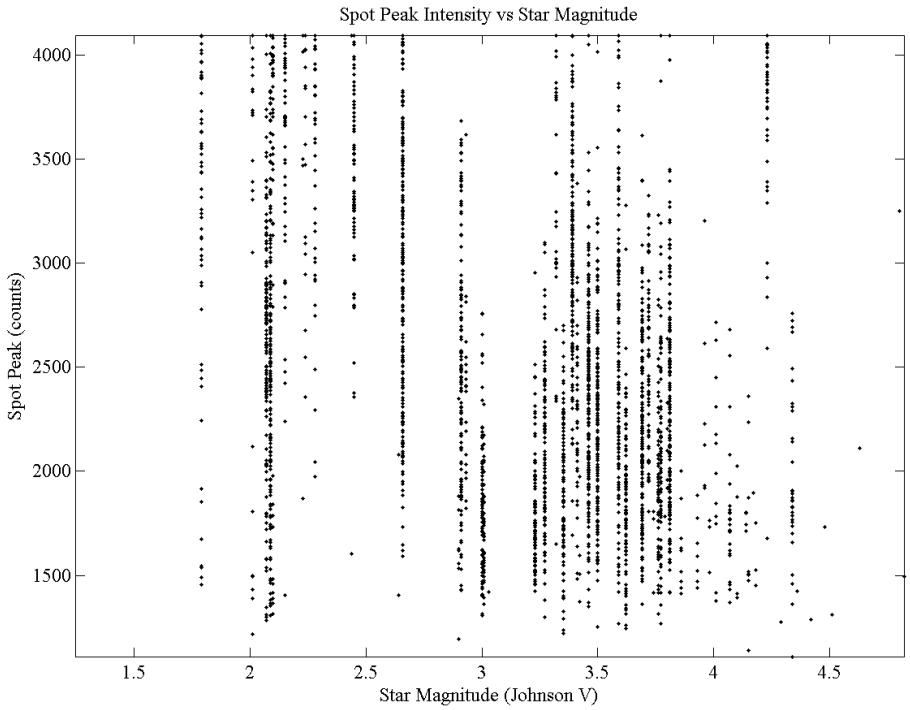

Figure C.13 Aptina – spot peak intensity compared to its identified star's magnitude (result set 4)

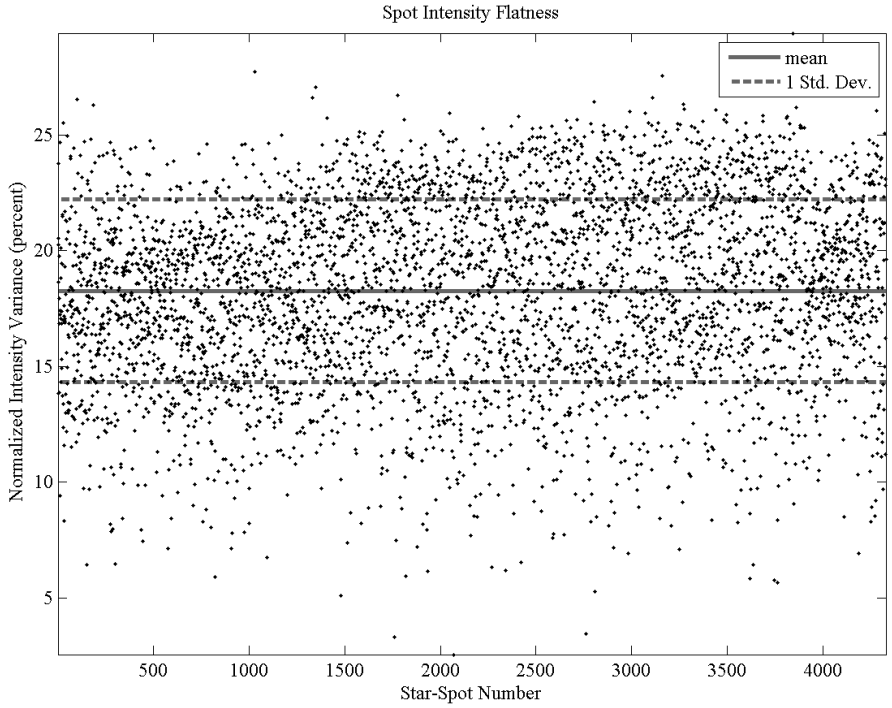

Figure C.14 Aptina – culmination of all dataset spot flatness factors (result set 4)

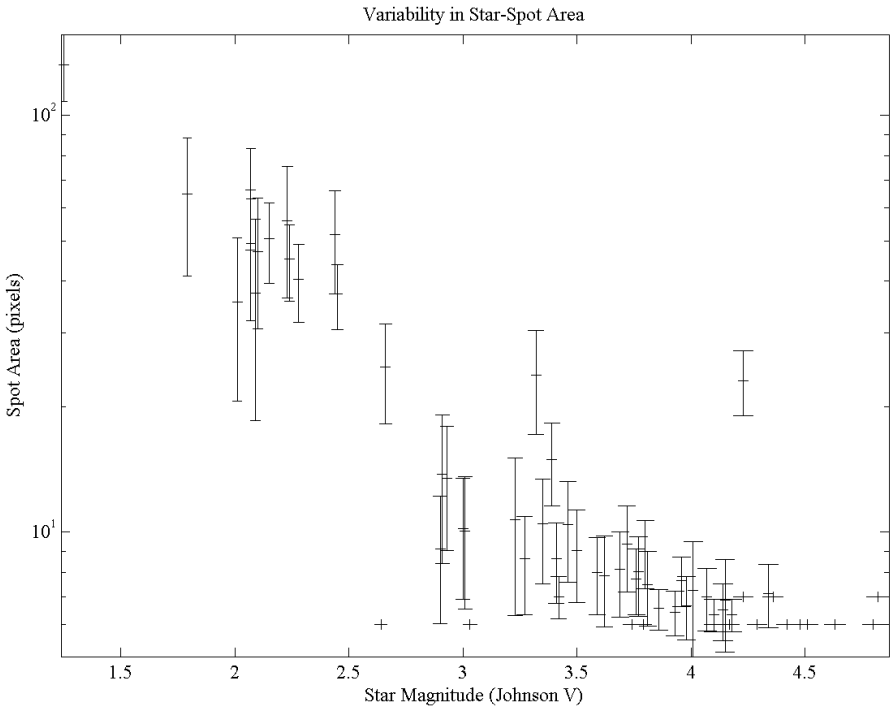

Figure C.15 Aptina – variability in starlight spot areas (result set 4)

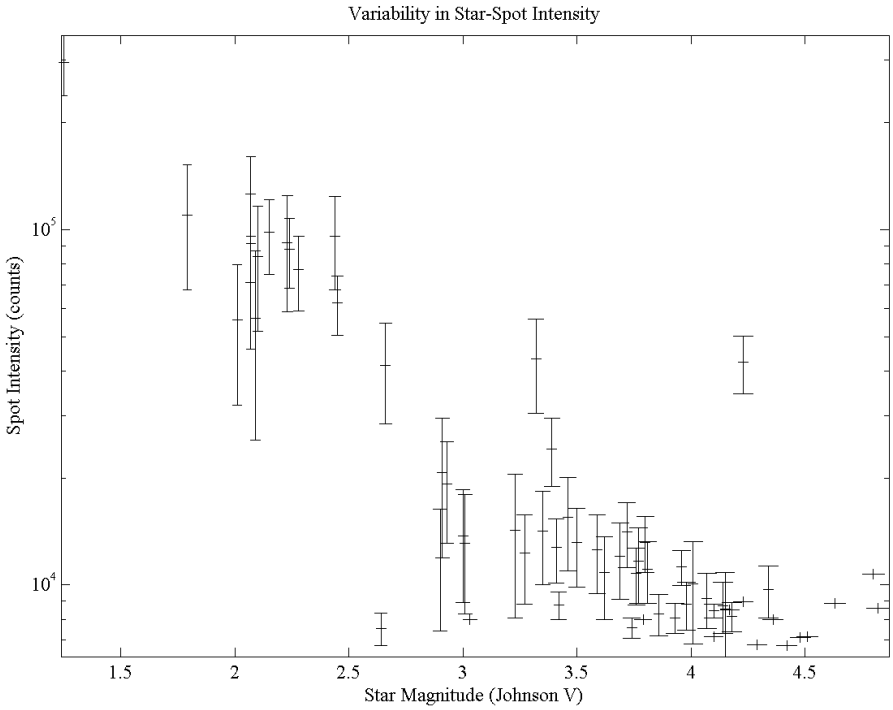

Figure C.16 Aptina – variability in starlight spot intensities (result set 4)

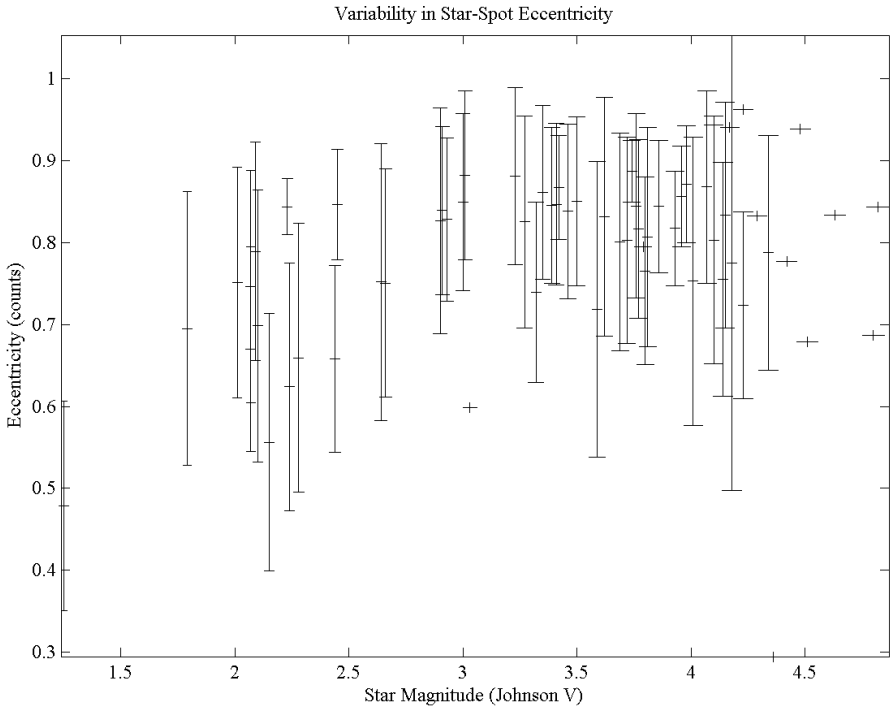

Figure C.17 Aptina – variability in starlight spot eccentricities (result set 4)

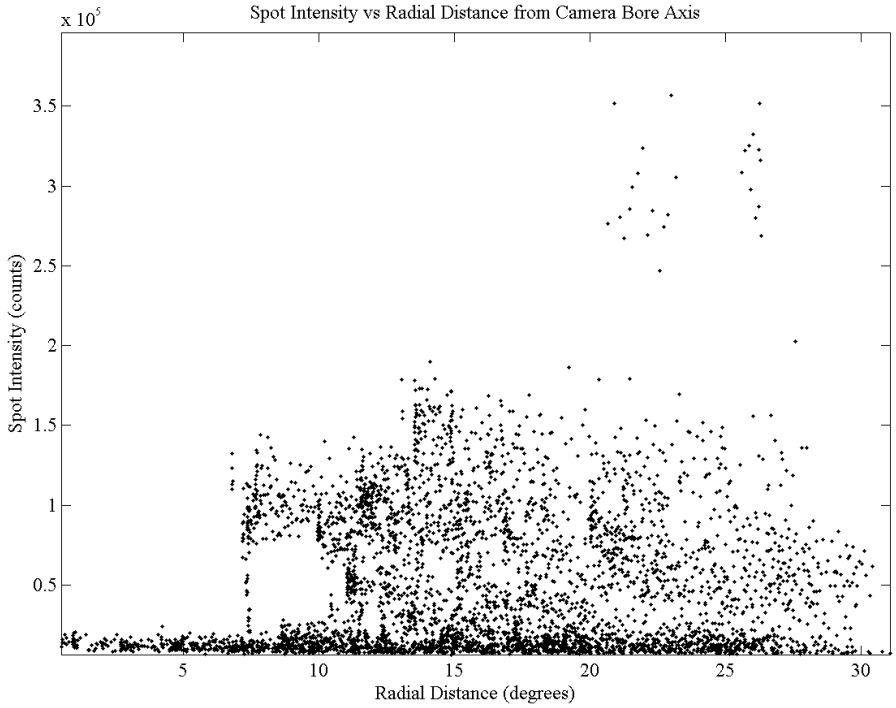

Figure C.18 Aptina – spot intensities compared to their radial distance away from the camera bore axis (result set 4)

