

---

Masters Theses

Student Theses and Dissertations

---

Fall 2017

## Development and implementation of star tracker based attitude determination

Casey Grant Smith

Follow this and additional works at: [https://scholarsmine.mst.edu/masters\\_theses](https://scholarsmine.mst.edu/masters_theses)



Part of the [Aerospace Engineering Commons](#)

Department:

---

### Recommended Citation

Smith, Casey Grant, "Development and implementation of star tracker based attitude determination" (2017). *Masters Theses*. 7724.

[https://scholarsmine.mst.edu/masters\\_theses/7724](https://scholarsmine.mst.edu/masters_theses/7724)

This thesis is brought to you by Scholars' Mine, a service of the Missouri S&T Library and Learning Resources. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact [scholarsmine@mst.edu](mailto:scholarsmine@mst.edu).

DEVELOPMENT AND IMPLEMENTATION OF STAR TRACKER BASED  
ATTITUDE DETERMINATION

by

CASEY GRANT SMITH

A THESIS

Presented to the Faculty of the Graduate School of the  
MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

2017

Approved by:

Kyle DeMars, Advisor

Henry Pernicka

Randy Moss

Copyright 2017  
CASEY GRANT SMITH  
All Rights Reserved

## ABSTRACT

As interest in nanosatellites grows within the university community, the demand for inexpensive, space-grade hardware grows as well. Star trackers can be a luxury item for some spacecraft and therefore are often not considered due to their cost. Ideally, a star tracker could be built using inexpensive parts so long as the software is available. Unlike many other attitude determination instruments, star trackers are renowned for their high accuracy, yielding accurate and precise attitude estimates. However, development of this software can be overwhelming for the university setting, especially when multiple missions are on hand. If these instruments were readily available for more spacecraft, university-sponsored missions could expand to higher orbits and possibly deep space applications.

Keeping in mind the cost and time constraints most university missions run into, the difficulty of developing an inexpensive star tracker stems from the integrated software. Hardware can be commercial off-the-shelf products, but the software is the more expensive of the two, and it is this software that is often lacking at the university level. With this, the proposed algorithm shows promise for the development, implementation, and testing of free star tracker software. The presented algorithm allows for a variety of interchangeable hardware, making it ideal for the academic community.

## ACKNOWLEDGEMENTS

First, I would like to thank my advisor and mentor Dr. Kyle DeMars. Your guidance and knowledge paved the path which allowed me to grow in my studies. Thanks to you, I have learned more and become interested in a field I never would have considered. The high standards you hold yourself, and all of your students, to has given me a strong work ethic which I look forward to bringing to my career.

Secondly, I would like to thank my committee members Dr. Hank Pernicka and Dr. Randy Moss. Your enthusiasm for what you do is nothing short of inspirational. The knowledge you passed along has greatly influenced my research and enjoyment in what I do.

Next, I would like to thank my lab mates. I think every day how I could have ended up in any lab, but I landed here. We are all one big family, and it's truly a group of friends I never thought I'd have. Without your support, your technical knowledge, and most importantly your advice, I would not have come as far as I have. James, Christine, Matt, Kari, Cameron, Kenny, Gunner, Sasha, and even Bruce, you all bring something unique to the lab, and without it, the lab would not be the same. I wish you all the best in your studies and future.

Lastly I want to thank my family and girlfriend. Without the love and unconditional support from my family I would not have been here, or made it this far. I wouldn't be the person I am today without you. To Hunter, words cannot express how important you are to me, and I sincerely would not have made it through grad school without you.

## TABLE OF CONTENTS

	Page
ABSTRACT . . . . .	iii
ACKNOWLEDGEMENTS . . . . .	iv
LIST OF ILLUSTRATIONS . . . . .	ix
LIST OF TABLES . . . . .	xi
SECTION	
1 INTRODUCTION . . . . .	1
1.1 OVERVIEW . . . . .	4
2 IMAGE CREATION . . . . .	7
2.1 PIXEL LOCATION DETERMINATION . . . . .	8
2.1.1 Checking the Field of View . . . . .	8
2.1.2 The Camera Matrix . . . . .	10
2.1.2.1 Intrinsic parameters . . . . .	10
2.1.2.2 Pinhole camera model . . . . .	11
2.1.2.3 Pixel coordinates . . . . .	12
2.1.2.4 Extrinsic parameters . . . . .	13
2.1.2.5 The formalized camera matrix . . . . .	15
2.1.3 Star Magnitude . . . . .	16
2.1.3.1 Converting to electrons . . . . .	18
2.1.3.2 Determining pixel spread . . . . .	19

2.2	STAR TRACKER NOISE CHARACTERIZATION . . . . .	23
2.2.1	Background and Motivation. . . . .	23
2.2.2	Noise Addition Process . . . . .	24
2.2.3	Photon Noise . . . . .	24
2.2.4	Electron Noise . . . . .	26
2.2.4.1	Dark current . . . . .	27
2.2.4.2	Pixel response non-uniformity . . . . .	27
2.2.4.3	Shot noise . . . . .	28
2.2.4.4	Read noise . . . . .	29
2.2.4.5	Amplifier noise . . . . .	30
2.2.5	Analog-To-Digital Noise . . . . .	31
2.3	FINAL IMAGE COMPARISON . . . . .	31
3	IMAGE PROCESSING . . . . .	35
3.1	NOISE REDUCTION . . . . .	36
3.2	THRESHOLDING AND STAR FINDING ALGORITHM . . . . .	37
3.3	CENTROIDING . . . . .	39
3.3.1	Center of Mass . . . . .	39
3.3.2	Uniform Mixture Model Approximation . . . . .	41
4	CATALOG FORMATION. . . . .	54
4.1	TROUBLES FOR CURRENT CATALOGS . . . . .	54
4.2	MOTIVATION . . . . .	56
4.3	BUILDING THE CATALOG . . . . .	57
4.3.1	Reference Catalog . . . . .	57

4.3.2	Catalog Outline . . . . .	58
4.3.3	Catalog Generation . . . . .	58
4.4	PROPERTY STATISTICS . . . . .	59
4.5	MONTE CARLO ANALYSIS. . . . .	64
4.6	MONTE CARLO ANALYSIS RESULTS . . . . .	67
5	ATTITUDE DETERMINATION. . . . .	73
5.1	BACKGROUND AND OVERVIEW . . . . .	73
5.2	ATTITUDE DETERMINATION SURVEY . . . . .	75
5.2.1	Davenport's $q$ -Method . . . . .	75
5.2.2	Quaternion Estimator (QUEST) . . . . .	77
5.2.3	Estimator of the Optimal Quaternion (ESOQ) . . . . .	77
5.2.4	Singular Value Decomposition (SVD) . . . . .	79
5.2.5	Fast Optimal Attitude Matrix (FOAM) . . . . .	81
5.3	THE QUATERNION ESTIMATOR . . . . .	82
5.3.1	The Quaternion Eigenvalue Problem. . . . .	82
5.3.2	QUEST Derivation . . . . .	86
5.4	ANALYSIS OF SURVEY . . . . .	89
6	SIMULATION AND RESULTS. . . . .	96
6.1	SIMULATION CONFIGURATION . . . . .	96
6.1.1	Simulation Time Analysis . . . . .	98
6.1.2	Simulation Centroiding Analysis . . . . .	100
6.1.3	Simulation Catalog Matching Analysis. . . . .	100
6.1.4	Simulation Attitude Determination Analysis . . . . .	101



7 CONCLUSIONS . . . . .106

BIBLIOGRAPHY . . . . .108

VITA . . . . .112

## LIST OF ILLUSTRATIONS

Figure	Page
2.1 An illustration depicting the geometry of a 3D pinhole camera model. <sup>2</sup> . . .	8
2.2 Every star seen inside of the 25° field of view plotted on a unit sphere.. . .	10
2.3 The results from the camera matrix, Eq. 2.14, for a 25° FOV.. . . . . .	16
2.4 A block diagram showing the conversion process from flux density to the associated number of electrons.. . . . . .	20
2.5 The pixel spread for the brightest star in Figure 2.6. . . . . . . . . . .	22
2.6 Illustrated here are each of the Gaussian functions that form the pixel spreads for each star in the FOV. . . . . . . . . . .	23
2.7 An illustration of the result from quantization error applied to Figure 2.5..	32
2.8 Final image comparison.. .	33
2.9 A comparison between image values, the blue line being the signal from Figure 2.8a and the red being the signal from Figure 2.8b.. . . . . . . . . . .	34
3.1 A histogram of Figure 2.8b.. .	37
3.2 An illustration of the star detection method.. . . . . . . . . . . . . . . . .	38
3.3 Illustrating the region of interest via the red boxes. . . . . . . . . . . . . . . . .	40
3.4 Zoomed in region of interest on the image's brightest star.. . . . . . . . . . .	40
3.5 Examples of non-negative, univariate, uniform mixture model pdfs.. . . . .	41
3.6 An uncorrelated Gaussian pdf without noise.. . . . . . . . . . . . . . . . .	50
3.7 The Gaussian pdf with minimal noise added.. . . . . . . . . . . . . . . . .	50
3.8 The centroids found by the uniform mixture model and center of mass methods are shown, along with the true centroid at (5,5). . . . . . . . . . .	52
3.9 The centroids found by the uniform mixture model and center of mass methods are shown, blue and green, respectively, for Figure 3.3.. . . . . . . . . . .	53

4.1	The 500 brightest stars of the Tycho-2 Catalog, in the visual spectrum, mapped onto a unit sphere. . . . .	55
4.2	The icosahedron sphere used to change positions over the sky, ensuring complete coverage. . . . .	59
4.3	Field of view for $\bar{\mathbf{q}}$ , blue dots indicating all the stars seen in the image, and the green dot represents the pointing vector. . . . .	69
4.4	Catalog matching example image. . . . .	70
5.1	The average computation time for each attitude estimation algorithm. . . .	94
5.2	The relative average computation time which relates how the methods perform when given more measurements. . . . .	95
6.1	A histogram of the Monte Carlo trial times. . . . .	99
6.2	A histogram of the number of iterations needed to converge. . . . .	103
6.3	One example of unsuccessful thresholding. . . . .	103
6.4	A second example of unsuccessful thresholding. . . . .	104
6.5	The attitude errors from the complete algorithm Monte Carlo simulation. .	105

## LIST OF TABLES

Table	Page
4.1 Tycho-2 Star Catalog Format. . . . .	57
4.2 Star-Set Triangle Properties. . . . .	64
4.3 Property Catalog Format, Page One – Perimeter. . . . .	68
4.4 Monte Carlo Example – Perimeter. . . . .	71
4.5 Monte Carlo Example – Area. . . . .	71
4.6 Monte Carlo Example – Polar Moment.. . . .	72
4.7 Monte Carlo Example – Side-Length Ratio. . . . .	72
4.8 Monte Carlo Example – Normal Coefficient. . . . .	72
5.1 Mean Attitude Errors in Arcseconds. . . . .	91
5.2 Computer Specifications. . . . .	93
6.1 Image Creation Parameters. . . . .	97
6.2 Computer Specifications. . . . .	98

## 1. INTRODUCTION

Star trackers have become one of the most accurate attitude measurement instruments on spacecraft over the past decades, providing accuracies on the order of arcseconds. Early star trackers were broken up into several categories, consisting of star scanners, gimballed star trackers, and fixed-head star trackers [1]. Current star trackers are based off these fundamental models, and starting around the early 1990s, the current solid state star trackers began replacing the older models. Current star trackers provide the same level of accuracy, but are much smaller, less expensive, and require less power [2]. When operating, star trackers work in two different modes: a “lost-in-space” mode and a tracking mode. The former mode occurs when the spacecraft has no *a priori* information about its current attitude, forcing the device to back out the attitude with greater uncertainty and is often more computationally expensive than the latter. Tracking mode starts once the stars in an image are known, and the system then tracks them in an image based on the current attitude dynamics [3]. The focus of this study is the more computationally expensive, lost-in-space mode, which can be a harder problem to solve than the tracking mode due to the lack of any *a priori* information.

With satellites becoming smaller and more compact, the need for inexpensive, reliable hardware is evident. Star trackers are one of the higher-priced satellite instruments, but arguably one of the most useful. Implementing one on-board a spacecraft can provide valuable measurements that are more accurate than several other types of sensors, such as inertial measurement units, magnetometers, sun sensors, etc. The aim is to develop a star tracker algorithm with inexpensive hardware considerations that will allow budget-constrained satellite projects to use a robust attitude determination procedure. Industry and researchers can conduct a wider range of tests with several smaller, low-cost satellites instead of a single, more complex satellite.

University interest has steadily increased over the past decades in developing these smaller satellites. Developing a customizable, inexpensive star tracker can greatly reduce costs for future university missions.

An issue any star tracker developer will inherently encounter is finding a means of testing. Even if quality imaging equipment is available, finding a location with limited light interference can be troublesome. Star field simulator software has been developed, but is not always freely available. Samaan *et al.* utilize the free simulator software dSPACE<sup>1</sup> and develop a star simulation package called Optical Sky field sImulator (OSI) [4]; however, this requires specific hardware to work. Instead, one would like to be able to digitally simulate an image based purely on a known star catalog and a chosen sensor and lens. The problem with this path is that image sensor errors are often overlooked, and random white noise is often added instead of focusing on the true errors common to image sensors [5–7]. Knutson provides an in-depth look at the common sensor errors, and Houtz provides a means of accounting for zodiacal light noise in their simulated images [3, 8]. By combining the works, and offering a comprehensive way of adding noise to the image, an accurate image is simulated using only a reference star catalog and commercial off-the-shelf hardware properties.

Image processing techniques for star identification are often simplified to center of mass calculations, as this is not a main focus for most. The center of mass method is the most widely used algorithm and can be seen in many works [7, 9, 10]. This method is often the basis for others, which utilize its simplicity and combine it with spacecraft dynamics to provide faster solutions [3]. Work done by Padgett and Delgado uses a grid search technique to locate stars in an image [6]. Two-dimensional Gaussian fitting have provided more precise centroiding than the standard center of mass method, and through iterations, it becomes increasingly accurate [11, 12]. The method developed here utilizes work done by Darling, from his dissertation, which facilitates a uniform

---

<sup>1</sup><https://www.dspace.com/en/inc/home.cfm>

mixture model, such as an image array, to be approximated by the parameters of a Gaussian that minimize the Kullback-Leibler (KL) divergence between the two without iteration [13]. This method determines the mean and covariance of a Gaussian distribution best fits an image acquired of a star, with respect to the KL-divergence. If an alternative solution is sought, such as a maximum likelihood solution, the mean and covariance obtained from the KL divergence minimization process can be used to initialize iterative improvement of a secondary cost function. Along with the uniform mixture model centroiding technique, an autonomous thresholding technique is developed that allows the star tracker to select the correct number of centroids from an image with zero user input.

An accurate and fast star identification technique is the core of what makes a star tracker unique. What most methods boil down to is a pattern matching algorithm. The traditional method involves matching angles between stars in an image with the angles between those stored in the star catalog [14, 15]. Mortari has developed a few different star identification schemes [16–18]. One method Mortari provides is the pyramid star identification technique that takes four stars in an image and compares the resulting pyramid with his catalog [19]. An alternative is to leverage the spherical area and polar moment of inertia of a spherical triangle formed by a star triad in an image [20]. The method was quickly expanded to the planar triangle, and although the authors made no direct comparison between the two methods, their results show better performance from the planar case [21]. The method of star identification presented here follows closely to that provided by Houtz and Frueh, who expand on the triangle properties. Their star identification is based on comparing four different properties in order to provide higher accuracy [8]. By comparing five planar triangle properties, the aim of this work is to decrease false star identification and provide a robust process that connects the star identification process with the custom star catalog formation.

Every system is a sum of its parts, and therefore no piece of the system can be lacking in order to provide the best performance. Intelligently integrating the system parts together can allow for a streamlined algorithm that makes intuitive sense. The major link of the presented star tracker algorithm lies between the image processing scheme and the star catalog matching process. The catalog formation process, based on star-set triangle properties, is linked to the autonomous thresholding technique that obtains exactly three sets of centroid coordinates in an image. This set of centroids act as the vertices of a triangle, whose properties are determined and are subsequently compared to those in the custom built star catalog. The star catalog contains statistics for every possible star-set triangle for any given field of view, and using the Mahalanobis distance, properties are measured against one another. The star-set that most closely represents these properties is chosen as the corresponding stars in the image.

### **1.1. OVERVIEW**

The thesis is broken up into four main sections, each section focusing on the separate operations a star tracker undergoes once capturing an image. However, without a realistic image to test on, the rest of the steps become impractical. Section 2 addresses this problem. In order to form an image, a specific sensor is needed in order to use its properties to replicate how it would see an image. As well, selecting a specific sensor allows for practical representations of the various errors that occur in star tracker images. The stars in the images are formed from a reference catalog and selected based on specified criteria discussed later.

With that, Section 3 discusses the image processing procedures. Star trackers are not known for well-developed image processing, since the goal is obtaining an attitude measurement. However, powerful image processing techniques are critical for detecting the correct objects in a noisy image. This study focuses on a simple



noise reduction technique that subtracts off a static background noise from the image as a whole. From there, two methods of centroiding are implemented. The first is a basic center of mass calculation, which uses the weights and positions of pixels to determine the center. The second centroiding algorithm utilizes a minimization of the Kullback-Leibler divergence between a uniform mixture model and a Gaussian distribution. By treating the pixel array as a multivariate uniform mixture model, the algorithm determines a best fit, with respect to the minimization of the KL-divergence, of a Gaussian over the array.