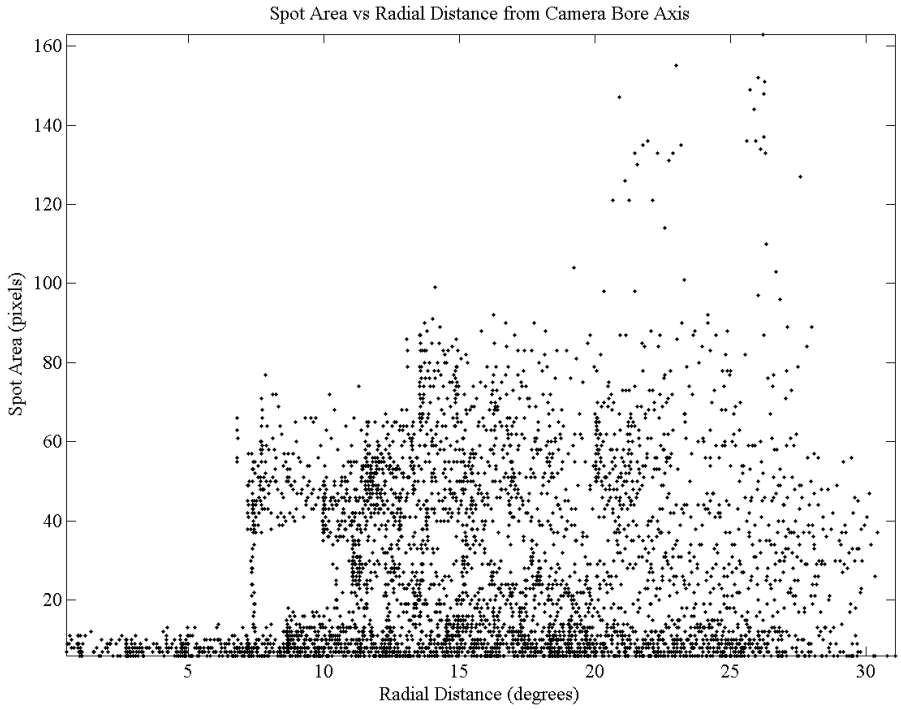

Figure C.19 Aptina – spot area compared to their radial distance away from the camera bore axis (result set 4)

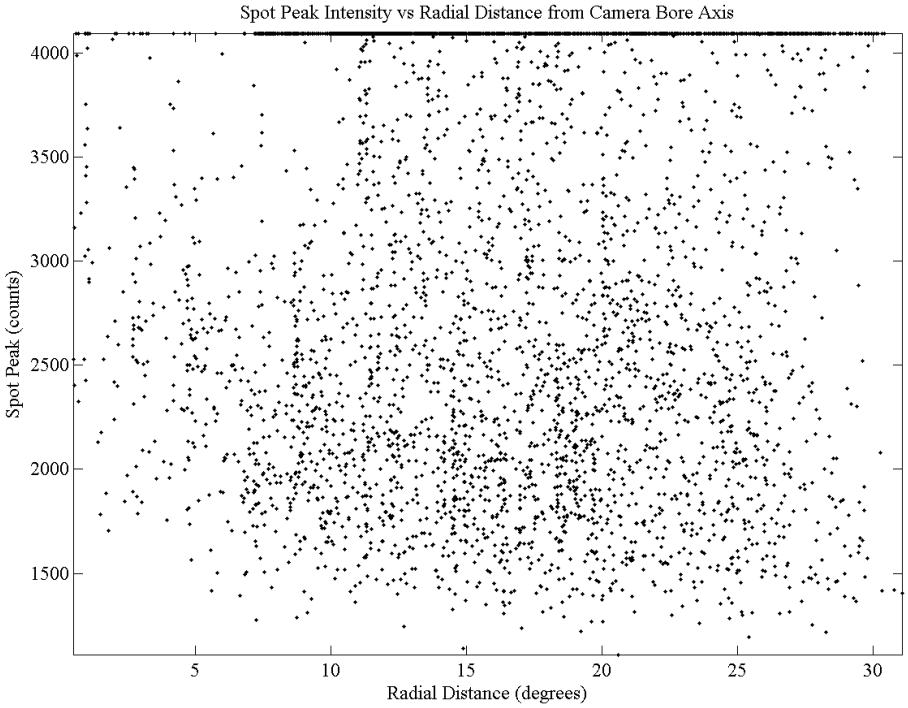

Figure C.20 Aptina – spot peak intensity compared to their radial distance away from the camera bore axis (result set 4)

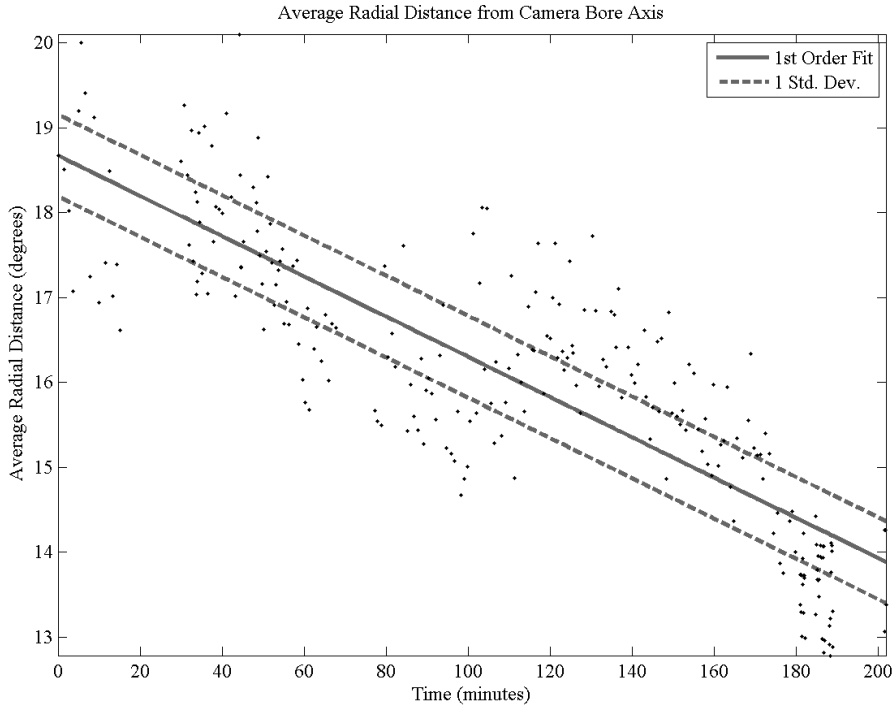

Figure C.21 Aptina – average spot radial distance away from the camera bore axis per image (result set 4)

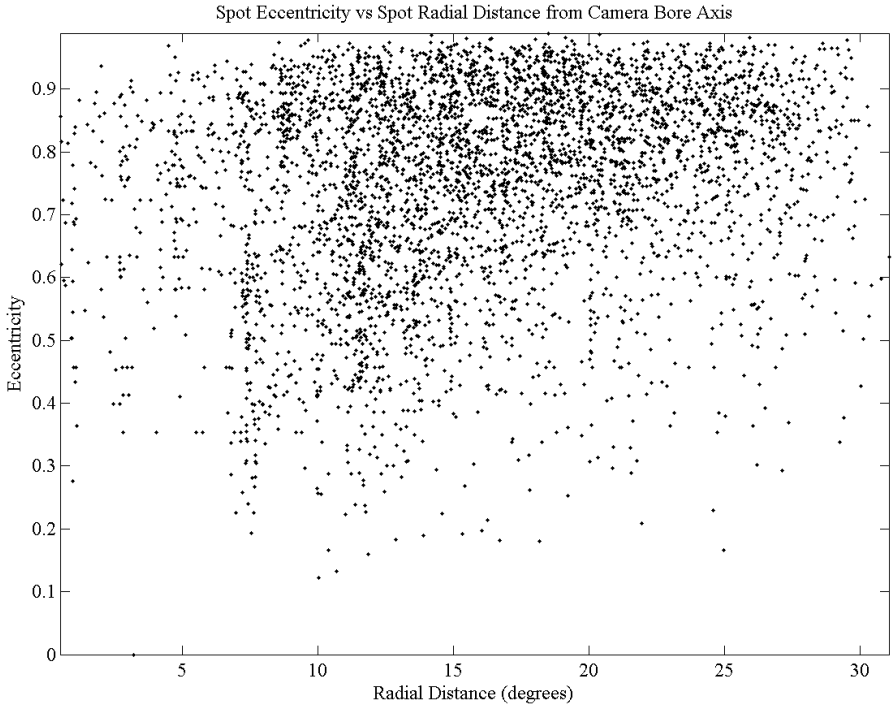

Figure C.22 Aptina – spot eccentricity compared to their radial distance away from the camera bore axis (result set 4)

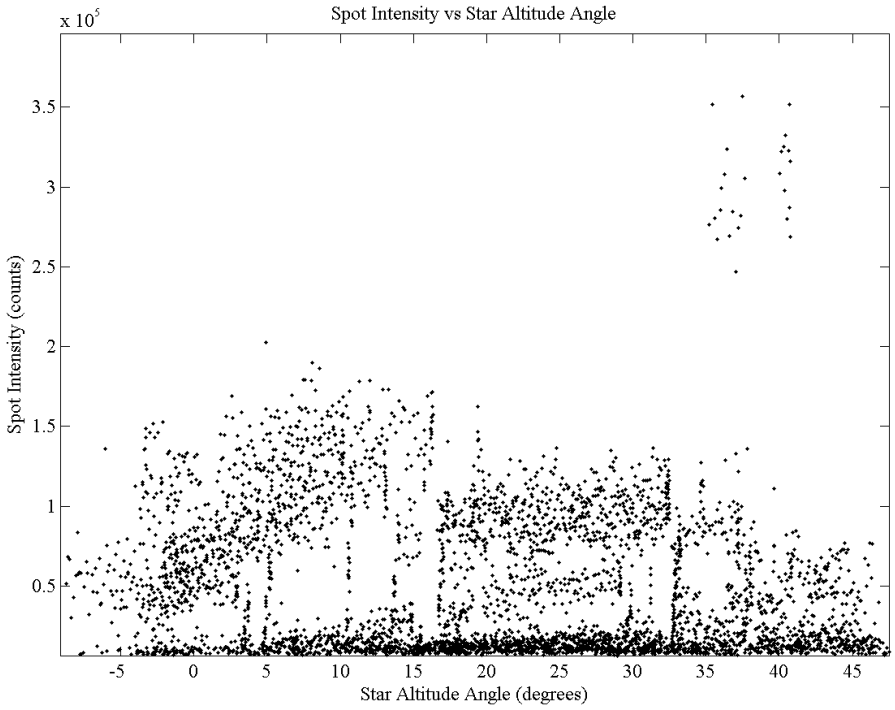

Figure C.23 Aptina – spot intensity compared to the star's altitude angle (result set 4)

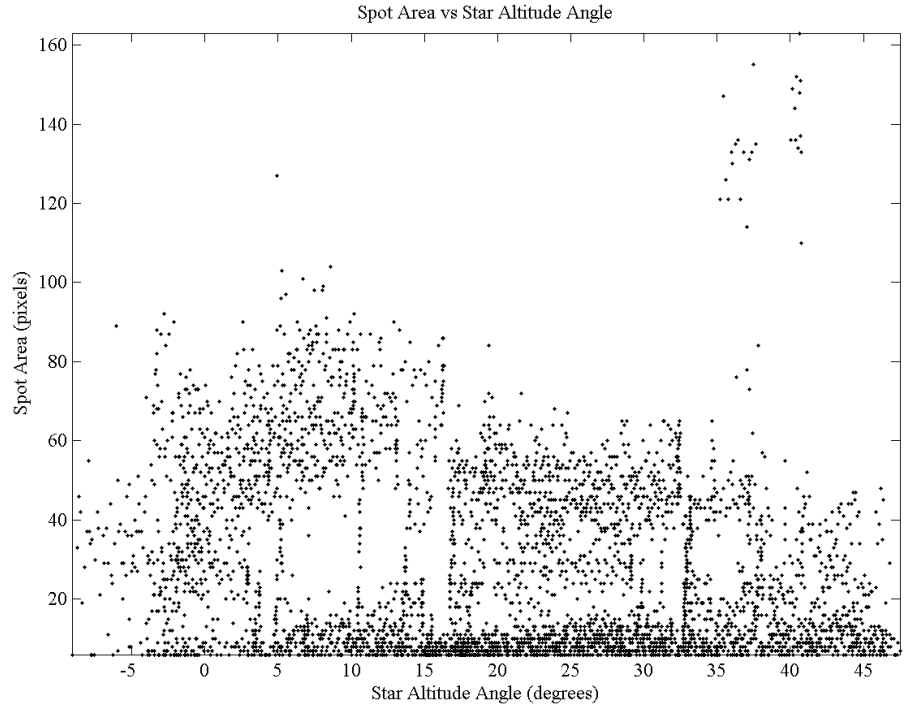

Figure C.24 Aptina – spot area compared to the star's altitude angle (result set 4)

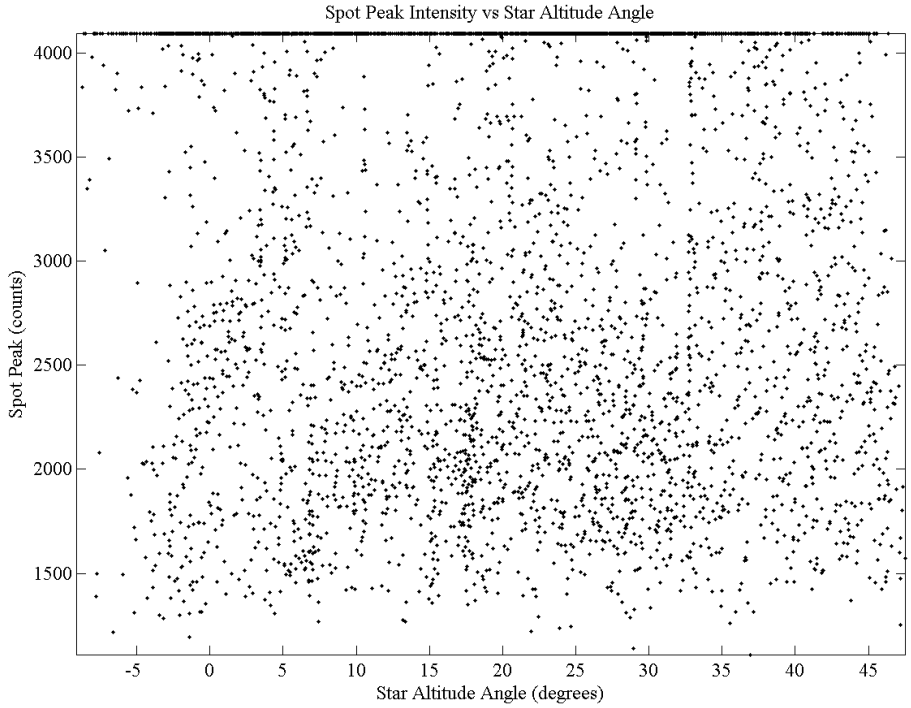

Figure C.25 Aptina – spot peak intensity compared to the star's altitude angle (result set 4)

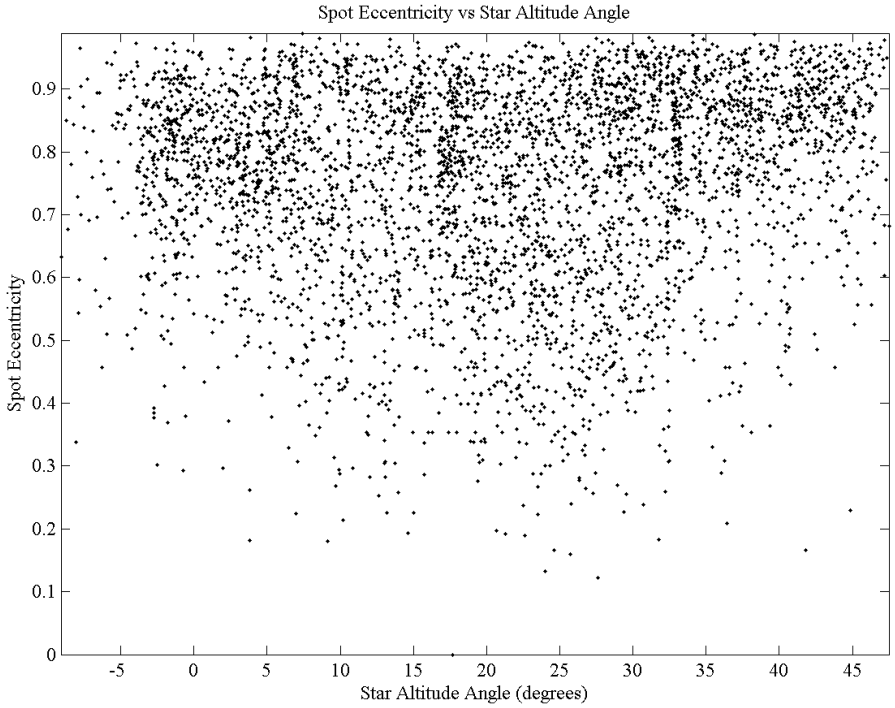

Figure C.26 Aptina – spot eccentricity compared to the star's altitude angle (result set 4)

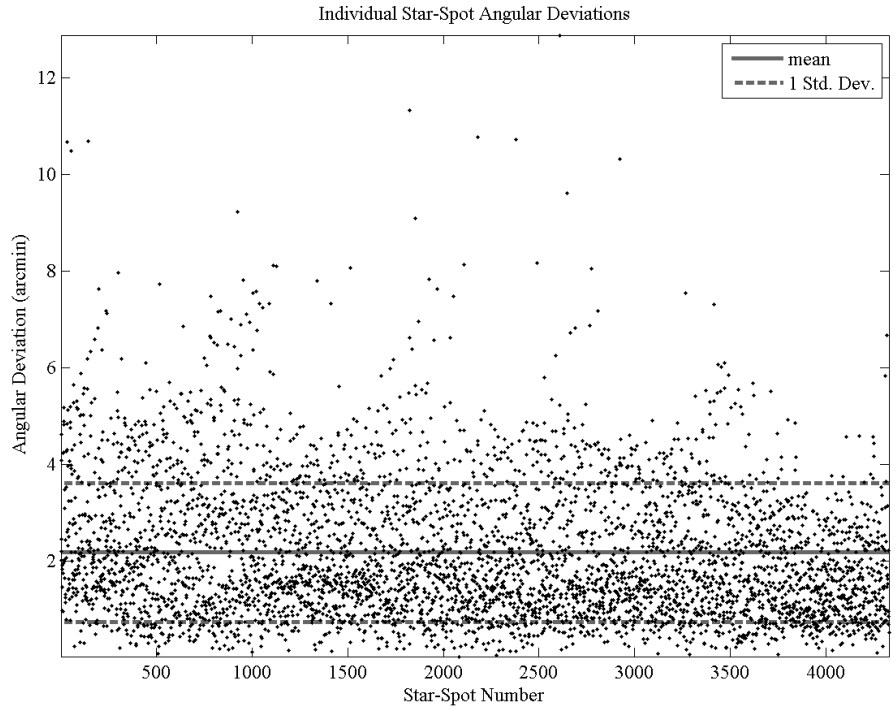

Figure C.27 Aptina – individual star to spot angular deviations (result set 4)

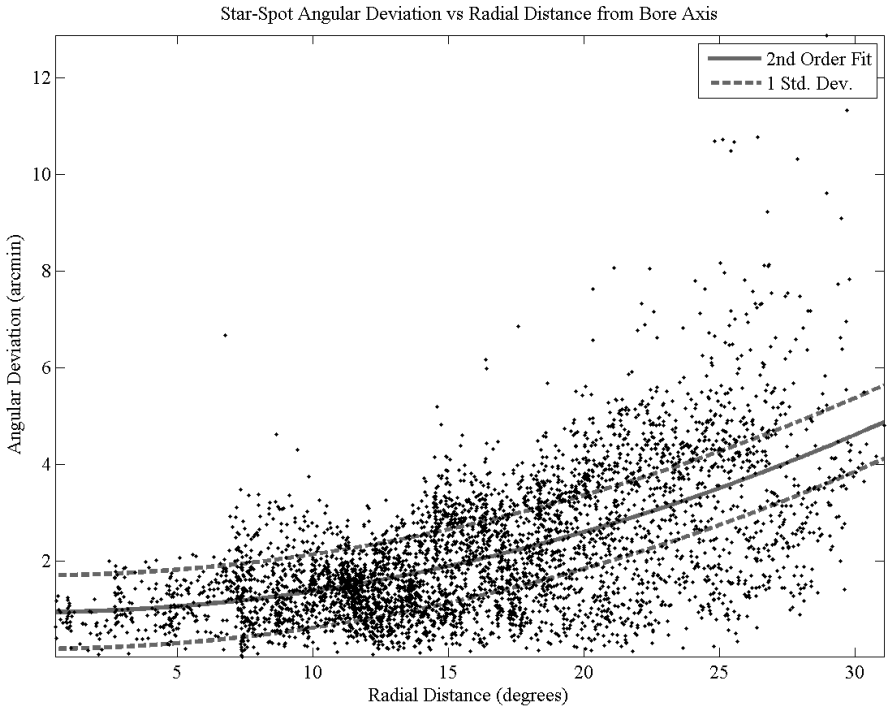

Figure C.28 Aptina – star to spot angular deviation compared to the spot's radial distance away from the camera bore axis (result set 4)

II. Aptina – Result Set 6

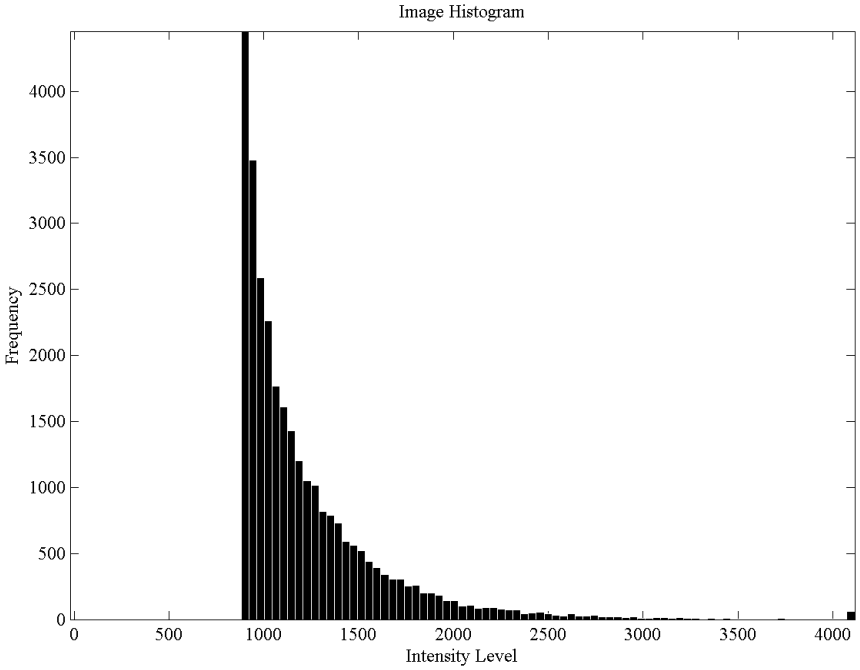

Figure C.29 Aptina – image histogram (result set 4)