Section 4 focuses on determining which star triad the identified centroids belong to. The method here is based off comparing triangle properties of a star triad from an image to those values stored in a custom star catalog. To build the custom star catalog, a Monte Carlo simulation is performed to characterize the statistics of each star triad's properties. The five planar properties are triangular perimeter, area, polar moment, the ratio of shortest to longest side length, and finally what is called the "normal coefficient." Once the triangle properties' means and standard deviations are found from the Monte Carlo simulation, a test of the Mahalanobis distance is used to compare a triad of stars in an image to those in the customized star catalog.

Included in Section 5 is a survey of the various attitude determination techniques available. The survey studies two types of attitude determination techniques: quaternion-based and attitude-matrix-based methods. These methods studied are Davenport's  $q$ -method, QUaternion ESTimator (QUEST), ESTimator of the Optimal Quaternion (ESOQ), Singular Value Decomposition (SVD), and Fast Optimal Attitude Matrix (FOAM). A comprehensive derivation of the QUEST method is provided, along with a detailed analysis of the covariance matrix produced by this method. The QUEST algorithm is chosen due to the vast acceptance and use of the quaternion-based method. The section ends with a characterization of the error and computational efficiency of each method.

The final core section of the thesis, Section 6, provides a full simulation of the star tracker algorithm. The simulation encompasses rigorous analysis of the image formation, image processing, catalog matching, and attitude determination algorithms to test the limits of these components. To follow the analysis, concluding remarks and future work considerations are provided in Section 7.

## 2. IMAGE CREATION

Obtaining an image is the first step of any star tracker. From this image, objects (stars) and their centroids are found. The next step includes further processing of the images to enable the star camera to determine the stars observed and ultimately the spacecraft's attitude. Single images taken by telescopes on Earth's surface only capture a fraction of the stars a spacecraft star tracker is capable of seeing above the atmosphere. The stars a telescope are able to capture would appear dimmer or distorted compared to those a spacecraft sees during flight due to Earth's atmosphere, which absorbs and scatters the light that passes through it. In order to test the star tracker's attitude determination software prior to launch, images must be simulated.

To create these virtual images, a reference star catalog is needed. The Tycho-2 star catalog is used in this work due to the number of stars as well as the accuracy of their characteristics. In order to have a realistic image, a magnitude cap is set and other less favorable stars are eliminated. A deeper discussion of this is found in Section 3.

A circular field of view (FOV) is selected, defined by a half angle, along with a pointing vector directed out the bore-sight of the camera. Stars within the chosen FOV are projected onto an image plane, where their magnitudes are converted to a corresponding electron count. The magnitudes are expressed as two-dimensional Gaussian functions in order to account for the  $(u, v)$  image directions, and are assumed to be circular. The standard deviation of each Gaussian is determined by the sensor and optics of the system and defined in units of pixel width. The "volume" under the Gaussian is proportional to the number of electrons the CCD sensor would receive for each star.

Along with the stars, the noise associated with CCD sensors and that due to the environment of space are added into the image in order to better capture what is

seen during flight. These consist of photon, electron, and analog-to-digital conversion errors. Once the image is formed with the stars in place, the three types of noise are added to the image as a whole.

## 2.1. PIXEL LOCATION DETERMINATION

**2.1.1. Checking the Field of View.** The field of view is defined here with only one angle, making it circular. This angle is defined by the angle from the center of the lens to the outer edge of the field of view, making it half of the entire FOV. Next, the camera's pointing direction must be determined in order to obtain the stars in its FOV. The camera frame is defined with the  $z$ -axis pointing out the bore-sight, the  $y$ -axis out of the top of the camera, and the  $x$ -axis completes the right-hand frame. Illustrated in Figure 2.1, the camera is centered at the origin  $\mathcal{O}$ . With the

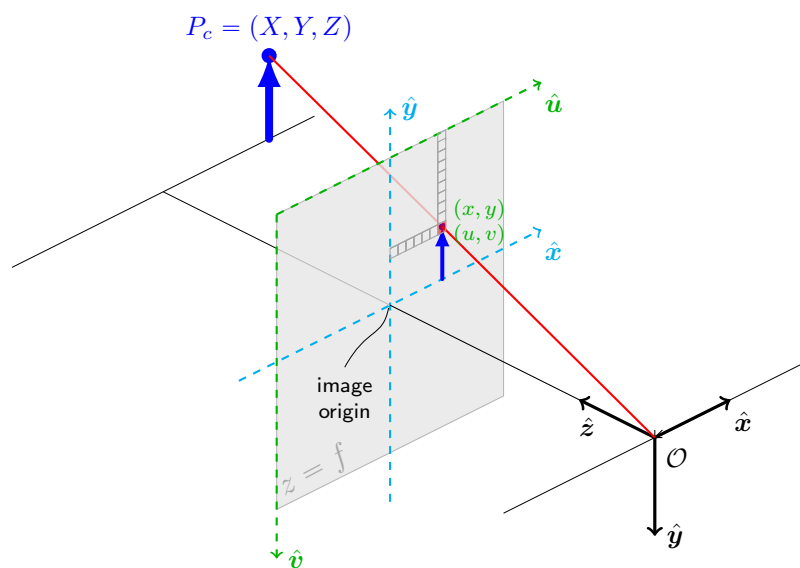


Figure 2.1. An illustration depicting the geometry of a 3D pinhole camera model.<sup>2</sup>

<sup>2</sup>Figure adapted from: <https://tex.stackexchange.com/questions/96074/more-elegant-way-to-achieve-this-same-camera-perspective-projection-model>

frame and FOV defined, next is the camera's attitude, or orientation with respect to another reference frame. Here, stars are in the inertial frame that is the J2000 Earth centered inertial (ECI) frame. J2000 is defined with the Earth's mean equator and equinox at 12:00 U.T. on January 1, 2000. For the camera frame, a random attitude is formed and expressed in the form of a complex four-component quaternion. To generate a random attitude, two random angles are obtained such that  $\theta \in [0, 2\pi)$  and  $\phi \in [0, \pi]$ , and are then formed into an axis

$$\mathbf{e} = \begin{bmatrix} \cos(\phi) \cos(\theta) \\ \cos(\phi) \sin(\theta) \\ \sin(\phi) \end{bmatrix}. \quad (2.1)$$

The axis can then be rotated by any angle  $\omega \in [0, \pi)$ , however since the FOV is circular, the pointing vector  $\mathbf{e}$  is all that is utilized. Finally, the axis is added to the vector portion of an identity quaternion such that

$$\tilde{\mathbf{q}} = \begin{bmatrix} \mathbf{e} \\ 1 \end{bmatrix} \quad (2.2)$$

The newly formed vector is normalized to form the attitude quaternion,  $\bar{\mathbf{q}}$ . This process allows for an intuitive approach to defining a random attitude for the camera frame since the two angles  $(\theta, \phi)$  relate to right-ascension and declination.

From here, the entire catalog is searched to find all stars within the camera's FOV by determining the angle between the camera's  $z$ -axis and each star's ECI unit vector. This angle is then checked to see if it is equal to or less than half the field of view angle. To give a visual idea of how much of the sky a star tracker with a  $20^\circ$  FOV could view is shown in Figure 2.2.

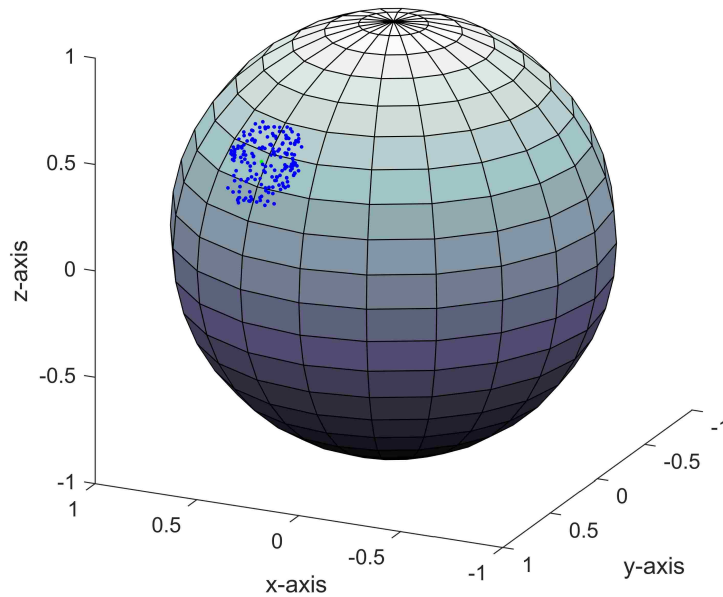


Figure 2.2. Every star seen inside of the  $25^\circ$  field of view plotted on a unit sphere. The blue dots represent stars and the green dot represents the camera bore-sight pointing vector.

**2.1.2. The Camera Matrix.** The next step to creating an image is to take the three-dimensional points representing the unit vectors of each star and project them onto a two-dimensional image plane. To do so, there are two sets of camera parameters to consider: the intrinsic parameters, which take into account the properties of the sensor; and the extrinsic parameters, which include the location and attitude of the camera with respect to the inertial frame. These parameters come together to form the camera matrix.

**2.1.2.1. Intrinsic parameters.** The sensor and lens chosen are crucial to determining the intrinsic parameters. These parameters are what characterize the optical, geometric, and digital characteristics of a camera. Two portions make up the complete intrinsic parameters matrix. The first half is defined by the focal length,  $f$ ,

with the second half describing the relationship between real world coordinates and pixel coordinates. This matrix can take into consideration the geometric distortion introduced by the lens. Unfortunately, geometric distortions are not easily characterized and are best determined from testing physical hardware. In order to understand the basis of the intrinsic parameters, it is necessary to understand a simple camera model and its geometry.

**2.1.2.2. Pinhole camera model.** The pinhole camera model allows for a simple way to describe the relationship between the 3D coordinates of a point and its projection onto an image plane through an ideal pinhole camera. It is assumed that the camera aperture is a point with no lens to focus the light, which gives an approximation to the projective transformation. The validity of this approach rests with the quality of the sensor and lenses used. Fortunately, typical star trackers require very high specifications for the camera sensor and lenses used; therefore, these assumptions hold.

Using basic geometry, the 2D image coordinates of a 3D point can be found, as illustrated in Figure 2.1. If the point  $P_c = (X, Y, Z)$  is a distance  $Z$  away from the origin in the camera frame, it is shown that the point will be projected onto an image plane of distance  $f$  away from the origin.

Through the similar triangles formed, it can be found that the 2D image coordinates,  $p = (x, y)$ , are formed by

$$x = f \frac{X}{Z} \quad \text{and} \quad y = f \frac{Y}{Z}, \quad (2.3)$$

which is known as a perspective projection. Given Eq. (2.3), the point  $P_c$  is now projected onto the 2D image plane a distance  $f$  from the origin. In order to simplify this process, a new point  $p' = (x', y', z')$  is introduced as the projective coordinates of the 3D point. The use of projective coordinates allows for all of the scaling,

translation, and rotation of  $P_c$  to be represented by a single matrix. The image coordinates for  $p$  are found using the projective coordinates, similarly to Eq. (2.3),

$$x = \frac{x'}{z'} \quad \text{and} \quad y = \frac{y'}{z'}. \quad (2.4)$$

Therefore,  $p = (x, y)$  will now be the projective set of homogeneous coordinates  $(x, y, 1) = (kx, ky, k)$  for any nonzero  $k$ , positive or negative. Here, only the scaling of the focal length is taken into account, yielding

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \quad (2.5)$$

where the center matrix is called the projection matrix  $\mathbf{M}_{\text{proj}}$  which makes up the first half of the intrinsic matrix.

**2.1.2.3. Pixel coordinates.** After reducing dimensions, the pixel coordinates of  $p$  must be found. The image plane is digitized and forms the  $(u, v)$  pixel array. As a result of this process, lens and sensor characteristics dictate how the image plane coordinates transform to the  $(u, v)$  pixel coordinates. Utilizing the pixel size and image plane origin, the other half of the intrinsic matrix is formed.

Due to the nature of CCD sensors, the pixel origin lies in the top left corner of the image at pixel  $(1, 1)$ . From there,  $u$  is positive along the traditional  $x$ -axis and  $v$  is positive along the negative  $y$ -axis. In the image plane, the  $x$ -axis and  $y$ -axis are the standard, positive to the right and positive up, respectively, as shown in Figure 2.1. Knowing this, and taking into account the pixel size of the sensor, pixel coordinates



$(u, v)$  are calculated as

$$u = \frac{1}{\rho_u} f \frac{X}{Z} + u_0 \quad \text{and} \quad v = -\frac{1}{\rho_v} f \frac{Y}{Z} + v_0 \quad (2.6)$$

where  $(u_0, v_0)$  is the pixel location of the origin of the image plane, and  $\rho_u$  and  $\rho_v$  are the sizes of each pixel in meters. Putting Eq. 2.6 into the similar matrix form as Eq. (2.5) yields

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} 1/\rho_u & 0 & u_0 \\ 0 & -1/\rho_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}. \quad (2.7)$$

The matrix in Eq. (2.7) is known as the affine matrix  $\mathbf{M}_{\text{aff}}$ . An affine matrix, in general, preserves points and straight lines by utilizing methods of translation, scaling, and rotations. Here, scaling is done by the pixel size and translation is handled with the origin shift. Again, using the same process for the image coordinate  $(x, y)$ , the point  $(u, v)$  is found by scaling with the  $w'$  component by

$$u = \frac{u'}{w'} \quad \text{and} \quad v = \frac{v'}{w'}. \quad (2.8)$$

**2.1.2.4. Extrinsic parameters.** The extrinsic parameters of the camera consider how the camera's attitude as well as its location relate to the stars' inertial coordinate system. The location of the camera is kept constant at the origin of the inertial coordinate system, and the camera's attitude, which mostly depends on its bore-sight pointing vector, is determined at random for these simulated images. With this, the two components that make up the extrinsic camera matrix are a transformation matrix,

$$\mathbf{R} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \quad (2.9)$$

and a column vector  $\mathbf{t} = [t_x \ t_y \ t_z]^T$ . These parameters are put together to form the following extrinsic parameters matrix,

$$\mathbf{M}_{\text{ext}} = \left[ \begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array} \right] = \left[ \begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \quad (2.10)$$

It is important to note that  $\mathbf{R}$  and  $\mathbf{t}$  do not directly correspond to the camera's attitude and location. That is, the extrinsic matrix describes how the inertial coordinate system is transformed relative to the camera frame. That relationship is often less intuitive since it is more natural to express the camera's position and attitude in the inertial space, rather than specifying how the reference coordinate system transforms to the camera's. To do so, the camera's attitude and location are specified in the inertial frame, with  $\mathbf{t}_C$  being the location of the camera center and  $\mathbf{R}_C$  representing the attitude of the camera. These are then put into a square matrix and inverted, which yields

$$\mathbf{M}_{\text{ext}} = \left[ \begin{array}{c|c} \mathbf{R} & \mathbf{t} \\ \hline \mathbf{0} & 1 \end{array} \right] = \left[ \begin{array}{c|c} \mathbf{R}_C & \mathbf{t}_C \\ \hline \mathbf{0} & 1 \end{array} \right]^{-1} = \left[ \begin{array}{c|c} \mathbf{R}_C^T & -\mathbf{R}_C^T \mathbf{t}_C \\ \hline \mathbf{0} & 1 \end{array} \right]. \quad (2.11)$$

The inversion is due to the fact that  $\mathbf{R}_C$  describes the attitude of the camera with respect to the inertial coordinate frame, however the stars must be rotated into the camera frame. Transformation matrices have the convenient property where their inverse is the opposite transformation. If a transformation matrix  $\mathbf{T}$  describes the rotation from frame  $\mathcal{A}$  to frame  $\mathcal{B}$ , then the matrix can be inverted to form  $\mathbf{T}'$  which describes the rotation between frame  $\mathcal{B}$  to frame  $\mathcal{A}$ . Thus, the relationships  $\mathbf{R} = \mathbf{R}_C^T$  and  $\mathbf{t} = -\mathbf{R}_C^T \mathbf{t}_C = -\mathbf{R} \mathbf{t}_C$  allow for the extrinsic parameters matrix to be formed using information about the camera. With the extrinsic matrix defined, the camera

frame point  $P_c$  can be calculated as

$$\begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \left[ \begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \quad (2.12)$$

where  $P_w = (U, V, W)$  is the location of a star in the inertial frame before it is transformed into the camera frame.

**2.1.2.5. The formalized camera matrix.** The goal of the camera matrix is to encapsulate all the projective transformations that occur when capturing an image into a single matrix equation. Merging both the intrinsic and extrinsic parameter matrices forms the complete camera matrix  $\mathbf{M}$ , such that

$$\mathbf{M} = \mathbf{M}_{\text{int}}\mathbf{M}_{\text{ext}} = \mathbf{M}_{\text{aff}}\mathbf{M}_{\text{proj}}\mathbf{M}_{\text{ext}}. \quad (2.13)$$

The matrices are combined into a single relationship that shows how the projective transformation changes the three-dimensional inertial point  $P_w$  into the two-dimensional pixel coordinates,

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \sim \begin{bmatrix} 1/\rho_u & 0 & u_0 \\ 0 & -1/\rho_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \left[ \begin{array}{ccc|c} R_{11} & R_{12} & R_{13} & t_x \\ R_{21} & R_{22} & R_{23} & t_y \\ R_{31} & R_{32} & R_{33} & t_z \\ \hline 0 & 0 & 0 & 1 \end{array} \right] \begin{bmatrix} U \\ V \\ W \\ 1 \end{bmatrix} \quad (2.14)$$

The order in which the matrices are arranged is key, noticing that the extrinsic matrix is on the right. This allows for the transforming of point  $P_c$  from the inertial coordinate frame to the camera coordinate frame. Once in the camera frame, the intrinsic matrices is multiplied, allowing for a reduction of dimensions. The tilde

in Eq. (2.14) implies that it is not the direct equation, but instead, the equation produces  $(u', v', w')$  from Eq. (2.7). In order to obtain the pixel coordinate  $(u, v)$ , the relationship shown in Eq. (2.8) is applied. The final product is a plot of pixel locations shown in Figure 2.3.

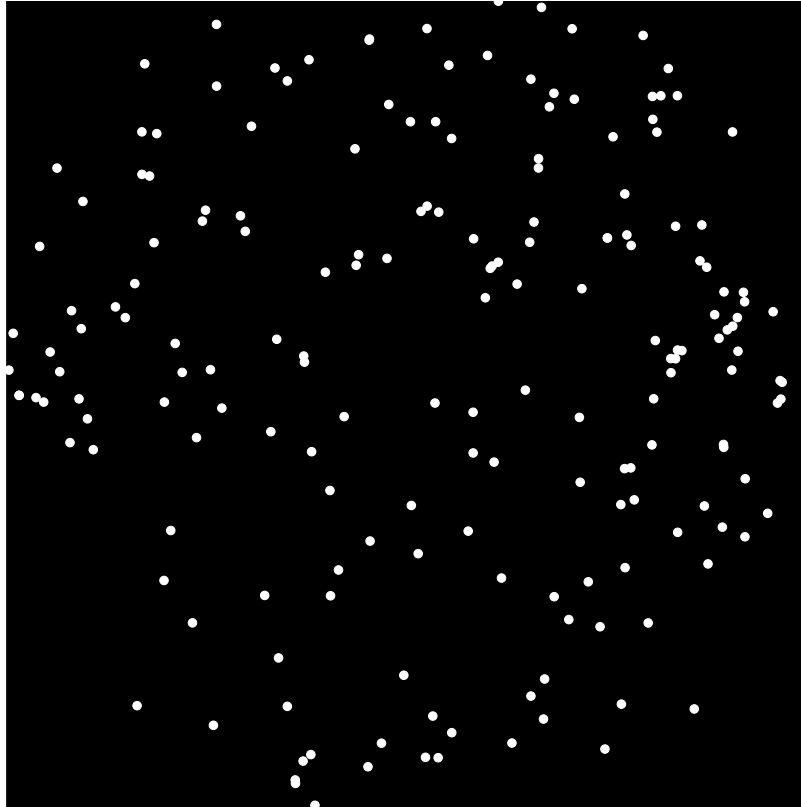


Figure 2.3. The results from the camera matrix, Eq. 2.14, for a  $25^\circ$  FOV. Each white dot is a  $(u, v)$  coordinate for its corresponding star seen in this FOV.

**2.1.3. Star Magnitude.** The previous section developed the method for determining the pixel location for a star in the FOV. The next step in producing a simulated image is to calculate its corresponding brightness, or magnitude. To calculate a star's magnitude it is important to understand astronomical magnitudes. In astronomy, the brightness, or magnitude, of a celestial body is defined in a logarithm-

mic scale and is often measured in a specific wavelength, spectrum, or photometric band. Typically, magnitudes are measured in the visual or near-infrared spectrum, wavelengths from hundreds of nanometers up to micrometers. There are two main methods for categorizing an object’s magnitude: apparent magnitude and absolute magnitude. Apparent magnitude corresponds to the brightness of an object an observer would see at Earth, whereas the absolute magnitude describes the inherent brightness of an object if it were placed 10 parsecs from Earth. One parsec is approximately 3.26 light-years in length. Since the scale is logarithmic, the brightest stars are close to zero in the absolute scale, and go negative depending on the zero point star used. Absolute magnitudes tell the most about a star, since the absolute magnitude captures the true nature of the star by equally comparing them. For this reason, the absolute magnitude is used to determine the corresponding pixel value in an image.

There are several types of magnitude systems which in turn use different types of wavelengths to obtain magnitude data. The most widely used magnitude system is the Johnson system that was developed by Harold L. Johnson in 1955. The work presented by Johnson set the standards for stellar photometry and defined a fundamental system of magnitudes and wavelength spectrums, mostly concentrating on the ultra-violet, blue, and visual (UBV) filters. UBV wavelength filters are commonly used for all magnitude systems, and the Johnson system has since been extended to the near-infrared filters.

The Tycho-2 star catalog gives the magnitudes  $B_T$  and  $V_T$ , where the subscript  $T$  distinguishes between the Johnson system and the Tycho-2 catalog system. Note that these blue and visual spectrum values are not the same as the Johnson system’s. The Tycho-2 magnitudes are, in general, more accurate than those of the Johnson system. On average, the Tycho-2 catalog has 130 measurements per star. Therefore, when linearly transforming from the Tycho-2  $B_T$  and  $V_T$  to the Johnson  $B$  and  $V$

values, residuals of up to 0.1 magnitudes may occur [22]. As well, the Tycho-2 catalog includes more stars than Johnson could have recorded. In the limited cases where the Johnson magnitudes are more accurate than those of the Tycho-2 catalog, the Johnson values are kept.

**2.1.3.1. Converting to electrons.** To simulate images for testing, the magnitudes of stars in the FOV must be converted to a corresponding number of electrons through a series of steps. The first step is to convert a star's absolute magnitude to an irradiance value. Solar irradiance is the power per unit area emitted by a star, and it is directly related to the magnitude of the star. Star magnitudes work by relating all stars to one specific star, called the zero point star. Every star in a specific photometric band share a relationship with the zero point star which is

$$m_* = m_{*,0} - 2.5 \log_{10} \left( \frac{F_*}{F_{*,0}} \right)$$

where  $m_*$  is the magnitude of a star in a given photometric band,  $F_*$  is that star's flux in the same photometric band, and  $F_{*,0}$  is the zero point star in the photometric band used. Traditionally, the zero point star is used, but any reference star can be used as long as each star in the band is compared to the same reference star. The 2.5 is due to the effect of using a logarithmic scale. A difference of 2.5 magnitudes corresponds to a decrease in flux by a factor of 10. From this relation, the magnitude of one star can be found by comparing it with another star's magnitude within its photometric band. This goes for the irradiance value of stars as well. Here, stars are compared to the Sun due to the abundant information known about it. Knowing the solar irradiance and magnitude of the Sun, as well as the magnitude of another star, allows for the irradiance of that star to be determined by using

$$m_* = m_{\odot} - 2.5 \log_{10} \left( \frac{I_*}{I_{\odot}} \right)$$

where  $m_{\odot}$  and  $I_{\odot}$  are the magnitude and irradiance of the Sun respectfully, and  $m_*$  is the magnitude of the star and  $I_*$  is the corresponding irradiance.

From here,  $I_*$  is multiplied by the square of the exposure time of the sensor. Next, it is converted into energy by multiplying by the pixel area taken up by the spread of the star. The energy is then divided by Planck's constant,  $h = 6.626070041 \times 10^{-34}$  Js, which is a proportionality constant between energy and photons. The final step is to adjust for the quantum efficiency of the sensor, which is the ratio of the number of electrons produced per photon. Accounting for the quantum efficiency is done by simply multiplying by this ratio. A block diagram in Figure 2.4 shows the transition and steps, as well as the progression of units from magnitude to the number of associated electrons. As well, Eq. 2.15 summarizes all of the steps into one computation

$$e = \frac{1}{h} I_{\odot} t_{\text{int}}^2 \sigma_x \sigma_y \rho_x \rho_y Q(\lambda) \times 10^{-(m_* - m_{\odot})/2.5} \quad (2.15)$$

where  $t_{\text{int}}$  is the integration time,  $\sigma_x$  and  $\sigma_y$  represent the pixel spread of each star,  $\rho_x$  and  $\rho_y$  are the pixel sizes, and  $Q(\lambda)$  is the quantum efficiency of the sensor for a given wavelength  $\lambda$ . Once the electron count for every star is found, the image is scaled to ensure that the maximum pixel value corresponding to the brightest star is set to the saturation limit the sensor can hold. This allows for the image to appear as bright as possible, without oversaturation of the pixels.

**2.1.3.2. Determining pixel spread.** Now that the number of electrons has been calculated, the area of pixels taken up by a star in the FOV can be determined. The electron count can be considered the “volume” under a two-dimensional Gaussian function, which provides a more intuitive way of determining the spread of a star.

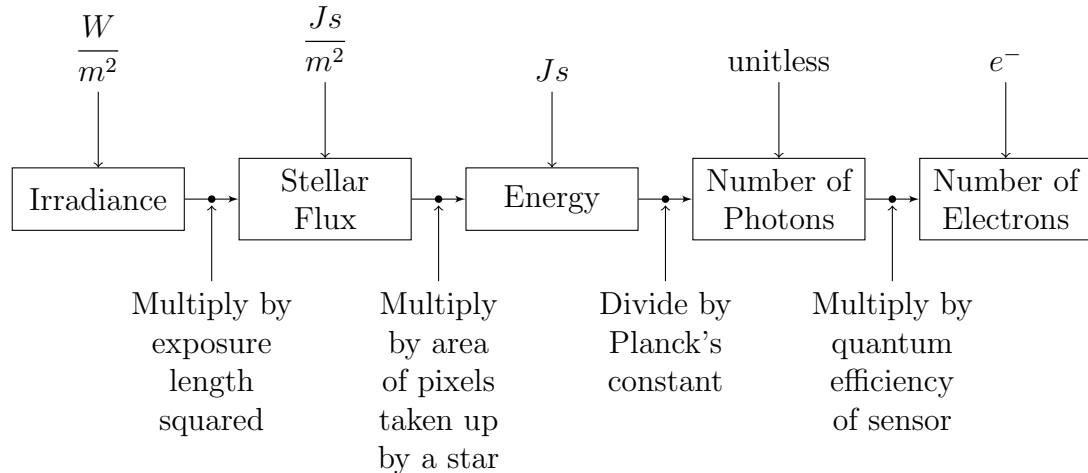


Figure 2.4. A block diagram showing the conversion process from flux density to the associated number of electrons.

The uncorrelated Gaussian function is given as

$$f(x, y) = A \exp \left( - \left( \frac{(x - u)^2}{2\sigma_x^2} + \frac{(y - v)^2}{2\sigma_y^2} \right) \right), \quad (2.16)$$

where  $A$  is the amplitude, or height, of the Gaussian function,  $u$  and  $v$  are the pixel locations found from Eq. (2.14), and  $\sigma_x^2$  and  $\sigma_y^2$  are the variance of the Gaussian in the  $x$  and  $y$  directions, respectively. What is intuitive about the uncorrelated Gaussian is the variances. Changing them allows for a more unique “shape” to the Gaussian, and in this case to the spread of pixels for a star. In this application the variances  $\sigma_x^2$  and  $\sigma_y^2$  are kept equal, producing circular spreads for each star.

Normally, when discussing Gaussian functions, one is expressing the probability density function (pdf) of a normally distributed variable. However, here the function is not assumed to be a pdf. The total “volume” under the function could be considered the total probability of a pdf if it equaled one, but here, it is found to equal the total number of electrons for a specific star. With this, the amplitude  $A$  of the Gaussian function can be found if the volume is explicitly solved for. To do so,



the volume must be calculated via

$$V = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) dx dy. \quad (2.17)$$

Turning attention to what is known as the Gaussian integral,

$$\int_{-\infty}^{\infty} e^{-x^2} dx = \sqrt{\pi},$$

it can be shown that

$$\int_{-\infty}^{\infty} ae^{-\frac{(x-b)^2}{2c^2}} dx = ac\sqrt{2\pi}.$$

Therefore,  $V$  can be solved by

$$\begin{aligned} V &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A \exp\left(-\left(\frac{(x-x_0)^2}{2\sigma_x^2} + \frac{(y-y_0)^2}{2\sigma_y^2}\right)\right) dx dy \\ &= A \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \exp\left(-\frac{(x-x_0)^2}{2\sigma_x^2}\right) \exp\left(-\frac{(y-y_0)^2}{2\sigma_y^2}\right) dx dy \\ &= \sqrt{2\pi} A \sigma_x \int_{-\infty}^{\infty} \exp\left(-\frac{(y-y_0)^2}{2\sigma_y^2}\right) dy \\ &= 2\pi A \sigma_x \sigma_y. \end{aligned}$$

Finally, solving for  $A$  gives

$$A = \frac{V}{2\pi\sigma_x\sigma_y}. \quad (2.18)$$

The stars are all kept to have a circular spread thus  $\sigma_x = \sigma_y = \sigma$ . Since the number of electrons for each star is assumed to be the total volume under their Gaussian function,  $V$  is different for each star. The standard deviation of the Gaussian is determined by sensor and optic parameters and is kept constant for each star regardless

of brightness. Each star's distribution is calculated in an  $8 \times 8$  buffer matrix, which is then added to the image as a whole, with its center placed the  $(u, v)$  pixel found by Eq. (2.14). Due to the nature of Gaussian functions, which asymptotically approach zero, the buffer was chosen to be this size in order to encompass as much of the star's pixel spread as possible without having to evaluate the function over the entire image. Utilizing Eqs. (2.16) and (2.18), the buffer matrix for this star is calculated and illustrated in Figure 2.5. Here, in Figure 2.6, the brightest star, Figure 2.5, is located at the point  $(113.01, 332.46)$  and thus the buffer matrix's center is placed at that location.

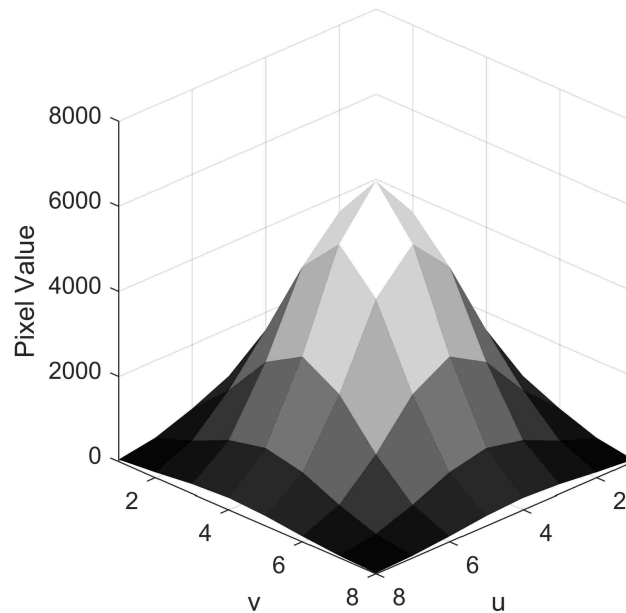


Figure 2.5. The pixel spread for the brightest star in Figure 2.6.

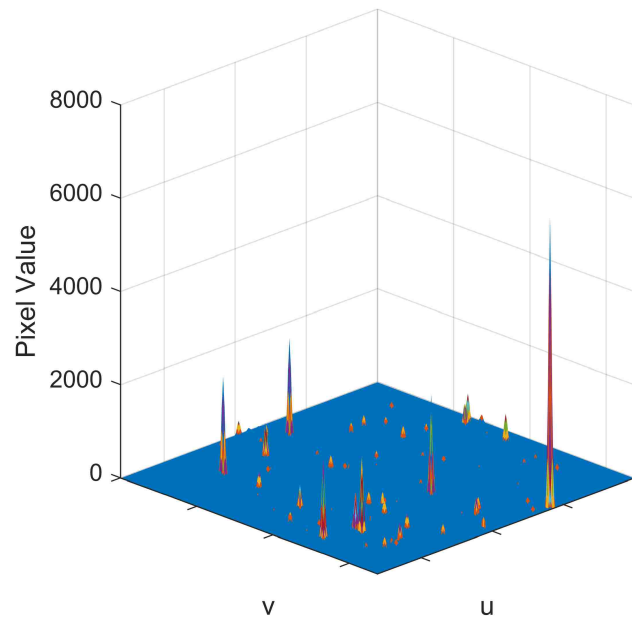


Figure 2.6. Illustrated here are each of the Gaussian functions that form the pixel spreads for each star in the FOV.

## 2.2. STAR TRACKER NOISE CHARACTERIZATION

**2.2.1. Background and Motivation.** Every image taken by any camera will have some inherent errors due to the nature of charge coupled devices (CCDs). These sensors work by exposing the photodetectors for a duration of time, called the exposure time, which allows photons to be collected and subsequently converted to an electrical charge. Once compiled, the accumulated charge is sequentially transferred into the two-dimensional CCD arrays and finally sent to the chip level output amplifier, where it is then converted into a voltage signal. With several steps between initial exposure to a voltage output, noise will arise from every aspect of the process. Accurately quantifying the noise a star tracker might encounter in space, and subsequently reproducing it, allows for any ground application of the image processing to be more representative of what is expected to occur during flight. Typical CCD noise can be broken up into three broad categories with their overall noise category

being general detector noise. The three basic groups of CCD errors come from photon noise, electron noise, and analog-to-digital (A/D) noise.