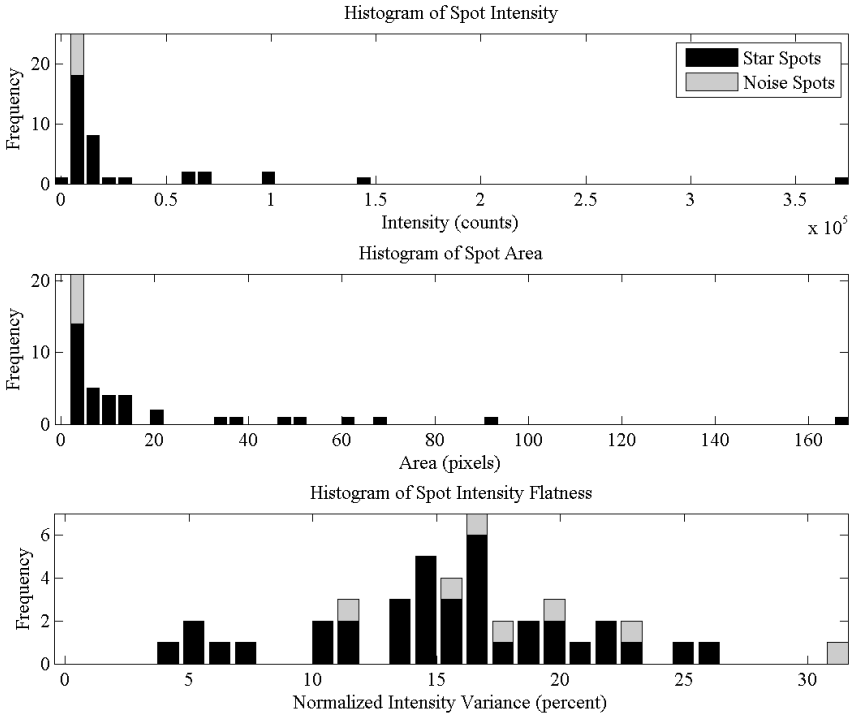

Figure C.30 Aptina – histograms of spot characteristics (top) spot intensity (middle) spot area (bottom) spot flatness factor (result set 4)

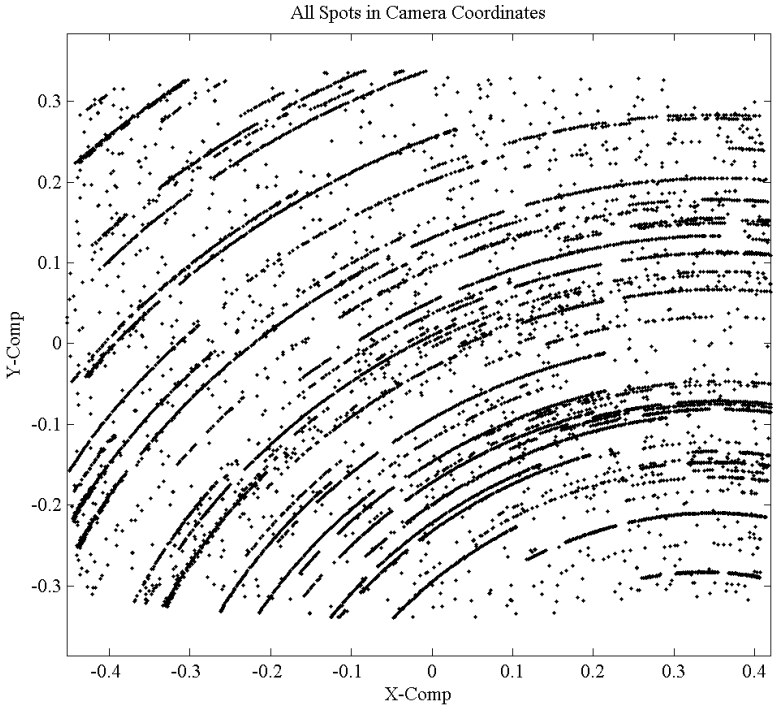

Figure C.31 Aptina – dataset spot slew (result set 4)

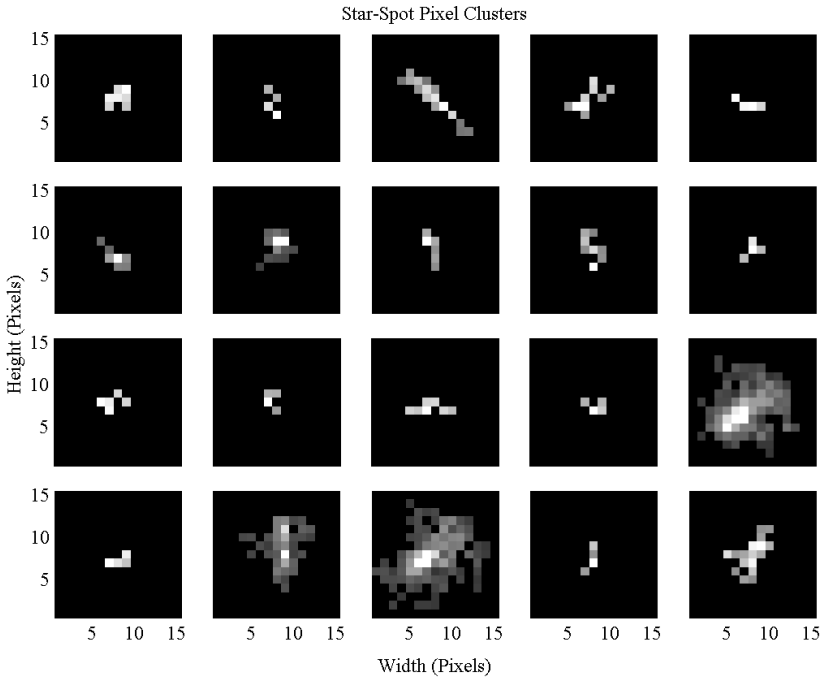

Figure C.32 Aptina – random starlight spot pixel clusters (result set 4)

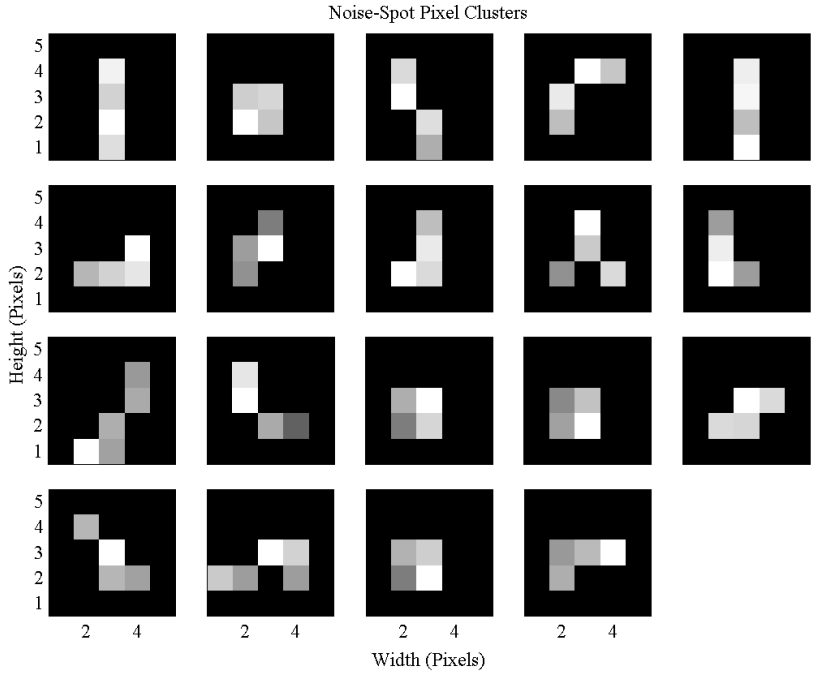

Figure C.33 Aptina – variation of a single starlight spot over time (result set 4)

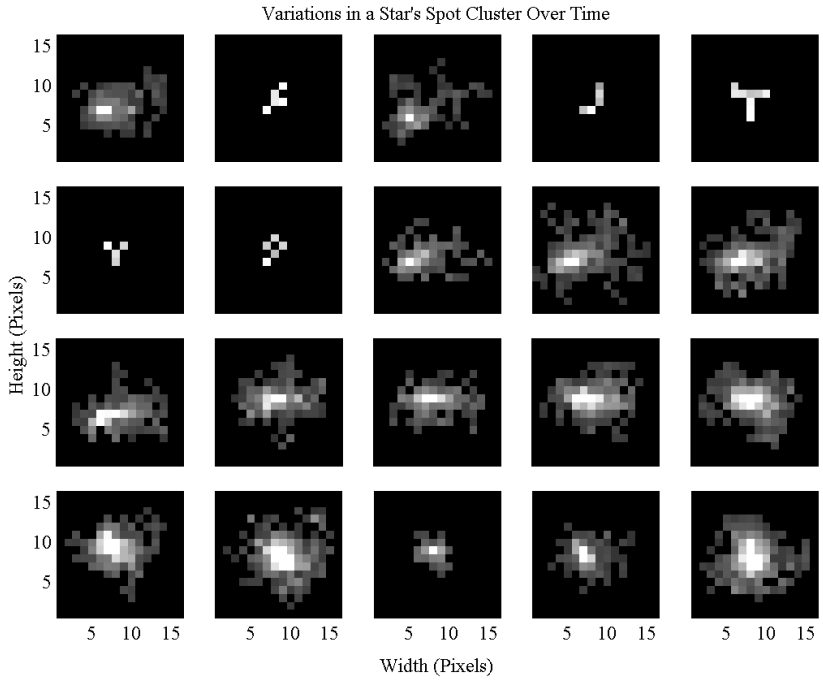

Figure C.34 Aptina – variation of a single starlight spot over time (result set 4)

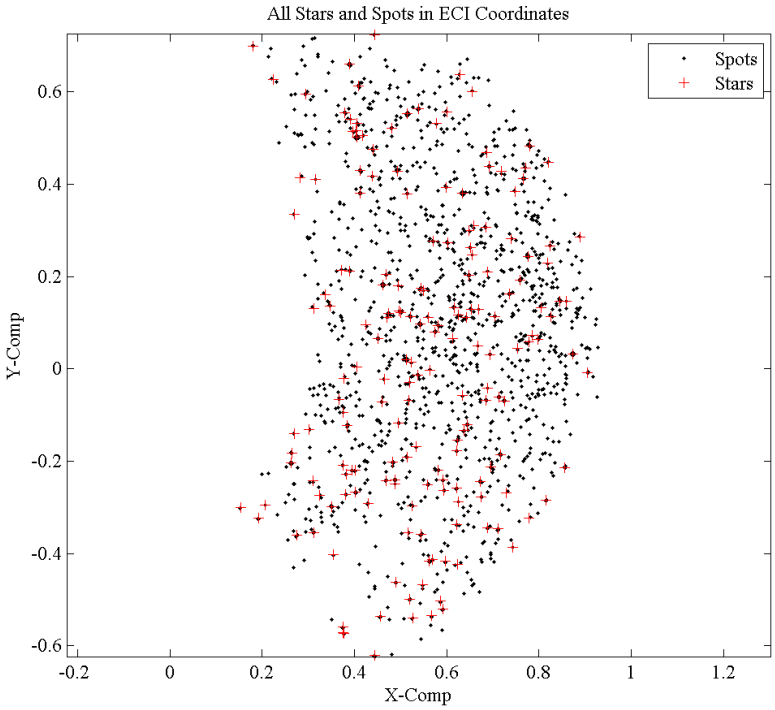

Figure C.35 Aptina – spots and stars mapped into the ECI reference frame (result set 4)