**2.2.2. Noise Addition Process.** Seeing noise in images and photos is quite common. Typically, noise isn't noticed by the human eye unless it is extreme and makes the image unrecognizable. However, to image processing software, subtle noise can cause issues with the specific processing the image is undergoing. Humans can easily see and read an image even with a high noise level, but common issues arise when there is blurring or an unwanted defocussing of the lens, which allows for images to become distorted. More subtle noise can affect image processing, which causes misreading of words, mis-detection of features, and in most star tracker applications, inaccurate centroiding.

Starting with a blank image,  $\mathbf{I}_0$ , adding noise is straightforward. The image starts as an image of solid black where every pixel is at zero electrons. Noise is then formed one at a time. First the zodiacal light is added to a blank image to form  $\mathbf{I}_{\text{zod}}$ . From here, dark current, pixel response non-uniformity, shot noise, and read noise form their own separate noise image. These images are then summed together and added to the original noiseless image,  $\mathbf{I}$ , by

$$\mathbf{I}_f = \mathbf{I} + \mathbf{I}_{\text{zod}} + \mathbf{I}_{\text{DC}} + \mathbf{I}_{\text{PRNU}} + \mathbf{I}_{\text{SN}} + \mathbf{I}_{\text{RN}}, \quad (2.19)$$

where  $\mathbf{I}_f$  is the final image. To account for saturation, the image is leveled out at the pixel well saturation level; therefore, any pixel value greater than this value is truncated and assigned the max value. The image is finally converted to a digital resolution of  $\mu = 64$  bits, which accounts for the (A/D) noise.

**2.2.3. Photon Noise.** Charged coupled devices work by exposing a sensor to light for a limited amount of time, and during that time, photons are collected on the capacitor array. These capacitor cells act like a reservoir, with each photon

striking the sensor and producing an electron. Over the exposure time, the sensor collects the electrons produced. During this process, photon noise is generated from the exposure length, line scan aberrations, lens vignetting loss and throughput, as well as stray light.

The exposure, or integration, length is the amount of time the CCD sensor is exposed to light, similar to a camera's shutter speed. A longer exposure time will result in more photons hitting the sensor and thus a brighter image, with the caveat of possibly over-saturating the pixels. Certain detectors have what is called rolling shutter, meaning that the image is read out one row of pixels at a time. This time delay between the first and last rows can have a negative effect if an imaged scene is moving quickly, which is referred to as line scan aberration. Lens vignetting and throughput refers to noise produced when light travels through the thinner and thicker, edges and center, parts of the camera lens. Stray light is simply the light that is reflected off other bodies in space such as the Earth, the Moon, parts of the spacecraft, or even other spacecraft in orbit.

For this work, the goal is to develop the software for a slow spinning spacecraft. Therefore, line scan aberration errors are not considered. As well, lens vignetting and throughput noise can be modeled in the calculations for the total number of photons collected in a pixel. However, they are closely related to optical errors and are not considered in this work.

Stray light is categorized as all unwanted light that strikes the sensor when it is exposed. A main source of stray light in space originates from the zodiacal dust cloud. Zodiac light is sunlight that has been diffused by small interplanetary dust particles. When attempting to measure and detect extremely faint stars, magnitude 15 and higher, accurate knowledge of the zodiacal background noise is imperative. A major component of the expected noise from stellar photometry comes from this zodiacal light. Fortunately, to a first-order approximation, the zodiacal cloud is uni-

form and consistent and only changes with a spacecraft's pose rather than with time or the measured spectral wavelength [23]. Levasseur-Regourd and Dumont have done extensive research into accurately mapping the zodiacal light and have produced tables that give brightness values for varying helioecliptic positions [24]. Zodiacal light is still quite faint when compared to most stars seen by typical star trackers; however, when pointing close towards the Sun, this noise increases. Assigning the value for the zodiacal light magnitude as  $\mathcal{J}_{\text{zod}}(\lambda, \beta)$  [8], the corresponding error is found to be

$$n_{\text{zod}} = 5s^2 \mathcal{J}_{\text{zod}}(\lambda, \beta) \times 10^{-15},$$

where  $s^2$  is the angular area of each pixel. The values given by Levasseur-Regourd and Dumont are in  $S_{10}(V)$ , or number of 10<sup>th</sup> visual magnitude stars of solar type per square degree, hence the changing in the order of magnitude. This error is different than the subsequent errors in the following sections because this noise is a magnitude value. It undergoes the same process to convert it to electrons and is then added to a blank image  $\mathbf{I}_0$  to form

$$\mathbf{I}_{\text{zod}} = \mathbf{I}_0 + n_{\text{zod}} \mathbb{1}_{m \times n},$$

where  $\mathbb{1}_{m \times n}$  is an  $m \times n$  matrix of ones used to add the noise to the entire image.

**2.2.4. Electron Noise.** By far, the most noise comes from electron noise. In the ideal case, one photon striking the sensor would result in exactly one electron being formed, giving way to the precise brightness of a pixel. Unfortunately, the one-to-one conversion does not happen due to the governing laws of physics as well as several other factors. The other sources of errors come from dark noise, pixel non-uniformity, shot noise, thermal noise, and read noise. Some of these sources can be modeled as a Poisson processes that depend on some rate, which is intuitive since cameras expose the sensor and experience a rate of arrival of the photons [25]. A

more convenient method is to use a Gaussian distribution for each of these making them relatively simple to combine. Given a Poisson distributed random variable, its distribution can be approximated as a Gaussian distribution if enough samples of the random variable's pdf are present. These noise values change depending on the sensor chosen, and a majority of the information regarding the noise values can be found off the sensors data sheet.

**2.2.4.1. Dark current.** Space grade cameras and telescope sensors frequently run into problems with thermal control. Having a “hot” camera can cause what is known as dark current, where, as the sensor heats up, thermally generated electrons build up and begin to saturate the pixels. Saturated pixels can quickly ruin centroiding of stars due to a star being too bright or even misshapen, and as the sensor heats up, the error increases accordingly. Integration time has a large effect on dark current. Lengthening integration time causes the sensor to increase in temperature by keeping it exposed to light longer. With this, the dark current noise is quantified as

$$\mathbf{I}_{DC} = \mathbf{I}_0 + n_{DC}\tau\mathbb{1}_{m \times n} + \sqrt{n_{DC}\tau}\mathbf{W}_{DC},$$

where  $\mathbf{I}_{DC}$  is the noise image for the dark current,  $n_{DC}$  is the mean noise of the dark current taken from the sensor data sheet,  $\tau$  is the exposure time, and  $\mathbf{W}_{DC}$  is an  $m \times n$  matrix of random samples of the normal distribution whose mean is zero and standard deviation is one,  $\mathcal{N}(0, 1)$ , with  $m \times n$  being the size of the image  $\mathbf{I}$ . The  $\mathbb{1}_{m \times n}$  is an  $m \times n$  matrix of ones used to add an overall bias to the image due to the dark current. Adding the third term ensures a more accurate representation of how dark current actually works since each pixel is independently affected by this noise.

**2.2.4.2. Pixel response non-uniformity.** As in any manufacturing process, items are constructed within an accepted machine tolerance. These manufac-

turing variations, when it comes to CCD sensors, can cause pixel geometries to vary from one another in the substrate material and microlens. Since pixel response non-uniformity (PRNU) occurs due to the physical characteristics of a sensor, it is impossible to completely eliminate and therefore is considered a standard property of sensors. PRNU is modeled as [26]

$$\mathbf{I}_{\text{PRNU}} = \mathbf{I}_0 + n_{\text{PRNU}} \mathbf{W}_{\text{PRNU}},$$

where  $\mathbf{I}_{\text{PRNU}}$  is the noise image for pixel response non-uniformity,  $\mathbf{W}_{\text{PRNU}}$  is a matrix of random samples of the normal distribution  $\mathcal{N}(0, 1)$ , and  $n_{\text{PRNU}}$  is the noise due to pixel response non-uniformity, which is found on the sensor data sheet.

**2.2.4.3. Shot noise.** Due to the underlying nature of photons, each photon that arrives to the sensor is its own event and cannot be precisely predicted. The fluctuating arrival of photons to the sensor is called shot noise and is fundamentally a Poisson distribution given as [27]

$$p_{\text{SN}}(N) = \frac{(\bar{N})^N e^{-\bar{N}}}{N!},$$

where  $N$  is the total number of photons collected and  $\bar{N}$  is the average of those values across the image. Conveniently, the mean,  $\mu_{\text{SN}}$ , and the variance,  $\sigma_{\text{SN}}^2$ , for a Poisson distribution are the same values. Using this along with the central limit theorem, a sum of independent and identically distributed random variables tend to become Gaussian as the number of samples increases. Therefore the shot noise is modeled as

$$\mathbf{I}_{\text{SN}} = \mathbf{I}_0 + \mathbf{n}_{\text{SN}} \mathbf{W}_{\text{SN}},$$

where  $\mathbf{I}_{\text{SN}}$  is the noise image for shot noise,  $\mathbf{W}_{\text{SN}}$  is a matrix of random samples of the normal distribution  $\mathcal{N}(0, 1)$ , and  $\mathbf{n}_{\text{SN}}$  is the shot noise at each pixel since shot noise



affects each pixel individually. Shot noise is determined to be the standard deviation of photons per pixel and equals the square-root of the pixel value at each pixel giving

$$n_{\text{SN}} = \sqrt{I},$$

where  $I$  is the image. It should be noted that dark current, shot noise, and read noise allow for the possibility of a negative noise value. A negative noise value indicates that a pixels value would be decreased and corresponds to a noise source causing the sensor to absorb less electrons.

**2.2.4.4. Read noise.** Once the sensor is exposed to light, photons start collecting in the sensor wells, producing electrons. Afterwards, these electrons are converted to a voltage that is proportional to the number of electrons in a well. This voltage is multiplied by the sensor amplifier gain, then digitized by an analog-to-digital converter. The digital number represents the number of photons/electrons from the sensor pixel, and the number of analog-to-digital units (ADUs) is often referred to as the raw units or data numbers (DN). The proportionality between raw data and photon/electron count is called the gain. The signal processing chain, in which read noise corrupts the data, starts with sensor read out then goes to the amplifier gain and is finally digitized to produce the raw data. During this process, voltage fluctuations are the source of read noise. Certain forms of read noise are frequency and process dependent, but manufacturers generally provide a mean read noise for a product. With this value, the read noise is added to the image by

$$I_{\text{RN}} = I_0 + n_{\text{RN}} \mathbb{1}_{m \times n},$$

where  $I_{\text{RN}}$  is the noise image for read noise and  $n_{\text{RN}}$  is the mean read noise. This noise is uniform across all pixels in an image [26], thus the noise is multiplied by  $\mathbb{1}_{m \times n}$ .

**2.2.4.5. Amplifier noise.** In order to convert the photons/electrons of a pixel into a voltage measurement, the sensor uses a capacitor and amplifier. Before measuring the pixel's charge, the capacitor must be reset to a base or reference level. Reset noise, or kTC noise, is induced during this process in the thermal variations of the reset transistor. The charge of the noise is given by

$$q_n = \sqrt{kTC},$$

where  $k = 1.38064852 \times 10^{-23}$  J/K is Boltzmann's constant,  $T$  is the operating temperature, and  $C$  is the capacitance. Manufacturers often report noise values as a number of electrons root mean square (rms), meaning that a number of rms electrons is the standard deviation around the mean pixel value. Reset noise can be quite large, such as 50 rms electrons, whereas typical read noise is on the order of 9 rms electrons [25].

A second source of amplifier noise can arise in the form of flicker noise, or  $1/f$  (one-over-f) noise, which appears from the pixel sampling rate of the amplifier. Lower frequencies cause more flicker noise, but at high frequencies of pixel read out, around 10 MHz, the read noise floor can be determined by just the  $1/f$  noise. The reason for flicker noise is due to the time constants of the interlaced circuitry in the image sensor silicon that turn on and off during read out, effectively trapping electrons in the sensor [28].

Due to the consistent nature of amplifier noise across all current imaging devices, removing it has become part of the device itself. Typical sensors employ what is called correlated double sampling, or CDS. To remove this noise, the pixel read out process is done twice. The first time while the pixel is still in the reset state, and the second once the charge has been transferred to the amplifier. Comparing the separate outputs allows the sensor to produce a less noisy output.

**2.2.5. Analog-To-Digital Noise.** When raw data is being converted to digital units, the raw data must be rounded to the nearest integer. This integer relies on the digital resolution of the system. Quantization error and saturation error are what primarily make up analog-to-digital (A/D) noise. Saturation error is when the brightest pixel, the pixel with the highest electron count, is converted to the highest level of digital resolution. The lowest electron count would be designated as 0, leaving the highest equal to

$$\text{maximum pixel value} = 2^\mu - 1,$$

where  $\mu$  is the digital resolution, which can be anywhere from 8 bit to 64 bit for typical star tracker cameras. Next, quantization noise is taken into account when rounding to the nearest integer. To better understand and visualize quantization error, recall Figure 2.5. In that figure, the star's distribution is continuous and allows for a more precise value at each pixel. Due to quantization, Figure 2.5 transforms into Figure 2.7, where the values of each pixel is rounded to the nearest integer.

### 2.3. FINAL IMAGE COMPARISON

The method used to form a virtual image of what a star tracker may encounter in space is presented. The process starts with a reference star catalog and a chosen sensor. From here, the pixel locations are determined for the given attitude and FOV of the camera with respect to the inertial frame used in the star catalog. Once the pixel locations are known, the corresponding pixel spread is found for each star, which depends on the star's magnitude as well as sensor properties. Lastly, the image is sent through the error addition process to produce the final image. The image without errors will appear to be the same as the final image to the human eye, however the overall signal of these images will be quite different. Examining Figure

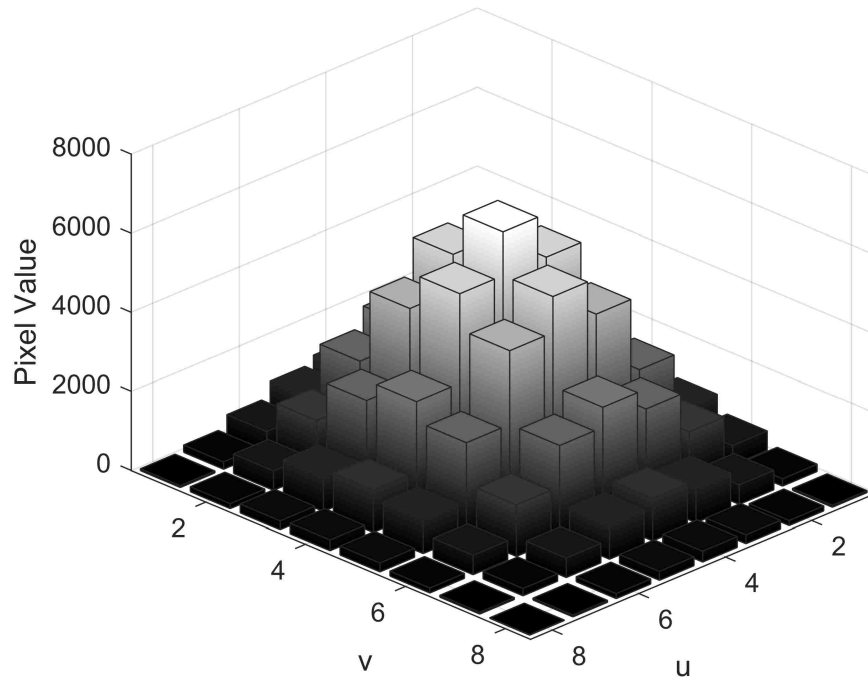


Figure 2.7. An illustration of the result from quantization error applied to Figure 2.5.

2.8a and Figure 2.8b shows that the differences in images before and after errors are introduced is near impossible to see from the naked eye. However, when looking at their respective signals in Figure 2.9, it is clear that in both the rows and columns of the image, the overall signal, and thus the overall image brightness, in Figure 2.8b is larger than that of Figure 2.8a. The comparison here is to show that the second image has a high level of noise present, despite it visually looking the same as the figure without noise.



(a) An example of a final image where no errors are present.



(b) The same image as in Figure 2.8a, except the errors determined in Section 2.2, are present.

Figure 2.8. Final image comparison.

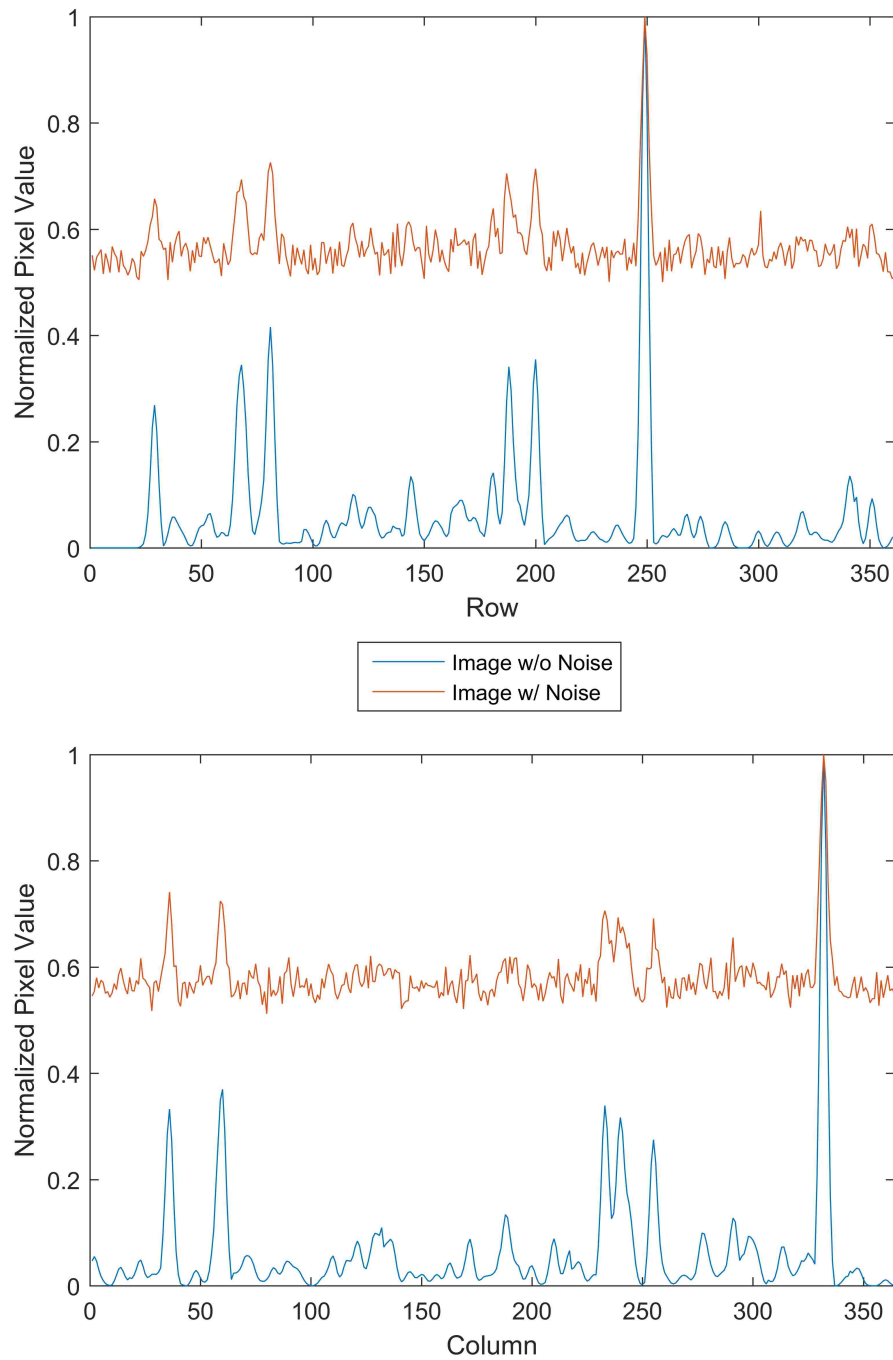


Figure 2.9. A comparison between image values, the blue line being the signal from Figure 2.8a and the red being the signal from Figure 2.8b.

### 3. IMAGE PROCESSING

A single image captured by the star camera can contain enough information to accurately determine the spacecraft's attitude. Therefore, these images must be handled appropriately to insure as much data are available for the centroiding algorithm. These images can be vastly different depending on the current mission parameters. Certain spacecraft require they spin in order to acquire data or even for stability, such as the latter Pioneer missions by NASA [2]. These spacecraft will require slightly different processing of the images than a non-spinning, or slow spinning spacecraft. Image processing techniques have improved over the years to allow for precise noise reduction and feature extraction. The methods used in this work are straightforward and do not concern themselves with producing real-time measurements required by spinning spacecraft. Instead, a robust algorithm is implemented with the understanding that they would be utilized for slow spinning spacecraft.

All images captured have some form of noise in them. Reducing the noise level without reducing the quality of the features in an image is key. Certain techniques work better depending on the scenario and what type of noise is expected. In general, the types of noise experienced for a star tracker image can be treated as uniform or Gaussian distributed due to the nature of CCD sensors. Straightforward methods of reducing image noise include smoothing filters, which typically set each pixel to an average value depending on the surrounding pixel values, and nonlinear filters, which can utilize the median pixel value instead of the average [29]. This approach works well to reduce salt-and-pepper noise and causes relatively little blurring. Since the noise is approximately uniform, a simple approach is applied here to determine the background noise for the images.

Centroiding techniques are of great importance and range anywhere from a simple center of mass calculation, Newton-Gauss linearization methods, and two-

dimensional Gaussian estimation [3, 8, 30]. In this work, a different technique is implemented to obtain the necessary centroids. The approach minimizes the Kullback-Leibler divergence between a Gaussian model and a uniform mixture model. If an array of pixels is assumed to be a two-dimensional uniform mixture, then it is shown that the mean and covariance of the mixture is the same mean and covariance of a Gaussian which minimizes the KL-divergence [13]. This method is favorable to the standard center of mass technique since it non-iteratively provides an optimal solution for the centroid.

### 3.1. NOISE REDUCTION

Since the method for matching stars to the catalog depends only on the relative brightness of the stars in the image, subtracting a constant from the image will not distort the pixel values individually, but rather as a whole. Other methods may cause inconsistent reduction of parts of the image. Allowing irregular noise reduction over an image may cause the image processing technique to select an incorrect star.

To find the background noise, a histogram is used to find where the grey level cut-off is. As one might expect, the majority of values are towards zero due to the nature of the images captured. The plot in Figure 3.1 utilizes a log scale to show the large number of lower grey levels, but cuts off the higher grey levels due to their infrequency and in order to emphasize the amount of background noise. This background noise level is determined to be two times the grey level of the last bin with more than 100 counts. For example, if the last bin with less than 100 counts is at a grey level of 500, then the background is determined to be a grey level of 1000. The background noise level is subtracted from the image as a whole and then passed to the centroiding algorithm.



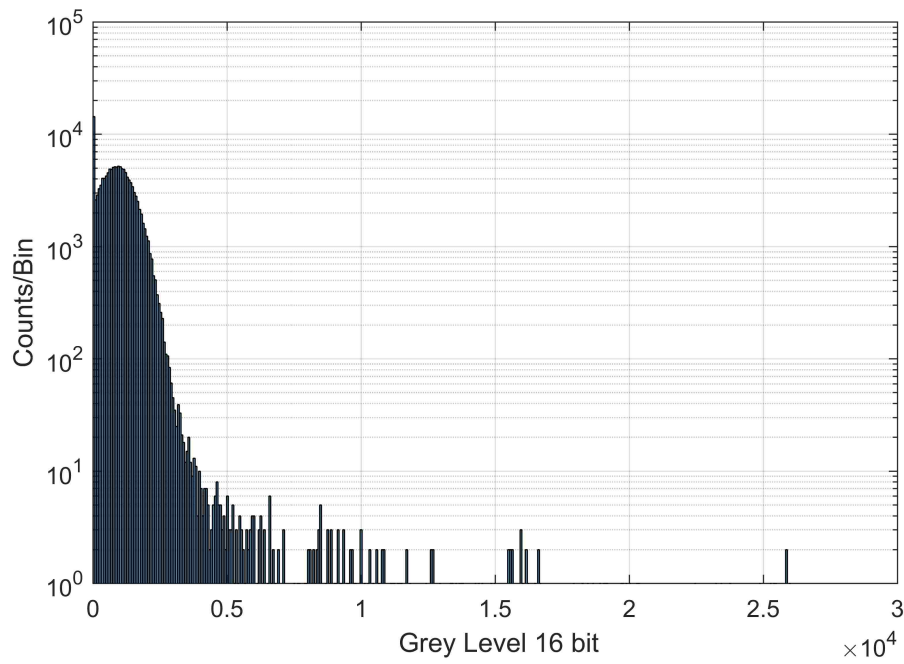


Figure 3.1. A histogram of Figure 2.8b.

### 3.2. THRESHOLDING AND STAR FINDING ALGORITHM

To reiterate, the output of the image processing scheme is a set of three centroids that correspond to the three brightest stars in an image. The thresholding algorithm is the first distinguishing feature of the star tracker algorithm. Finding three stars allows for the star-set triangle to be formed, taking the stars' pixel locations as the vertices. It is common that star trackers will attempt to find the brighter stars in an image, since they are easier to separate from the background. Due to this approach, only the relative brightness of stars within the image are necessary to determine, and not the actual brightness of the stars found.

To start, the threshold is set to 75% of the maximum pixel value in an image. Next, the centroiding algorithm attempts to find the center of every star above that threshold. If the algorithm detects a pixel above the threshold, such as the pixel denoted by a the green “+” in Figure 3.2, the pixel is considered part of a candidate

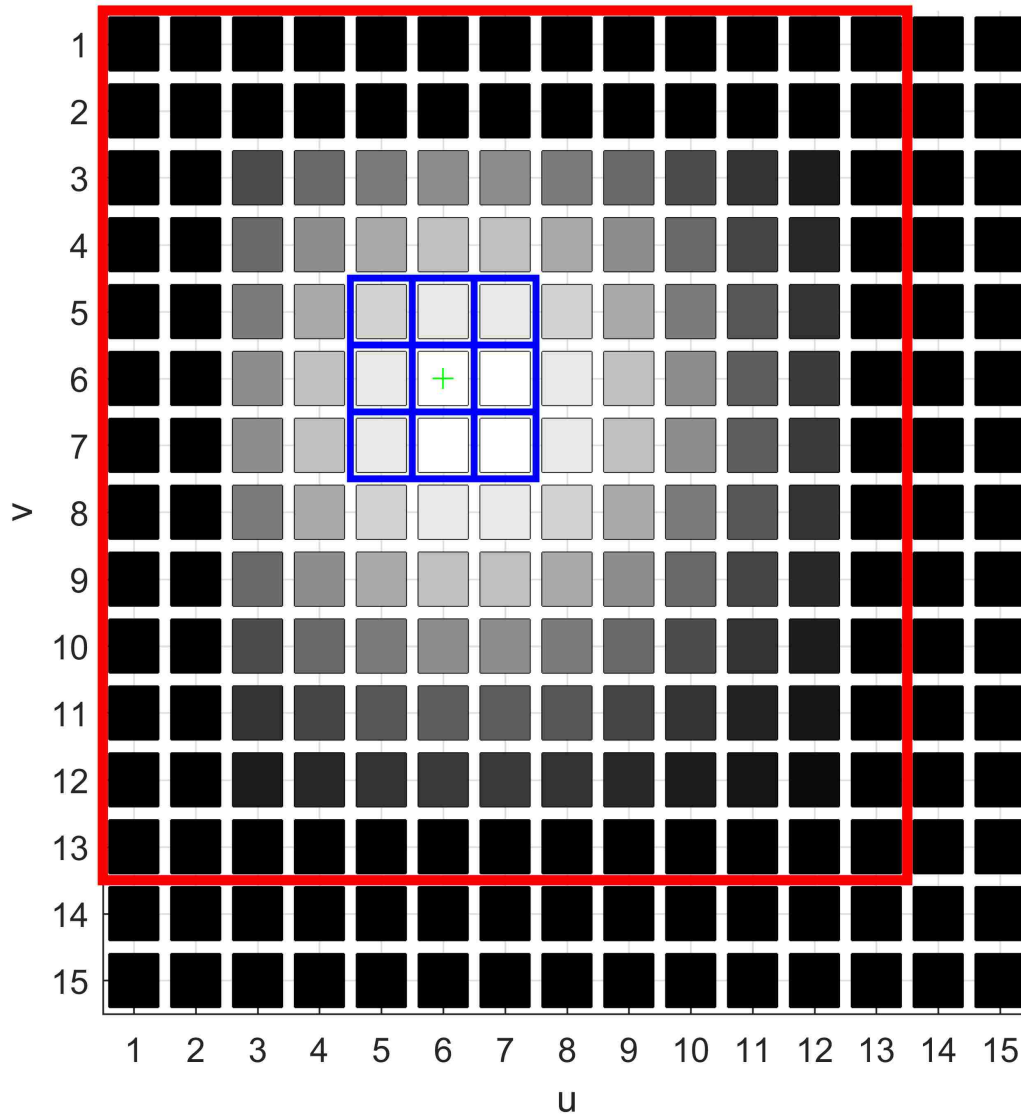


Figure 3.2. An illustration of the star detection method. The green “+” signifies the pixel above the threshold value. The blue is the  $3 \times 3$  search template, and the red indicates the border of the region of interest.

star. Then, a  $3 \times 3$  search template is then formed around this pixel, outlined in blue. The mean value of this grid is calculated, and if this value is above the threshold as well, the algorithm assumes it has found a star. Since the image is searched from left to right, top to bottom, it is assumed that the search template will locate a star from the top left. Using this information, the region of interest (ROI), shown in red in

Figure 3.2, around the star is formed such that it encompasses all of the star, adding more space to the bottom right of the ROI.

With the region containing one star known, the algorithm subtracts off the average ROI border pixel value to further reduce any noise from neighboring stars that may skew the centroiding process. With the region of interest finalized, it is then passed to the centroiding algorithm. After finding the centroids of stars above the threshold, the algorithm performs a check to see if there are three sets of centroid coordinates. If there are fewer than three, or even zero, it then iteratively lowers the threshold by 5% until only three stars are found. In the case where lowering the threshold allows for four stars to be found instead of three, the algorithm fine tunes the threshold value by 0.1% until there are the correct number of stars.

### 3.3. CENTROIDING

**3.3.1. Center of Mass.** Calculating the center of mass for the region of interest is simple and fast. To find the center  $\mathbf{m}_{\text{COM}}$ , the average position of the pixels are found, each weighted by their specific brightnesses

$$\mathbf{m}_{\text{COM}} = \frac{\sum_{i=1}^n J_i \mathbf{x}_i}{\sum_{i=1}^n J_i} + 0.5, \quad (3.1)$$

where  $i$  is the index of each pixel,  $n$  is the total number of pixels in the ROI,  $\mathbf{x}_i$  is the pixel coordinate, and  $J_i$  is the intensity at each pixel. The 0.5 that is added allows for transforming the coordinate system from the pixel coordinates to the Cartesian  $x$  and  $y$  values. The difference in coordinates is due to CCD arrays starting at an origin of  $(0, 0)$ , but the star camera software starts indexing at  $(1, 1)$ . The resulting centroid of the brightest star in Figure 3.3 is shown in Figures 3.4a and 3.4b.

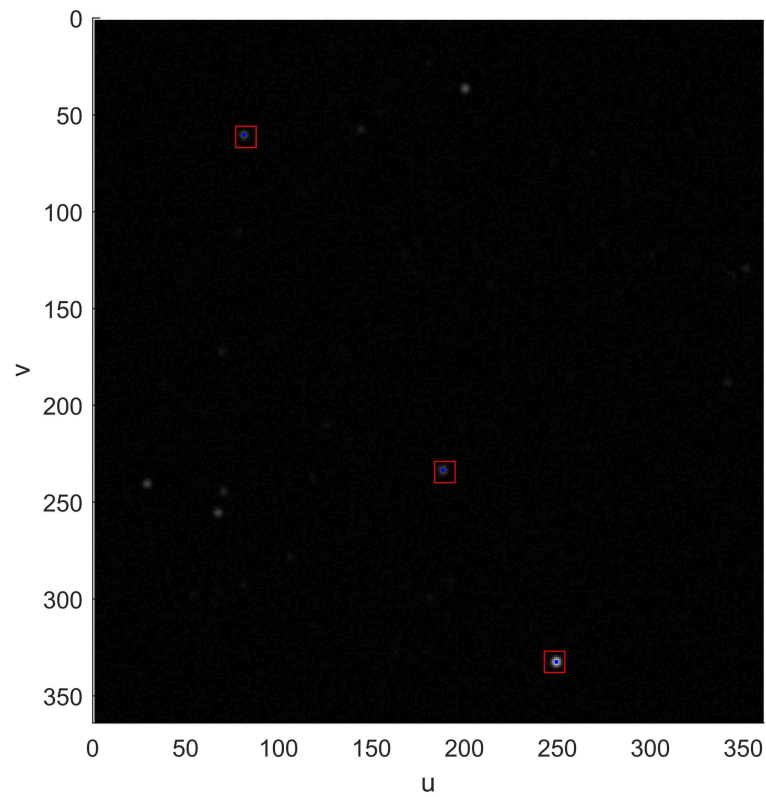
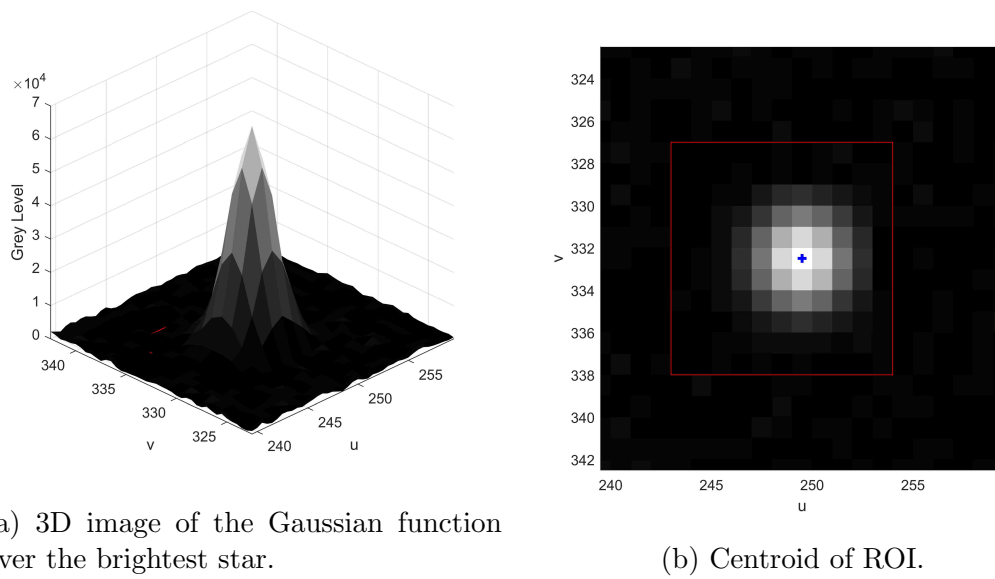


Figure 3.3. Illustrating the region of interest via the red boxes.