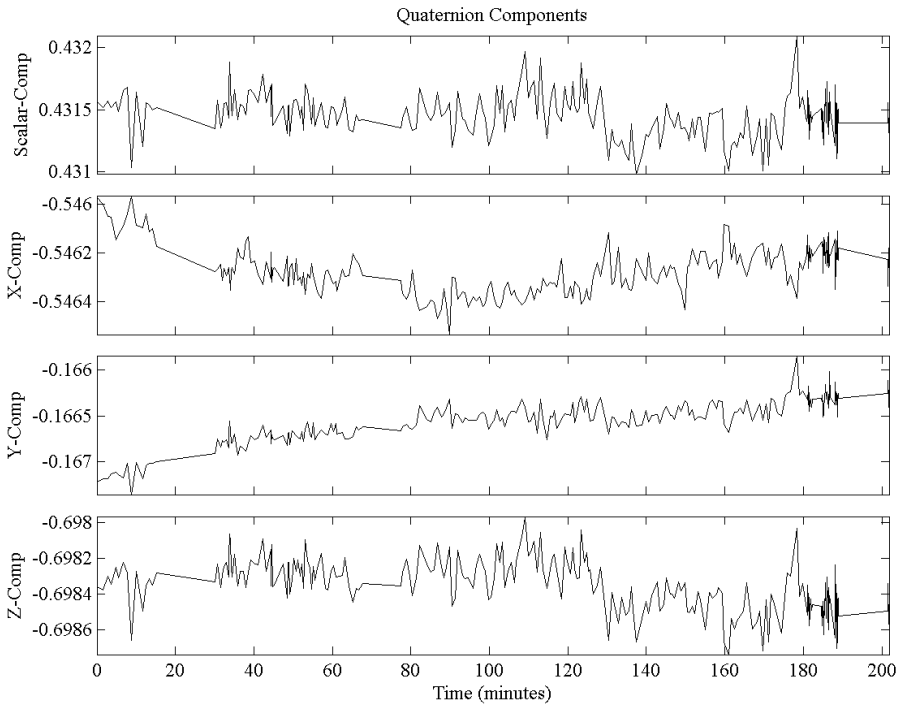

Figure C.36 Aptina – variation of the components of the camera mounting quaternion (camera to ENU) (result set 4)

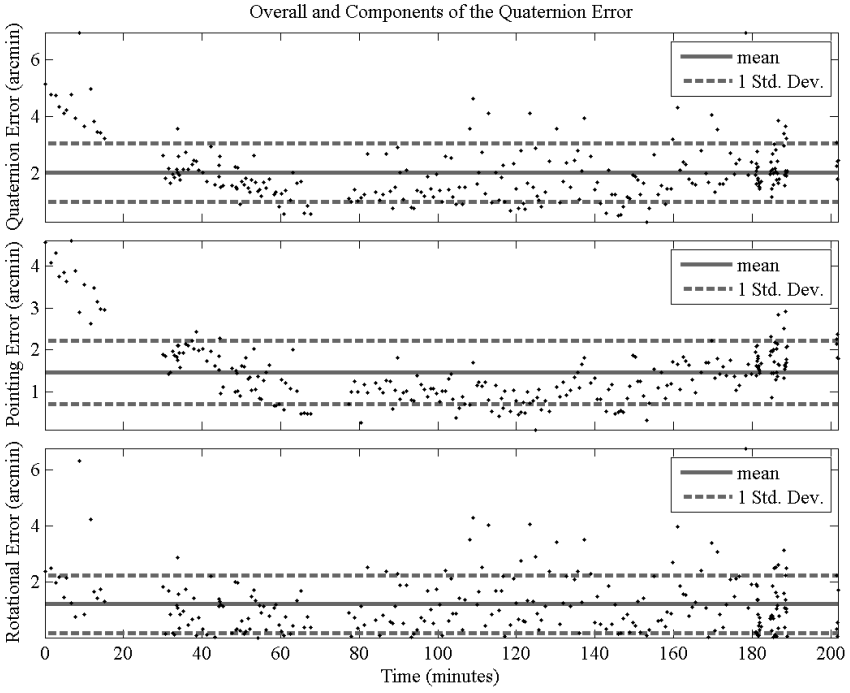

Figure C.37 Aptina – quaternion error and the pointing and rotational components (result set 4)

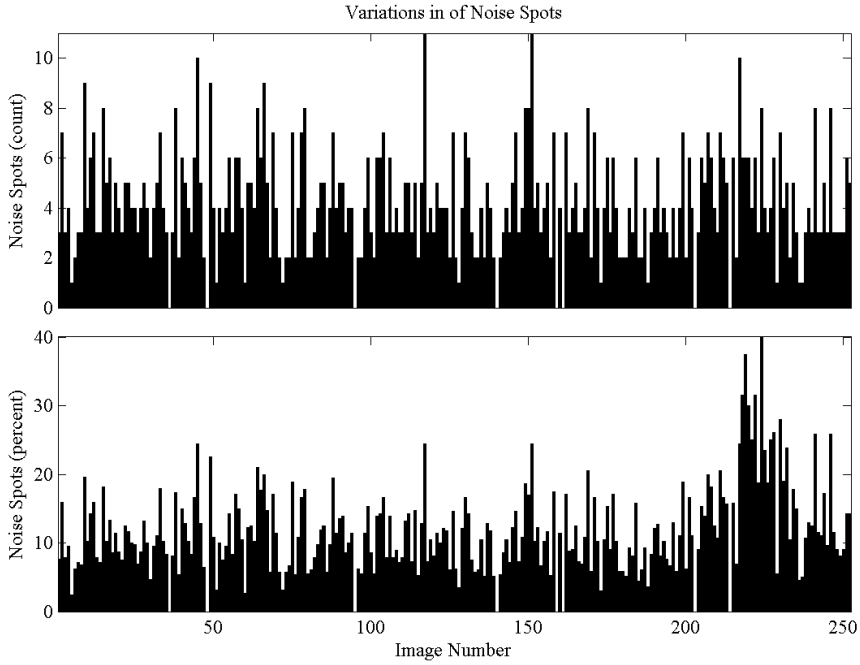

Figure C.38 Aptina – noise spots per image over the dataset (result set 4)

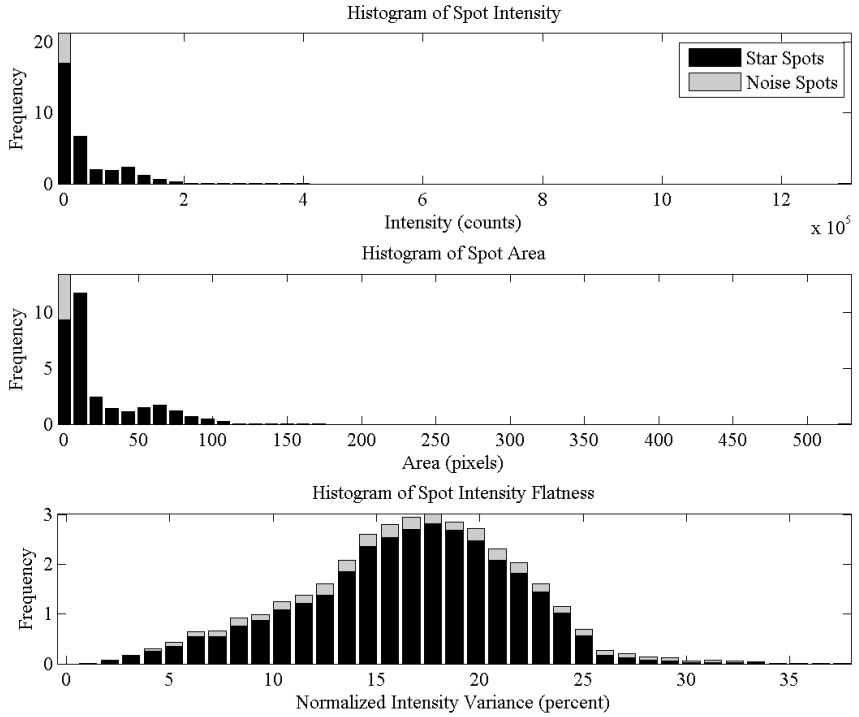

Figure C.39 Aptina – average histograms over the dataset (result set 4)

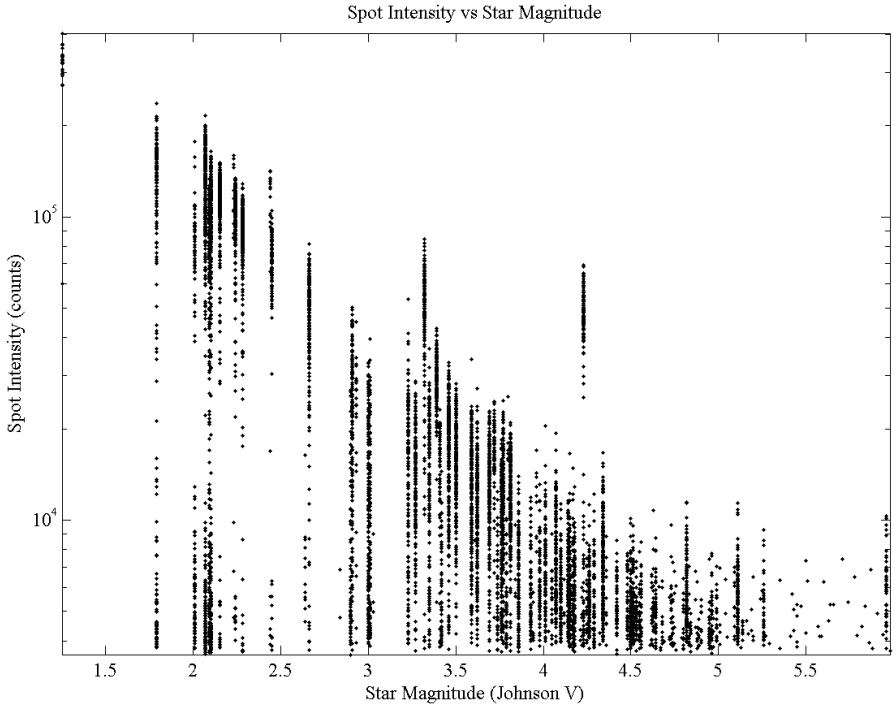

Figure C.40 Aptina – spot intensity compared to its identified star's magnitude (result set 4)