(a) 3D image of the Gaussian function over the brightest star.

(b) Centroid of ROI.

Figure 3.4. Zoomed in region of interest on the image's brightest star. The red box is the ROI boundary, and the blue cross signifies the centroid.

**3.3.2. Uniform Mixture Model Approximation.** An alternative method of calculating the centroid is considered here and derived for the one-dimensional case. The method developed allows for non-iteratively determining a Gaussian mean and covariance that best fit, with respect to the Kullback-Leibler divergence, a uniform mixture model pdf. Consider the non-negative functions that may look like

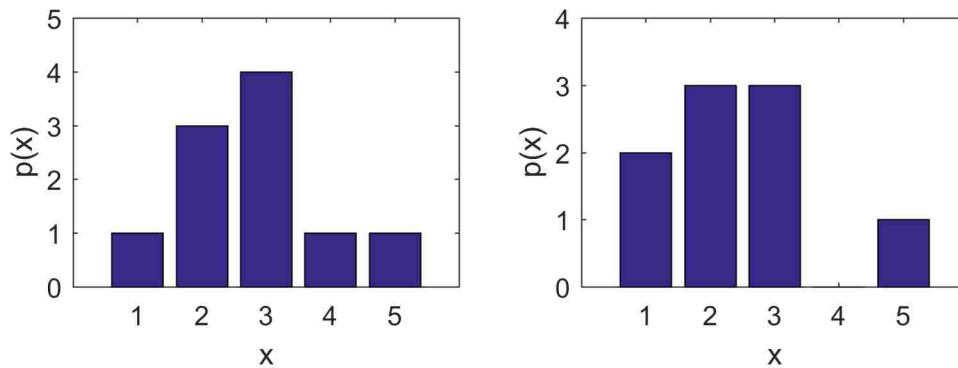


Figure 3.5. Examples of non-negative, univariate, uniform mixture model pdfs.

and have the form

$$p(x) = \sum_{i=1}^L w_i \mathcal{U}(a_i, b_i) \quad \text{s.t.} \quad w_i \geq 0 \quad \text{and} \quad \sum_{i=1}^L w_i = 1$$

with  $\mathcal{U}(a_i, b_i)$  denoting the uniform distribution, which is given by

$$\mathcal{U}(a_i, b_i) = \begin{cases} \frac{1}{b_i - a_i} & a_i \leq x \leq b_i \\ 0 & \text{otherwise.} \end{cases}$$

If the weights  $w_i$  should not sum to unity, which may occur when dealing with an arbitrary positive function, they must be normalized first. Ultimately, there will exist a specific weight for each component of the mixture. Additionally, the support for

each component is known via the parameters  $a_i$  and  $b_i$ . Both  $a_i$  and  $b_i$  are natural representations of the  $i^{\text{th}}$  component, but it is sometimes more convenient to express  $\mathcal{U}(a_i, b_i)$  using its mean and variances, given by

$$m_i = \frac{a_i + b_i}{2} \quad (3.2)$$

$$v_i = \frac{1}{12}(b_i - a_i)^2. \quad (3.3)$$

Now, in order to find the Gaussian parameters that best describe this pdf, without iteration, consider the Kullback-Leibler divergence given by

$$D_{KL}[p||q] = \int p(x) \ln \frac{p(x)}{q(x)} dx,$$

where

$$p(x) = \sum_{i=1}^L w_i \mathcal{U}(a_i, b_i)$$

$$q(x) = \mathcal{N}(m, P) = \frac{1}{\sqrt{2\pi P}} \exp \left\{ -\frac{1}{2} \frac{(x - m)^2}{P} \right\}.$$

First, the KL divergence is computed for this case. Note that the divergence can be written as

$$D_{KL}[p||q] = \int p(x) \ln p(x) dx - \int p(x) \ln q(x) dx.$$

Next,  $p(x)$  and  $q(x)$  are substituted, yielding

$$D_{KL}[p||q] = \int \sum_{i=1}^L w_i \mathcal{U}(a_i, b_i) \ln \sum_{i=1}^L w_i \mathcal{U}(a_i, b_i) dx - \int \sum_{i=1}^L w_i \mathcal{U}(a_i, b_i) \ln \mathcal{N}(m, P) dx.$$

Taking a look at the first term, there is no dependency on the mean or variance of the Gaussian; therefore, this integral is denoted by  $-H$ , giving

$$D_{KL}[p||q] = -H - \int \sum_{i=1}^L w_i \mathcal{U}(a_i, b_i) \ln \mathcal{N}(m, P) dx.$$

Moving the summation out of the integral, and substituting in the pdfs

$$\begin{aligned} &= -H - \sum_{i=1}^L w_i \int \mathcal{U}(a_i, b_i) \ln \mathcal{N}(m, P) dx \\ &= -H - \sum_{i=1}^L \frac{w_i}{b_i - a_i} \int_{a_i}^{b_i} \left[ -\frac{1}{2} \ln(2\pi P) - \frac{1}{2} \frac{1}{P} (x - m)^2 \right] dx. \end{aligned}$$

With this, the next step is to evaluate the integral, giving

$$\begin{aligned} D_{KL}[p||q] = -H + \sum_{i=1}^L \frac{w_i}{b_i - a_i} &\left\{ \frac{1}{2} (b_i - a_i) \ln(2\pi P) \right. \\ &\left. + \frac{1}{6} \frac{1}{P} \left[ (b_i^3 - a_i^3) - 3(b_i^2 - a_i^2)m + 3(b_i - a_i)m^2 \right] \right\}. \end{aligned} \quad (3.4)$$

With this, Eq. (3.4) calculates the KL-divergence between the uniform mixture model and a Gaussian distribution. The divergence holds for any Gaussian, and the divergence is zero if and only if  $p(x) = q(x)$ . As well, since  $D_{KL} \geq 0$ , it makes sense to choose the mean and covariance of the Gaussian that minimizes  $D_{KL}$ . If the input parameters of the mixture are known,  $a_i$  and  $b_i$ , then the cost function is chosen as

$$J(m, P) = D_{KL}[p||q]. \quad (3.5)$$

In order to find a candidate minimum, the first-order conditions for optimality [13],

$$\frac{\partial J}{\partial m} = 0 \quad \text{and} \quad \frac{\partial J}{\partial P} = 0$$

must be solved. However, finding a solution does not guarantee a minimizing solution. To show that the candidate minimum is in fact a minimum, the matrix of second derivatives must be positive definite [13], or more concisely,

$$\begin{bmatrix} \frac{\partial^2 J}{\partial m^2} & \frac{\partial^2 J}{\partial m \partial P} \\ \frac{\partial^2 J}{\partial m \partial P} & \frac{\partial^2 J}{\partial P^2} \end{bmatrix} > 0.$$

Considering the first-order conditions, and recalling that  $H$  is not dependent on  $m$  or  $P$ , the first derivative with respect to  $m$  is found to be

$$\begin{aligned} \frac{\partial J}{\partial m} &= \frac{\partial}{\partial m} \left\{ -H + \sum_{i=1}^L \frac{w_i}{b_i - a_i} \left\{ \frac{1}{2}(b_i - a_i) \ln(2\pi P) \right. \right. \\ &\quad \left. \left. + \frac{1}{6} \frac{1}{P} \left[ (b_i^3 - a_i^3) - 3(b_i^2 - a_i^2)m + 3(b_i - a_i)m^2 \right] \right\} \right\} \\ &= \frac{1}{P} \sum_{i=1}^L w_i \left[ m - \frac{1}{2}(a_i + b_i) \right]. \end{aligned}$$

After finding the derivative, it must be set equal to zero

$$0 = \frac{1}{P} \sum_{i=1}^L w_i \left[ m - \frac{1}{2}(a_i + b_i) \right],$$

and, while keeping in mind that the variance must be positive, the terms are shifted around to produce

$$\sum_{i=1}^L w_i m = \sum_{i=1}^L w_i \frac{a_i + b_i}{2}.$$



Recall that the weights  $a_i$ , for  $i = 1, \dots, n$ , are formed such that they sum to one, which allows for the mean to be solved such that

$$m = \sum_{i=1}^L w_i \frac{a_i + b_i}{2}.$$

Looking back at Eq. (3.2), the definition of the uniform mixture model pdf mean,

$$m = \sum_{i=1}^L w_i m_i, \quad (3.6)$$

it is clear that the sum of the individual means of the uniform mixture is that which allows for the first derivative with respect to  $m$  to be zero. Now, looking at the second first-order condition

$$\begin{aligned} \frac{\partial J}{\partial P} &= \frac{\partial}{\partial P} \left\{ -H + \sum_{i=1}^L \frac{w_i}{b_i - a_i} \left\{ \frac{1}{2}(b_i - a_i) \ln(2\pi P) + \right. \right. \\ &\quad \left. \left. \frac{1}{6} \frac{1}{P} [(b_i^3 - a_i^3) - 3(b_i^2 - a_i^2)m + 3(b_i - a_i)m^2] \right\} \right\} \\ &= \sum_{i=1}^L \frac{w_i}{b_i - a_i} \left\{ \frac{1}{2} \frac{1}{P} (b_i - a_i) - \frac{1}{6} \frac{1}{P^2} [(b_i^3 - a_i^3) - 3(b_i^2 - a_i^2)m + 3(b_i - a_i)m^2] \right\}. \end{aligned}$$

Setting this derivative to zero yields

$$\begin{aligned} 0 &= \sum_{i=1}^L \frac{w_i}{b_i - a_i} \left\{ \frac{1}{2} \frac{1}{P} (b_i - a_i) - \frac{1}{6} \frac{1}{P^2} [(b_i^3 - a_i^3) - 3(b_i^2 - a_i^2)m + 3(b_i - a_i)m^2] \right\} \\ \sum_{i=1}^L 3w_i P &= \sum_{i=1}^L \frac{w_i}{b_i - a_i} [(b_i^3 - a_i^3) - 3(b_i^2 - a_i^2)m + 3(b_i - a_i)m^2] \\ &= \sum_{i=1}^L w_i \frac{b_i^3 - a_i^3}{b_i - a_i} - 3m^2. \end{aligned}$$

Utilizing the requirements that the weights must sum to one, it is found that

$$P = \frac{1}{3} \sum_{i=1}^L w_i \frac{b_i^3 - a_i^3}{b_i - a_i} - m^2. \quad (3.7)$$

Working the fraction into a more convenient form gives

$$(b_i - a_i)^3 = b_i^3 - 3b_i^2a_i + 3b_ia_i^2 - a_i^3$$

such that

$$(b_i^3 - a_i^3) = (b_i - a_i)^3 + 3b_i^2a_i - 3b_ia_i^2 \quad (3.8)$$

$$(b_i^3 - a_i^3) = (b_i - a_i)^3 + 3b_ia_i(b_i - a_i). \quad (3.9)$$

Inserting Eq. (3.8) into Eq. (3.7) and simplifying gives way to

$$P = \sum_{i=1}^L w_i \left[ \frac{1}{12}(b_i - a_i)^2 + \left( \frac{a_i + b_i}{2} \right) \right] - m^2.$$

Finally, recalling the definitions of the component means and variances for a uniform mixture model, Eqs. (3.2) and (3.3), it shows that

$$P = \sum_{i=1}^L w_i (v_i + m_i^2) - m^2$$

or in another form

$$P = \sum_{i=1}^L w_i [v_i + (m_i - m)^2]. \quad (3.10)$$

Currently, the forms of  $m$  and  $P$  in Eqs. (3.6) and (3.10), respectively, are the forms that satisfy the first-order optimality conditions. However, it is unknown whether they will minimize the KL-divergence between the uniform mixture model and the

Gaussian distribution. Therefore, moving to the second derivatives, it can be shown that

$$\frac{\partial^2 J}{\partial m^2} = \frac{1}{P}, \quad (3.11)$$

$$\frac{\partial^2 J}{\partial m \partial P} = 0, \quad (3.12)$$

$$\frac{\partial^2 J}{\partial P^2} = -\frac{1}{2} \frac{1}{P^2} + \frac{1}{3} \frac{1}{P^3} \left[ \sum_{i=1}^L \frac{b_i^3 - a_i^3}{b_i - a_i} - 3m^2 \right]. \quad (3.13)$$

The term in square brackets is  $3P$  from Equation 3.7, and therefore

$$\begin{aligned} \frac{\partial^2 J}{\partial P^2} &= -\frac{1}{2} \frac{1}{P^2} + \frac{1}{3} \frac{1}{P^3} 3P \\ &= -\frac{1}{2} \frac{1}{P^2} + \frac{1}{P^2} \\ &= \frac{1}{2} \frac{1}{P^2}. \end{aligned}$$

Thus, the matrix of second derivatives is found to be

$$\begin{bmatrix} \frac{\partial^2 J}{\partial m^2} & \frac{\partial^2 J}{\partial m \partial P} \\ \frac{\partial^2 J}{\partial m \partial P} & \frac{\partial^2 J}{\partial P^2} \end{bmatrix} = \begin{bmatrix} \frac{1}{P} & 0 \\ 0 & \frac{1}{2} \frac{1}{P^2} \end{bmatrix} \quad (3.14)$$

which is positive definite as long as  $P > 0$ . Now it is clear that

$$m = \sum_{i=1}^L w_i m_i \quad (3.15)$$

$$P = \sum_{i=1}^L w_i [v_i + (m_i - m)^2] \quad (3.16)$$

are the mean and covariance of the Gaussian distribution that minimizes the KL-divergence between a uniform mixture model and a Gaussian distribution, provided that  $P > 0$ . Conveniently, these parameters which minimize the KL-divergence here

are the mean and covariance of the uniform mixture model. The mean of  $p(x)$  is

$$\begin{aligned}
 m &= \int_{-\infty}^{\infty} xp(x) dx \\
 &= \int_{a_i}^{b_i} x \sum_{i=1}^L w_i \frac{1}{b_i - a_i} dx \\
 &= \sum_{i=1}^L w_i \frac{(a_i + b_i)}{2} \\
 &= \sum_{i=1}^L w_i m_i
 \end{aligned}$$

Similarly, the variance of  $p(x)$  is

$$\begin{aligned}
 \Pi &= \int_{-\infty}^{\infty} (x - m)^2 p(x) dx \\
 &= \int_{a_i}^{b_i} x^2 \sum_{i=1}^L \frac{a_i + b_i}{2} dx - m^2 \\
 &= \frac{1}{3} \sum_{i=1}^L w_i \frac{b_i^3 - a_i^3}{b_i - a_i} - m^2 \\
 &= \frac{1}{3} \sum_{i=1}^L w_i \frac{b_i^3 - a_i^3}{b_i - a_i} - \left[ \sum_{i=1}^L w_i m_i \right]^2.
 \end{aligned}$$

This equation can be manipulated into a similar form shown previously in Equation 3.10. Thus, it has been proven that the mean and covariance of the Gaussian distribution which minimize the KL-divergence are exactly the mean and covariance of the uniform mixture model. However, the results have only been proven for the univariate case.

A similar derivation can be done for the multivariate case as the univariate; however, with five parameters to minimize instead of two, it is felt that there will be little information gained by this. Instead, to demonstrate the multivariate case,

consider the two-dimensional uniform mixture model

$$\mathcal{U}(a_i, b_i, c_i, d_i) = \begin{cases} \frac{1}{b_i - a_i} \frac{1}{d_i - c_i} & a_i \leq x \leq b_i, c_i \leq y \leq d_i \\ 0 & \text{otherwise.} \end{cases}$$

The parameters  $(a_i, b_i)$  are the parameters associated to the “columns” of the pdf, and  $(c_i, d_i)$  are those associated with the “rows.” Utilizing the Kullback-Leibler divergence,  $p(\mathbf{x})$  and  $q(\mathbf{x})$  are found to be

$$p(\mathbf{x}) = \sum_{i=1}^L w_i \mathcal{U}(a_i, b_i, c_i, d_i)$$

$$q(\mathbf{x}) = \mathcal{N}(\mathbf{m}, \mathbf{P}) = \frac{1}{\sqrt{2\pi|\mathbf{P}|}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \mathbf{m})^T \mathbf{P}^{-1} (\mathbf{x} - \mathbf{m}) \right\}.$$

To build the pixel array, and allowing for an intuitive approach, a Gaussian pdf is formed with mean and covariance

$$\mathbf{m} = \begin{bmatrix} 5 \\ 5 \end{bmatrix} \quad \text{and} \quad \mathbf{P} = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}.$$

shown in Figure 3.6. Next, a small amount of uniform noise,  $v = 0.005 \times \mathcal{U}(0, 1, 0, 1)$ , is added to each pixel individually to form Figure 3.7.