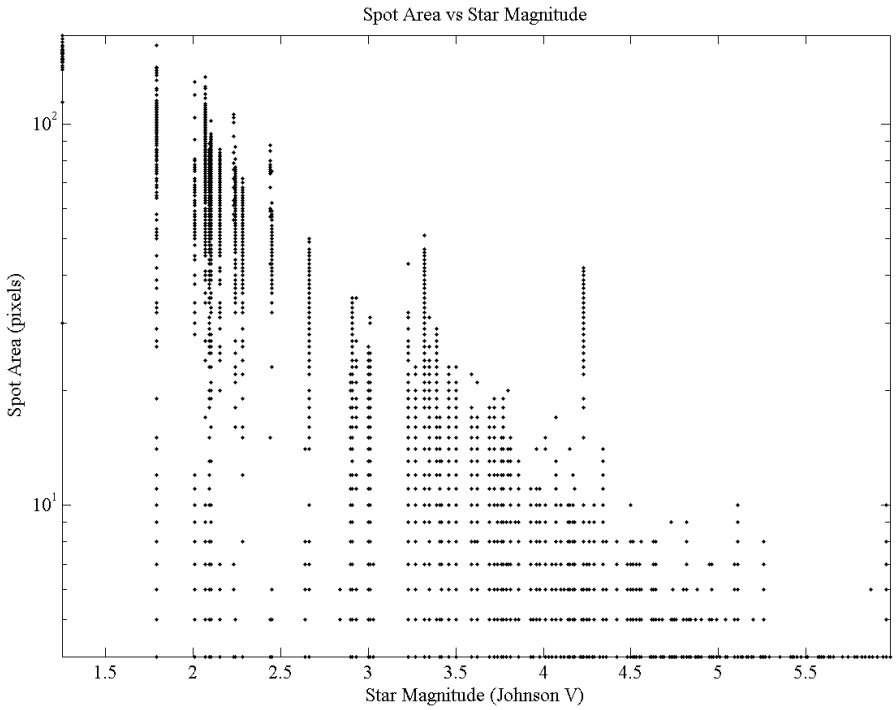

Figure C.41 Aptina – spot area compared to its identified star's magnitude (result set 4)

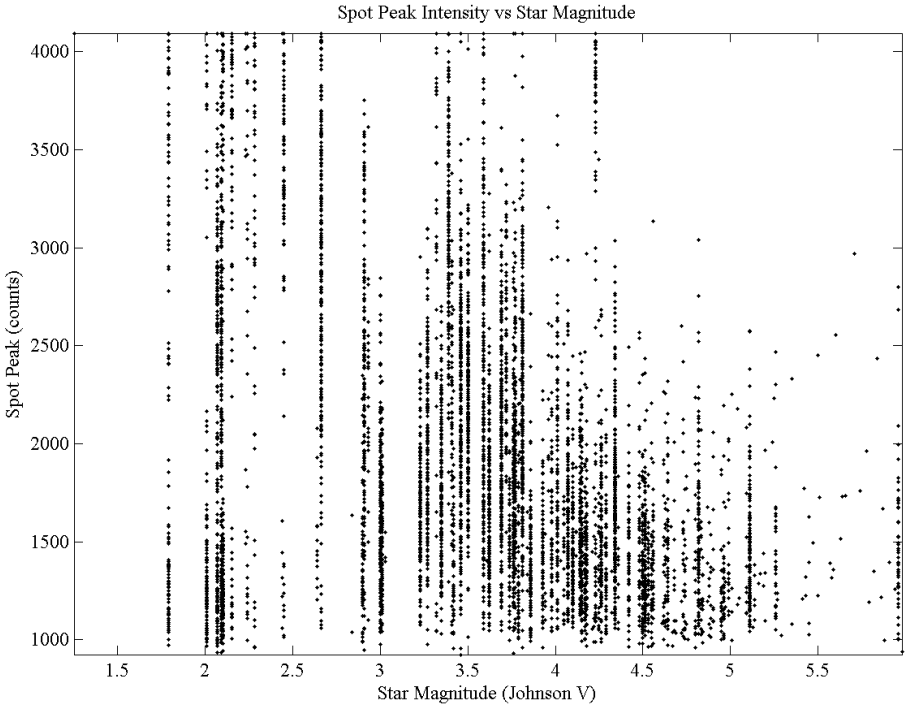

Figure C.42 Aptina – spot peak intensity compared to its identified star's magnitude (result set 4)

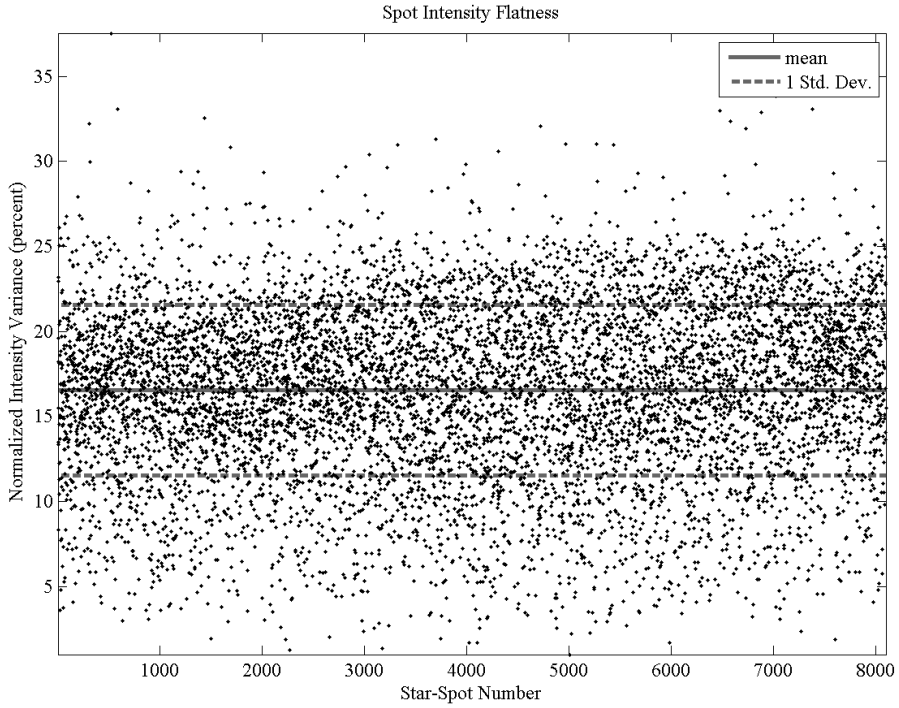

Figure C.43 Aptina – culmination of all dataset spot flatness factors (result set 4)

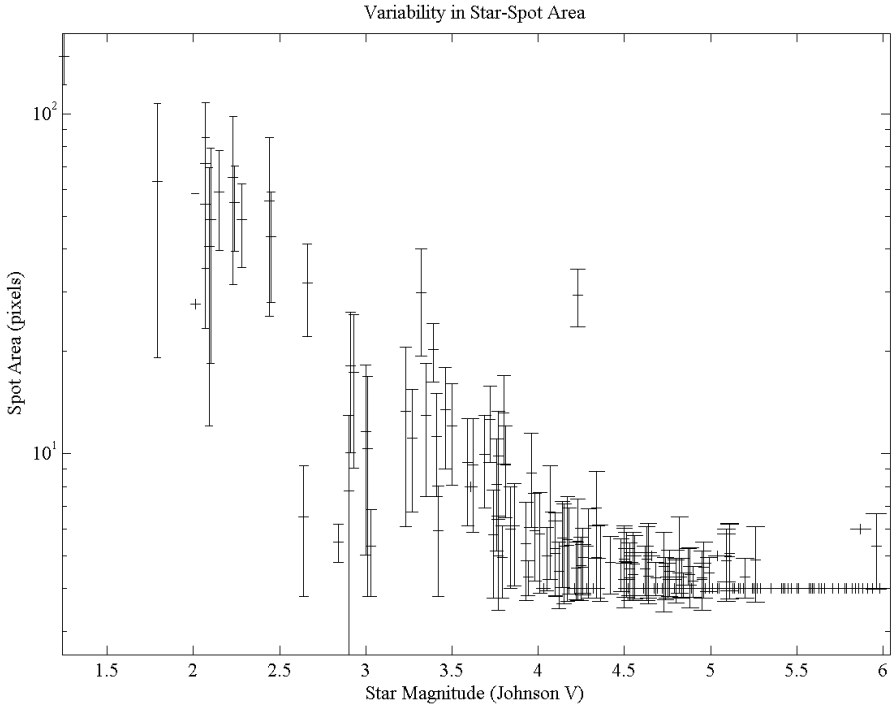

Figure C.44 Aptina – variability in starlight spot areas (result set 4)

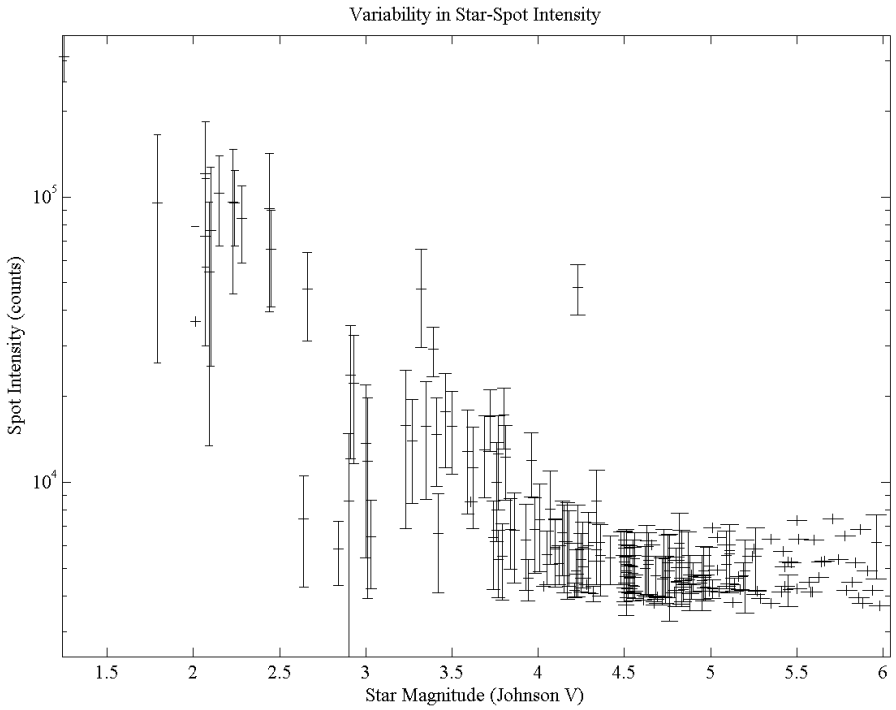

Figure C.45 Aptina – variability in starlight spot intensities (result set 4)

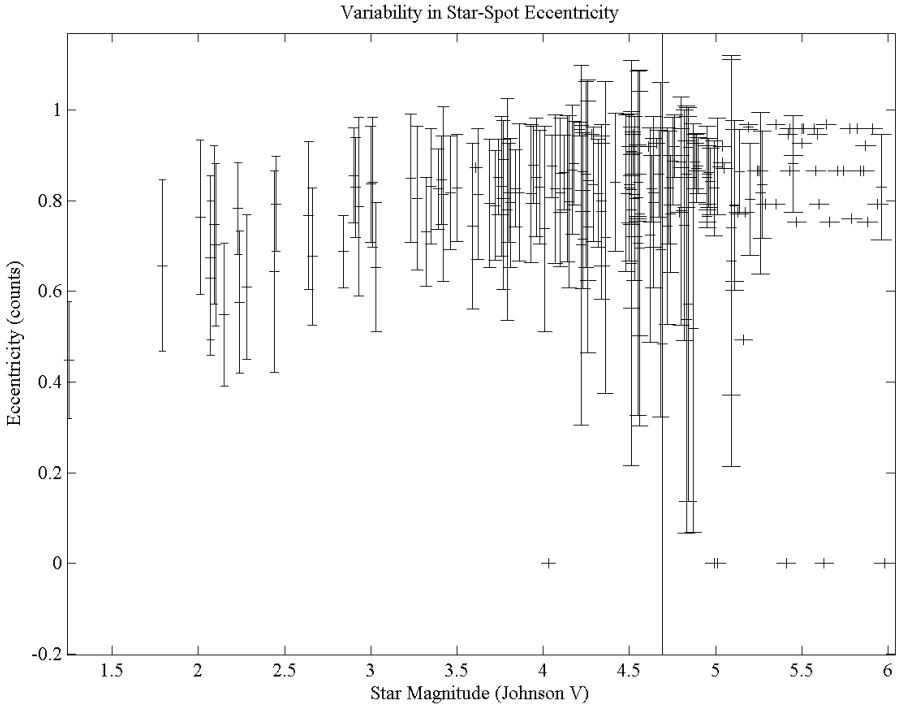

Figure C.46 Aptina – variability in starlight spot eccentricities (result set 4)