First, the multivariate uniform mixture parameters

$$\begin{aligned} \mathbf{a} &= [a_1, \dots, a_L], & \mathbf{b} &= [b_1, \dots, b_L], \\ \mathbf{c} &= [c_1, \dots, c_L]^T, & \mathbf{d} &= [d_1, \dots, d_L]^T \end{aligned}$$

must be determined. The parameters  $\mathbf{c}$  and  $\mathbf{d}$  are transposed to indicate they are the vertical parameters of the pdf. If the pdf is considered to be an array of pixels, then

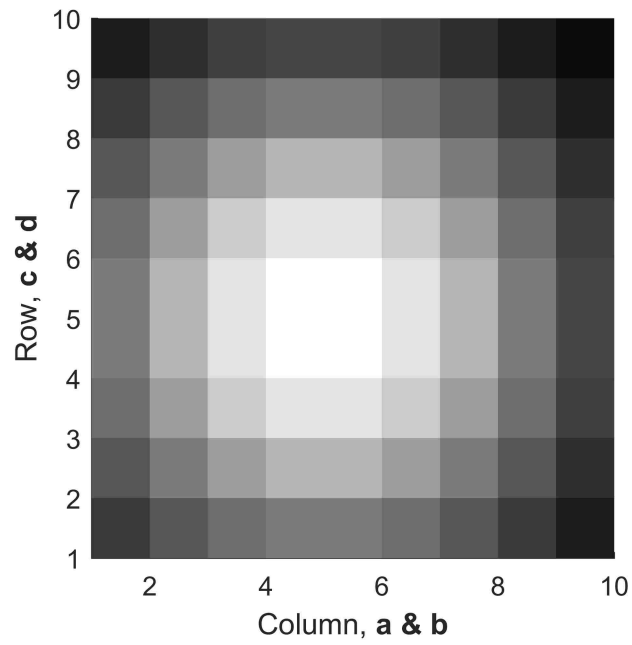


Figure 3.6. An uncorrelated Gaussian pdf without noise.

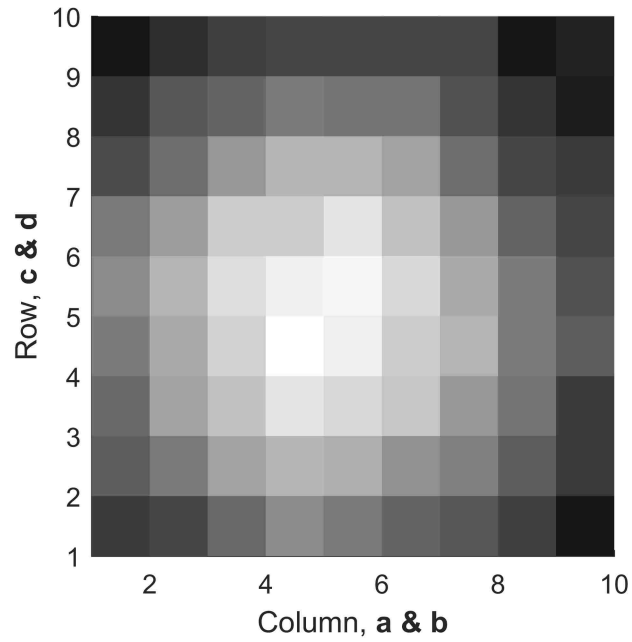


Figure 3.7. The Gaussian pdf with minimal noise added.

the uniform mixture parameters are the bounds of each pixel. These are found to be

$$\begin{aligned} \mathbf{a} &= [1, 2, 3, 4, 5, 6, 7, 8, 9] & \mathbf{b} &= [2, 3, 4, 5, 6, 7, 8, 9, 10] \\ \mathbf{c} &= [1, 2, 3, 4, 5, 6, 7, 8, 9]^T & \mathbf{d} &= [2, 3, 4, 5, 6, 7, 8, 9, 10]^T \end{aligned}$$

The weights,  $w_{i,j}$ , for the pixels correspond to the pixel's brightness, or the grey value, where solid black is zero and solid white is one. After normalizing the weights such that they sum to one, the mean is found by

$$\mathbf{m}_{\text{UM}} = \left[ \sum_{i=1}^9 \sum_{j=1}^9 w_{i,j} \frac{a_i + b_i}{2} \quad \sum_{i=1}^9 \sum_{j=1}^9 w_{i,j} \frac{c_j + d_j}{2} \right]^T.$$

Comparing the uniform mixture (UM) method to the center of mass (COM) method, the centroid from each method is to be

$$\mathbf{m}_{\text{UM}} = \begin{bmatrix} 4.7736 \\ 4.7902 \end{bmatrix} \quad \text{and} \quad \mathbf{m}_{\text{COM}} = \begin{bmatrix} 5.4812 \\ 5.5178 \end{bmatrix}.$$

These are shown in Figure 3.8, along with the true mean centered at (5, 5). Along with the mean of the uniform mixture model, the covariance can also be obtained, unlike in the center of mass centroiding method. For this application, the goal is to develop a non-iterative way of determining the centroid of a star, but not its spread, and because of this only the mean is of importance. Further steps can be taken that use the covariance matrix for calculating the optimum Gaussian parameters, with respect to the KL-divergence. Doing so invokes iterations as it determines these parameters. Others have used this iterative process to further predict a Gaussian fit over a two-dimensional pixel array [8] with a different cost function.

Now, turning back to the image used for the center of mass algorithm, Figure 3.3, the centroids of the stars can be calculated using the uniform mixture model

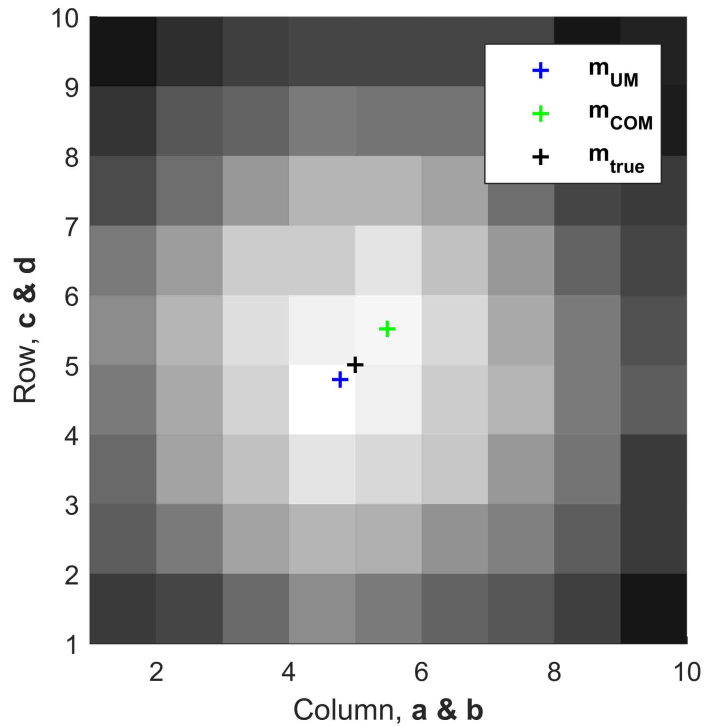


Figure 3.8. The centroids found by the uniform mixture model and center of mass methods are shown, along with the true centroid at (5, 5).

algorithm. Following the same process used in the multivariate example, Figure 3.9 shows the centroid of the three brightest stars using both the center of mass and uniform mixture model methods. The centroids calculated are exactly the same for both methods, unlike in Figure 3.8. They produce near exact solutions, but this is due to the low level of noise in the image, as well as the symmetric shape of the Gaussian used to produce the image. This indicates that in images with high levels of noise, the uniform mixture model method provides a more reliable centroid. Again, the advantage to the uniform mixture model method lies in the non-iterative way of determining the centroid, as well as the possibility of iteratively determining a more precise calculation.



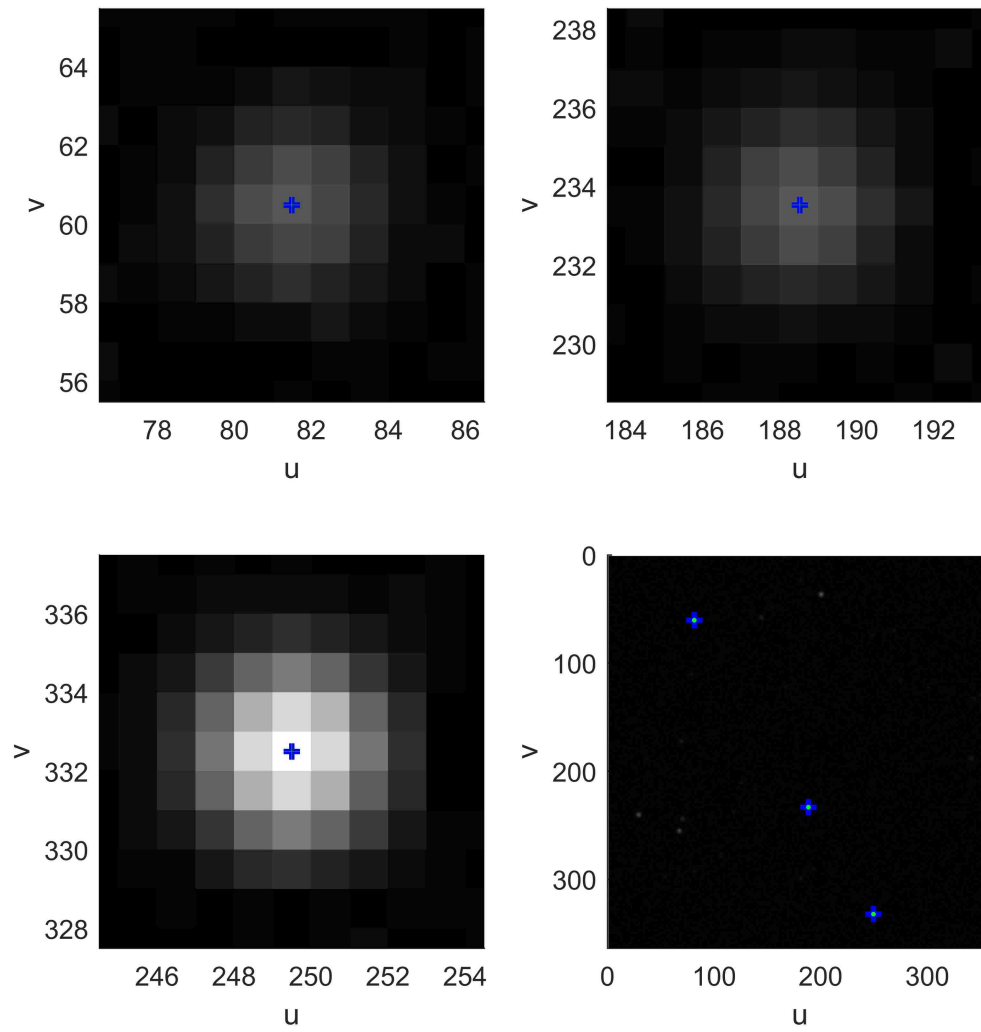


Figure 3.9. The centroids found by the uniform mixture model and center of mass methods are shown, blue and green, respectively, for Figure 3.3.

## 4. CATALOG FORMATION

### 4.1. TROUBLES FOR CURRENT CATALOGS

The difficulty arises due to the fact that a generic magnitude cut-off cannot be determined. Some works have a star magnitude limit of 6.0, whereas others have their limit set at magnitude 7.5 or even 8.0 [6, 31]. There are two problems with having a limit on the magnitude of stars in the catalog. As previously stated, hardware degrades over time and sensors may no longer be able to detect dimmer stars. The other issue is that with any level of cut-off, there will be less information for the star tracker to use in order to produce an accurate measurement. The goal in the presented works is to have a minimum disk space capacity with the maximum amount of information.

The main concern with any reduction of the reference star catalog is a lack of stars in the field of view. Ideally, the star catalog would have a uniform distribution over the entire sphere, allowing for a consistent number of stars to always be present in the field of view. Unfortunately, stars are far from uniform, with most concentrated in the Galactic Plane and a lack of stars in the North Galactic Pole [32]. Figure 4.1 provides a visual of the absence of uniformity by the brightest stars in the catalog.

The issue of non-uniformity has been addressed a handful of times. The first attempt at building a nearly uniform catalog was done by Vedder in 1993 [33]. Vedder's technique uses the geometric distribution of stars in order to build an optimal star catalog based on global and local uniformity criteria. The global criterion measures the uniformity of the sphere of stars as a whole, whereas the local criteria measures the variation in the number of stars in a smaller region of constant shape. Essentially, Vedder's technique makes sure that the catalog as a whole is as uniform as possible and that within every image taken, the stars detected that are also in the catalog are as uniform as possible. Vedder does not take into account the brightness

of the stars in the field of view, however. Bauer includes brightness into the weighting of stars while using statistical measurements to generate uniformity [34]. Samaan *et al.* employ a different technique involving uniform points or patches (small areas) about a sphere to try and keep five stars within every image [35]. A newer method of catalog generation has been put forth by Saifudin *et al.* who use a clustering algorithm that stores fewer stars located in high density regions and stores more stars in low density regions [36]. A problem with these methods is that the star catalog is designed and built differently from the way a star tracker might detect stars and match their patterns with the catalog.

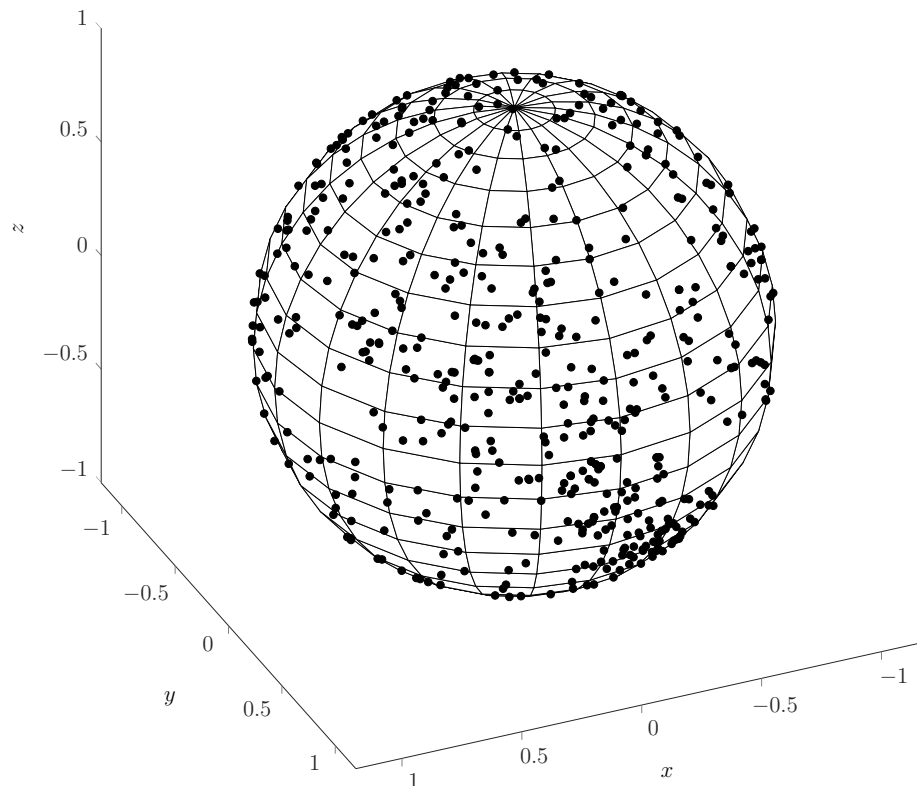


Figure 4.1. The 500 brightest stars of the Tycho-2 Catalog, in the visual spectrum, mapped onto a unit sphere.

## 4.2. MOTIVATION

The traditional star identification approach uses an angle calculation for determining the stars within the FOV, which calculates the angle between two stars and matches them to the catalog by checking that angle against every star in the catalog. This approach is tedious and can consume computational time to achieve a possible solution. Even then, it will, on average, require more than three stars to produce an accurate measurement [21].

The separation between catalog generation and star identification can force the catalog search algorithm to spend more time searching for the correct pattern and corresponding stars. This being said, the main objective of the aforementioned techniques is to save on memory allocation, minimizing the disk space taken up by the star catalog, not the time efficiency of the star tracker itself. The motivation for the catalog structure and design presented here is to formally connect the process by which the star tracker processes images and the way the catalog is utilized. Linking these processes should, in turn, decrease the search time between matches, utilize only the brightest stars, and provide a dependable star camera algorithm from start to finish.

To recall, stars are found in the images by first doing a template search. Once a star has been found, a bounding box is constructed around the star and the center of the star is calculated. After three stars are detected, the triangle is formed, and using the planar triangle technique is used to build the custom star catalog. The downside to this method is that it cannot utilize any sort of uniform distribution to the stars in the catalog since it depends on the brightest stars in any given field of view. One significant advantage, however, that this technique has is that it is invariant of the absolute brightness of a star; only the relative star magnitude matters. The algorithm selects the three brightest stars in an image, which means that it must only compare the stars within the image to one another in order to form the planar triangle.