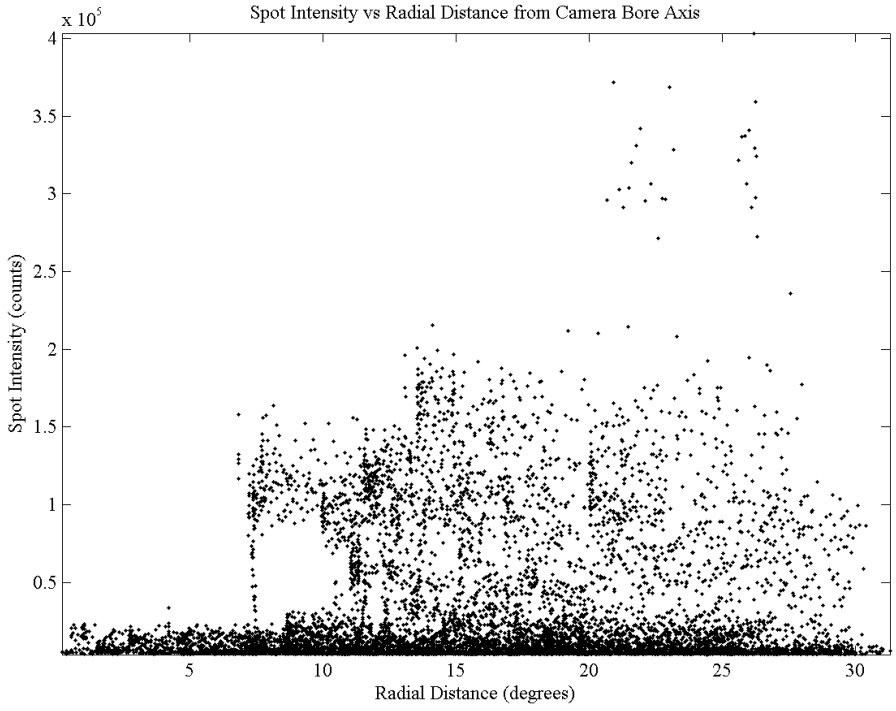

Figure C.47 Aptina – spot intensities compared to their radial distance away from the camera bore axis (result set 4)

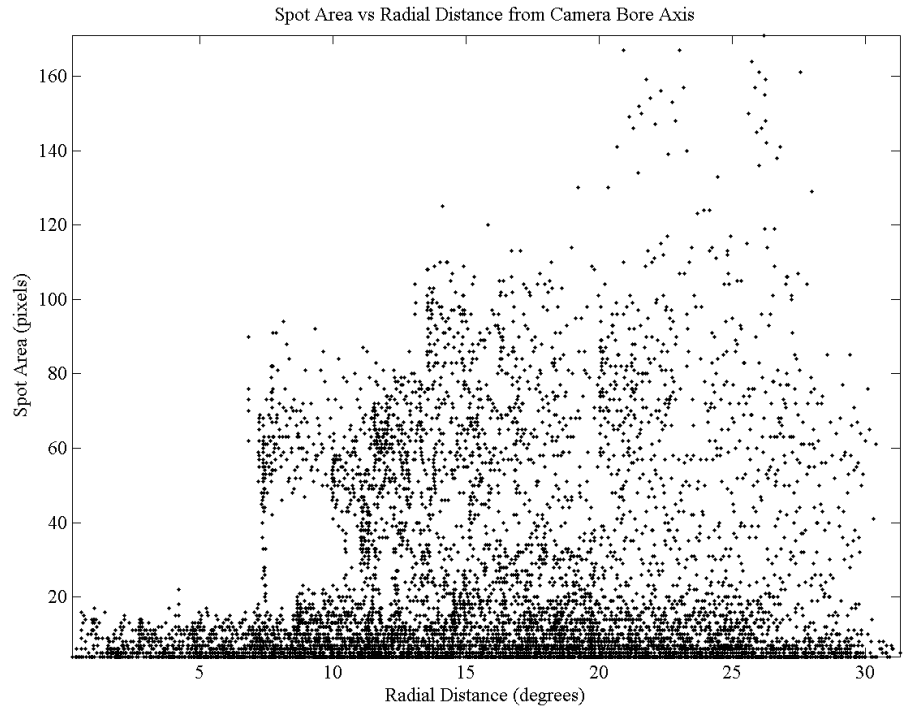

Figure C.48 Aptina – spot area compared to their radial distance away from the camera bore axis (result set 4)

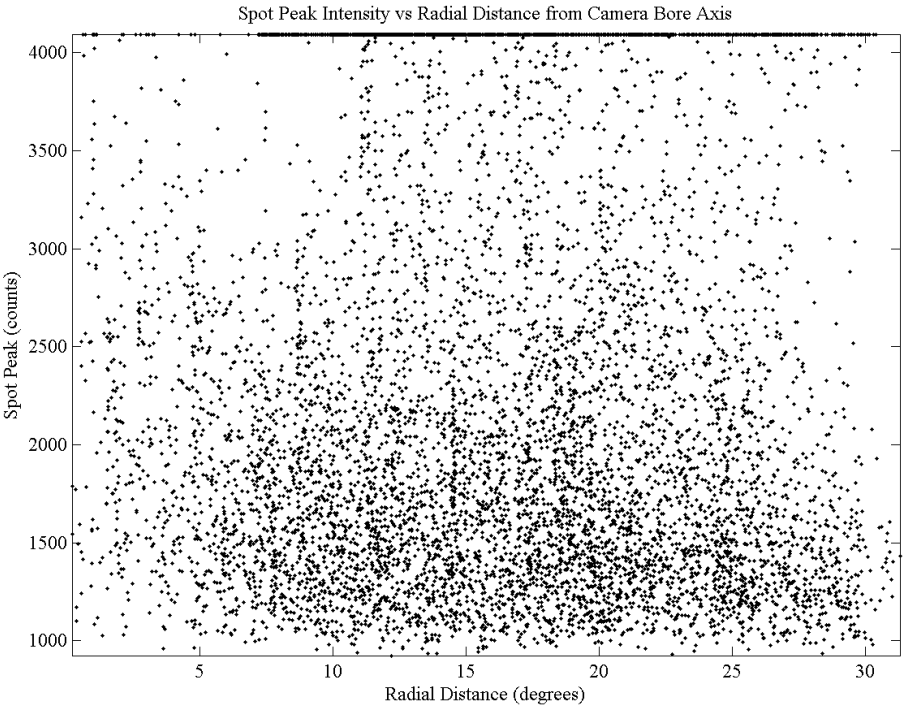

Figure C.49 Aptina – spot peak intensity compared to their radial distance away from the camera bore axis (result set 4)

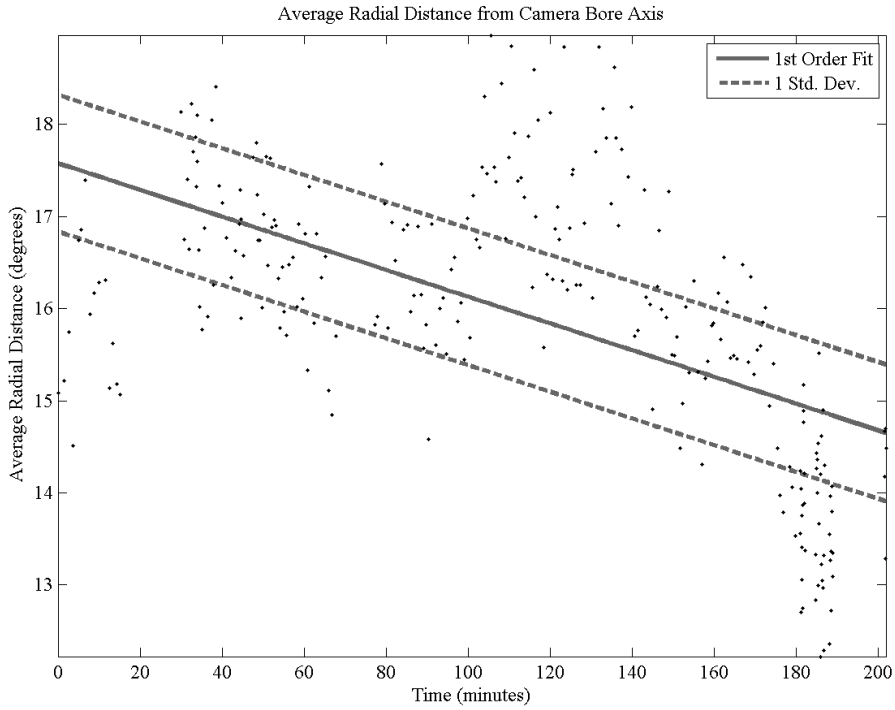

Figure C.50 Aptina – average spot radial distance away from the camera bore axis per image (result set 4)

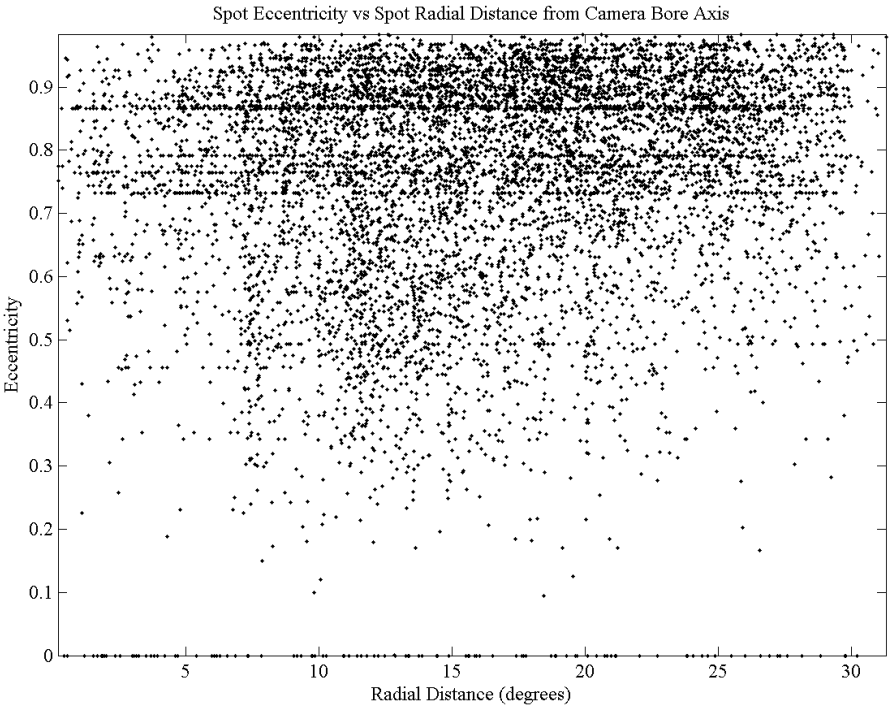

Figure C.51 Aptina – spot eccentricity compared to their radial distance away from the camera bore axis (result set 4)

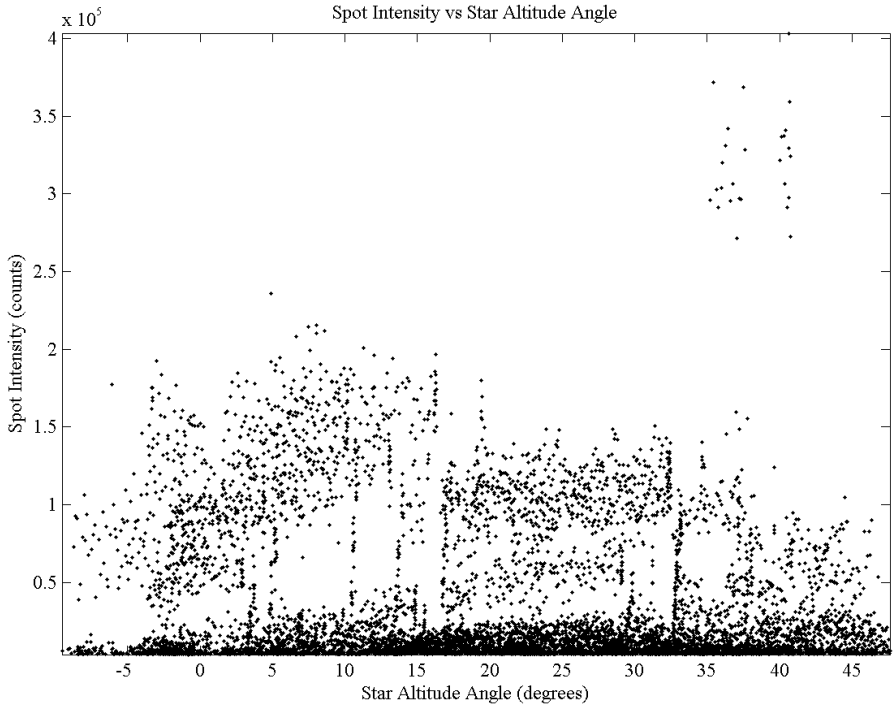

Figure C.52 Aptina – spot intensity compared to the star's altitude angle (result set 4)

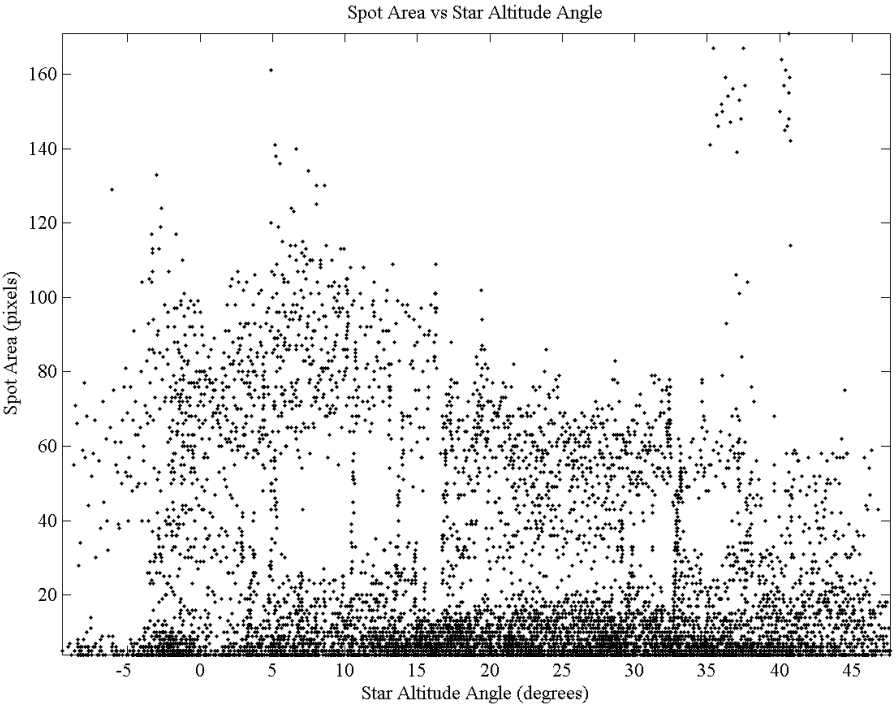

Figure C.53 Aptina – spot area compared to the star's altitude angle (result set 4)

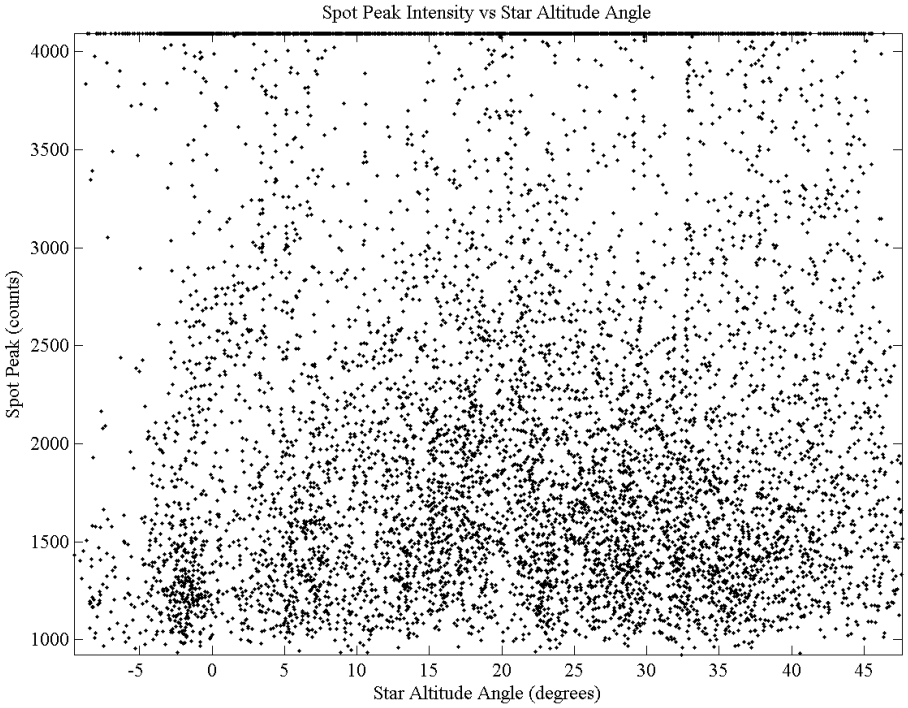

Figure C.54 Aptina – spot peak intensity compared to the star's altitude angle (result set 4)

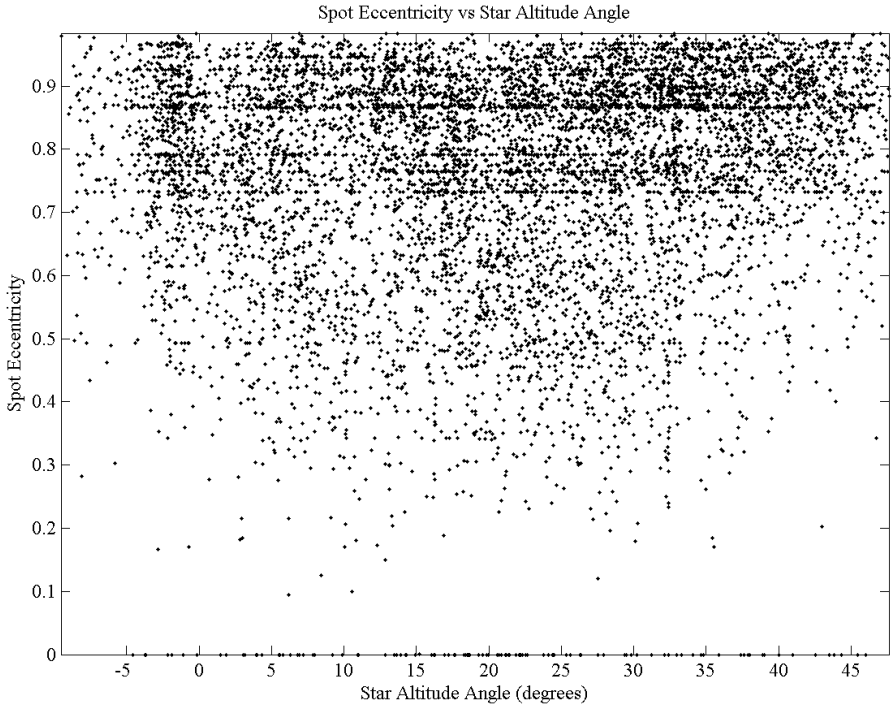

Figure C.55 Aptina – spot eccentricity compared to the star's altitude angle (result set 4)

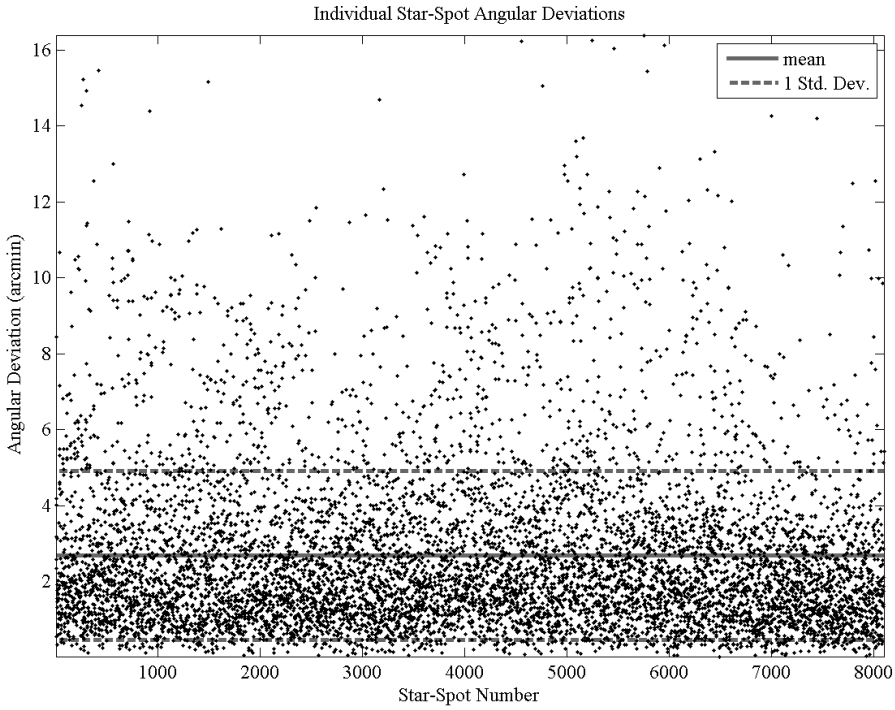

Figure C.56 Aptina – individual star to spot angular deviations (result set 4)

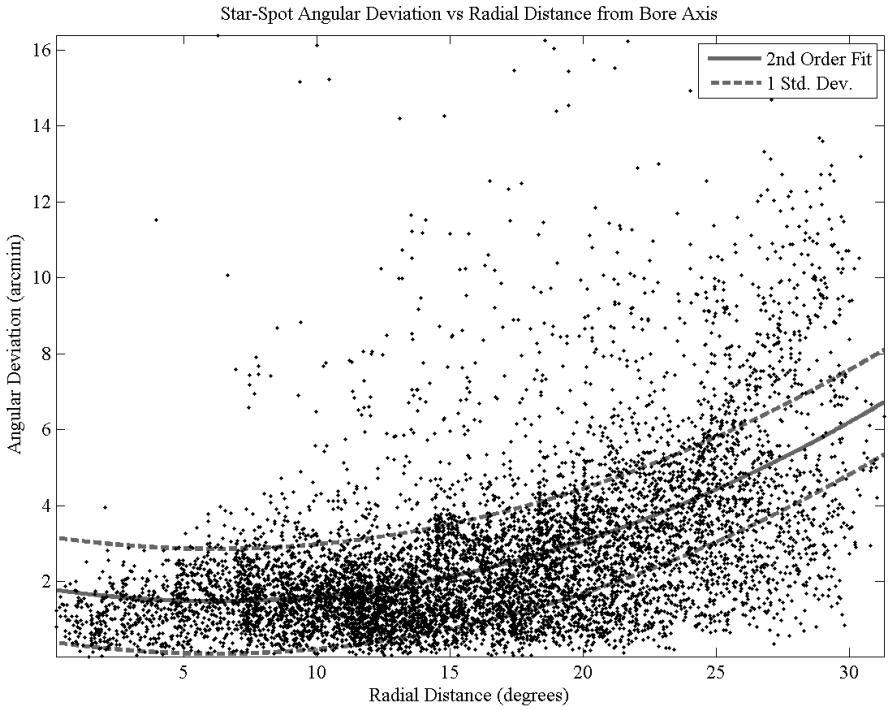

Figure C.57 Aptina – star to spot angular deviation compared to the spot's radial distance away from the camera bore axis (result set 4)