Table 4.1. Tycho-2 Star Catalog Format.

Col.	Property	Meaning	Units
1	Ra	Right Ascension	radian
2	Dec	Declination	radian
3	pmRa	Proper Motion of the Right Ascension	milliarcsec per year
4	pmDec	Proper Motion of the Declination	milliarcsec per year
5	ErrRa	Error in Right Ascension	milliarcsec
6	ErrDec	Error in the Declination	milliarcsec
7	ErrpmRa	Error in the P.M. of the Ra	milliarcsec per year
8	ErrpmDec	Error in the P.M. of the Dec	milliarcsec per year
9	B <sub>T</sub>	Magnitude of a star (blue spectrum)	magnitude
10	ErrB <sub>T</sub>	Error in the Mag. of a star (blue spectrum)	magnitude
11	V <sub>T</sub>	Magnitude of a star (visual spectrum)	magnitude
12	ErrV <sub>T</sub>	Error in the Mag. of a star (visual spectrum)	magnitude
13	prox	Proximity indicator	arcsecond

### 4.3. BUILDING THE CATALOG

**4.3.1. Reference Catalog.** In order to built a customized star catalog, a reference catalog is needed as a base to build off of. Ideally, the reference will contain a high number of stars, and provide the necessary photometric band that the imaging sensor operates in. For these reasons, the Tycho-2 star catalog is chosen for this work. The Tycho-2 catalog is the largest collection of stars available, with over 2.5 million stars recorded from the European Space Agency’s Hipparcos satellite [22]. Along with the vast number of stars available, the Tycho-2 catalog is used due to the Hipparcos satellite’s ability to measure changes in a star’s position on the order of milliarcseconds per year. The format of this catalog can be found in Table 4.1. Many of these qualities are explained in the Tycho-2 guide [37]. The Tycho-2 catalogue has a large amount of extremely faint stars (fainter than  $B_T \approx 13$  and  $V_T \approx 12$ ), and most star trackers cannot detect these stars. Here, the maximum magnitude is kept 6.5 to allow for comparisons between other works, which encompasses only 0.32% of

the entire catalog. Along with the magnitude cut-off, stars with similar brightness that were too close to distinguish between were not considered.

**4.3.2. Catalog Outline.** The catalog utilizes triangular properties outlined in both Houtz's and Cole's works, but Houtz uses spherical triangles [8, 21]. Cole provides proof that planar triangles can provide better performance, which is the method used here. There is no maximum for disk space or number of stars, but instead the focus is on a more reliable system that utilizes only the brightest stars. The properties calculated for each star should be as uncorrelated as possible in the interest of providing the most amount of unique information for a triangle. Triangles have the convenience of having multiple measurable properties, whereas the standard angle measurements only have the one quantity and rely mostly on pivoting techniques [21]. From basic geometry, any triangle needs at least three pieces of information to be identified, such as two sides and the inner angle, or simply all three sides or all three internal angles. These simpler characteristics, more often than not, cannot provide enough data to differentiate between two similar triangles. Therefore, more complex properties are considered. These include perimeter, area, polar moment, the ratio between the shortest and longest side, and the normal coefficient.

**4.3.3. Catalog Generation.** A systematic approach to the formation of the catalog will allow for every possible combination of three stars in the sky to be found for a given field of view. Therefore, if different hardware is used, this step must be replicated to accommodate for the different specifications. The process takes time, but it is only necessary to execute once before flight. To ensure that no portion of the sky is missed, points are equally distributed over a sphere with a minimum distance between points being  $\delta \leq 1^\circ$ . Each point dictates the bore-sight, or  $z$ -axis, of the star tracker. To produce this uniform sphere of points, an icosahedron sphere is formed with enough positions equidistant from one another to produce the required  $\delta$  value, as shown in Figure 4.2. Utilizing the axis-angle attitude representation, the

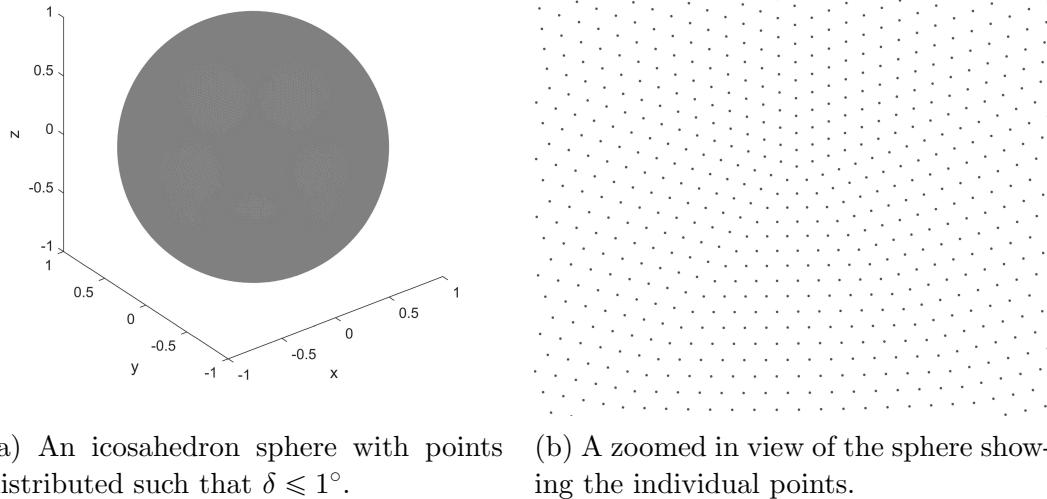


Figure 4.2. The icosahedron sphere used to change positions over the sky, ensuring complete coverage.

quaternion representing the attitude of the star tracker is formed via Eq. (2.2), where  $\mathbf{e}$  is the pointing vector of each point on the sphere. At each  $\mathbf{e}_i$  an image is generated and passed through the image processing chain to produce the three centroids. These centroids form the vertices of the triangle, whose properties are then calculated.

#### 4.4. PROPERTY STATISTICS

The five properties of planar triangles used here that distinguish one triangle from another are the perimeter, area, polar moment, the ratio of the shortest to longest triangle side, and the normal coefficient. These were taken from a paper by Houtz [8], who use the latter four properties. However, the goal for this star camera is accuracy and precision, whereas Houtz focuses on conserving hardware memory[8]. Thus, determining more properties, such as the perimeter, will allow for a higher chance of producing a correct triangle match, and hence a correct measurement. Unfortunately, image sensors are not perfect instruments, and any image taken will

have noise associated to it, descriptions of which are found in Section 2.2. In order to account for the noise, the statistics of the measured triangle properties are needed. Knowing the distribution for the parameters will enable a systematic approach in the star identification process. This proves challenging since the noise is added, not in the property calculation, but in the pixel extraction from the images. The pixel location  $p = (i, j)$  of a star centroid is corrupt by some amount of noise  $v$  such that  $p^* = (i + v, j + v) = (i^*, j^*)$ . Since  $i$  and  $j$  compose the same pixel, it is assumed that they receive the same amount of noise.

Taking  $\mathbf{x}$  to be the set of centroid locations

$$\mathbf{x} = \begin{bmatrix} i_1 & i_2 & i_3 \\ j_1 & j_2 & j_3 \end{bmatrix}^T$$

and  $\mathbf{v}$  to be the noise added to each pixel

$$\mathbf{v} = \begin{bmatrix} v_1 & v_2 & v_3 \\ v_1 & v_2 & v_3 \end{bmatrix}^T,$$

the noise corrupted pixels that are extracted from the image are

$$\mathbf{x}^* = \mathbf{x} + \mathbf{v} = \begin{bmatrix} i_1 + v_1 & j_1 + v_1 \\ i_2 + v_2 & j_2 + v_2 \\ i_3 + v_3 & j_3 + v_3 \end{bmatrix} = \begin{bmatrix} i_1^* & j_1^* \\ i_2^* & j_2^* \\ i_3^* & j_3^* \end{bmatrix}.$$

Using this, the sides of a triangle whose vertices are the star centroids are calculated as

$$\begin{aligned} a &= \left[ (i_1^* - i_2^*)^2 + (j_1^* - j_2^*)^2 \right]^{1/2} \\ b &= \left[ (i_1^* - i_3^*)^2 + (j_1^* - j_3^*)^2 \right]^{1/2} \\ c &= \left[ (i_2^* - i_3^*)^2 + (j_2^* - j_3^*)^2 \right]^{1/2}. \end{aligned}$$



From here, the perimeter, which is the sum of the sides, is calculated as

$$\mathcal{P} = a + b + c. \quad (4.1)$$

The next property, area, utilizes the semi-perimeter  $s = \mathcal{P}/2$  and is found to be

$$\mathcal{A} = (s(s - a)(s - b)(s - c))^{1/2}, \quad (4.2)$$

and similarly, the polar moment for a planar triangle is found as

$$\mathcal{J} = \frac{\mathcal{A}(a^2 + b^2 + c^2)}{36}. \quad (4.3)$$

The side length ratio is the relationship between the shortest to longest sides of the triangle, given as

$$\mathcal{S} = \frac{S_s}{S_l} = \begin{cases} \frac{a}{b} & \text{if } a < c < b \\ \frac{a}{c} & \text{if } a < b < c \\ \frac{b}{a} & \text{if } b < c < a \\ \frac{b}{c} & \text{if } b < a < c \\ \frac{c}{a} & \text{if } c < b < a \\ \frac{c}{b} & \text{if } c < a < b \end{cases}, \quad (4.4)$$

where  $S_s$  is the shortest side  $S_l$  is the longest side. Finally, the normal coefficient is the magnitude of the cross product of the shortest and longest sides. Since  $a$ ,  $b$  and  $c$  are all magnitudes, they cannot be used to calculate this property. Here, the vectors

$\mathbf{k}_n$ , for  $n = 1, 2, 3$ , are formed by

$$\mathbf{k}_1 = \begin{bmatrix} i_1^* \\ j_1^* \\ f \end{bmatrix}, \quad \mathbf{k}_2 = \begin{bmatrix} i_2^* \\ j_2^* \\ f \end{bmatrix}, \quad \text{and} \quad \mathbf{k}_3 = \begin{bmatrix} i_3^* \\ j_3^* \\ f \end{bmatrix},$$

where  $f$  is the focal length of the camera. To the shortest and longest sides, the variables

$$\begin{aligned} \tilde{\mathbf{a}} &= \mathbf{k}_1 - \mathbf{k}_2 & \tilde{a} &= \|\tilde{\mathbf{a}}\| \\ \tilde{\mathbf{b}} &= \mathbf{k}_1 - \mathbf{k}_3 & \tilde{b} &= \|\tilde{\mathbf{b}}\| \\ \tilde{\mathbf{c}} &= \mathbf{k}_2 - \mathbf{k}_3 & \tilde{c} &= \|\tilde{\mathbf{c}}\| \end{aligned} \quad \text{and}$$

are defined. The normal coefficient is found by

$$\mathcal{N} = \gamma \|\mathbf{S}_s \times \mathbf{S}_t\| = \gamma \begin{cases} \|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\| & \text{if } \tilde{a} < \tilde{c} < \tilde{b} \\ \|\tilde{\mathbf{a}} \times \tilde{\mathbf{c}}\| & \text{if } \tilde{a} < \tilde{b} < \tilde{c} \\ \|\tilde{\mathbf{b}} \times \tilde{\mathbf{a}}\| & \text{if } \tilde{b} < \tilde{c} < \tilde{a} \\ \|\tilde{\mathbf{b}} \times \tilde{\mathbf{c}}\| & \text{if } \tilde{b} < \tilde{a} < \tilde{c} \\ \|\tilde{\mathbf{c}} \times \tilde{\mathbf{a}}\| & \text{if } \tilde{c} < \tilde{b} < \tilde{a} \\ \|\tilde{\mathbf{c}} \times \tilde{\mathbf{b}}\| & \text{if } \tilde{c} < \tilde{a} < \tilde{b} \end{cases}, \quad (4.5)$$

where  $\gamma = \pm 1$  depending on the orientation of the triangle. If the long side is oriented counter-clockwise from the short side,  $\gamma = -1$ , otherwise  $\gamma = 1$ . With the normal coefficient being a piece-wise function, it is beneficial to look at it element by element.

Taking a look at the first element of Eq. (4.5), function gives

$$\begin{aligned}\tilde{\mathbf{a}} \times \tilde{\mathbf{b}} &= \left[ (i_1^* - i_3^*)(j_1^* - j_2^*) - (i_1^* - i_2^*)(j_1^* - j_3^*) \right] \hat{\mathbf{i}} + 0\hat{\mathbf{j}} + 0\hat{\mathbf{k}} \\ &= \left[ i_1^*(j_2^* - j_3^*) + i_2^*(j_1^* - j_3^*) + i_3^*(j_1^* - j_2^*) \right] \hat{\mathbf{i}} + 0\hat{\mathbf{j}} + 0\hat{\mathbf{k}},\end{aligned}$$

and taking the magnitude produces

$$\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\| = i_1^*(j_2^* - j_3^*) + i_2^*(j_3^* - j_1^*) + i_3^*(j_1^* - j_2^*).$$

Looking at the second element

$$\begin{aligned}\tilde{\mathbf{a}} \times \tilde{\mathbf{c}} &= \left[ (i_1^* - i_3^*)(j_2^* - j_3^*) - (i_2^* - i_3^*)(j_1^* - j_3^*) \right] \hat{\mathbf{i}} + 0\hat{\mathbf{j}} + 0\hat{\mathbf{k}} \\ &= \left[ i_1^*(j_2^* - j_3^*) + i_2^*(j_3^* - j_1^*) + i_3^*(j_1^* - j_2^*) \right] \hat{\mathbf{i}} + 0\hat{\mathbf{j}} + 0\hat{\mathbf{k}},\end{aligned}$$

it can be shown that all of the elements of the normal coefficient function produce the same magnitude, such that

$$\begin{aligned}\|\tilde{\mathbf{a}} \times \tilde{\mathbf{b}}\| &= i_1^*(j_2^* - j_3^*) + i_2^*(j_3^* - j_1^*) + i_3^*(j_1^* - j_2^*) \\ &= \|\tilde{\mathbf{a}} \times \tilde{\mathbf{c}}\| \\ &= \|\tilde{\mathbf{b}} \times \tilde{\mathbf{c}}\|\end{aligned}$$

and

$$\|\tilde{\mathbf{b}} \times \tilde{\mathbf{a}}\| = \|\tilde{\mathbf{c}} \times \tilde{\mathbf{a}}\| = \|\tilde{\mathbf{c}} \times \tilde{\mathbf{b}}\|.$$

The star-set triangle properties are summarized in Table 4.2.

Table 4.2. Star-Set Triangle Properties.

Property	Equation	Comment
$\mathcal{P}$	$a + b + c$	
$\mathcal{A}$	$(s(s-a)(s-b)(s-c))^{1/2}$	
$\mathcal{J}$	$\frac{\mathcal{A}(a^2 + b^2 + c^2)}{36}$	
$\mathcal{S}$	$\frac{S_s}{S_l}$	Cases can be found in Eq. (4.4)
$\mathcal{N}$	$\gamma \left( i_1^*(j_2^* - j_3^*) + i_2^*(j_3^* - j_1^*) + i_3^*(j_1^* - j_2^*) \right)$	Orientation dictates $\gamma = \pm 1$

#### 4.5. MONTE CARLO ANALYSIS

With the properties explicitly defined, the goal is to quantify the mean and variance of these properties for each star set. Monte Carlo simulation techniques provide a probabilistic analysis of functions whose pdf is difficult to analytically solve. The approach utilizes a numerical experimentation technique to obtain the statistics of the output for a system given the statistics of the input [38]. Here, the output consists of the five triangle properties. The input variables are the errors associated to the image, which alter the centroiding process, giving rise to variations in the property calculations. By running the entire image formation and image processing scheme  $N$  number of times, the mean and variance of the properties can be approximated for each star set.

Unlike typical circumstances where the noise is added to the  $f(x)$ ,

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{v},$$

the noise is now part of the pdf,

$$g(\mathbf{x}) = f(\mathbf{x}) + \mathbf{v} \rightarrow g(\mathbf{x}) = f(\mathbf{x}, \mathbf{v}).$$

For example, the area of a planar triangle is given as

$$\mathcal{A} = f_{\mathcal{A}}(\mathbf{x}, \mathbf{v}) = (s(s-a)(s-b)(s-c))^{1/2},$$

where the noise  $\mathbf{v}$  is found in the centroid which alter the values of  $a$ ,  $b$ , and  $c$ .

Taking a step back, the mean and covariance for an arbitrary pdf  $p(\mathbf{x})$  are given by

$$\mathbf{m} = \mathbb{E}\{\mathbf{x}\} \quad \text{and} \quad \mathbf{P} = \mathbb{E}\{(\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^T\}$$

and expanding the terms within the expectation operator gives

$$\mathbf{P} = \mathbb{E}\{\mathbf{x}\mathbf{x}^T - \mathbf{m}\mathbf{x}^T - \mathbf{x}\mathbf{m}^T + \mathbf{m}\mathbf{m}^T\} = \mathbb{E}\{\mathbf{x}\mathbf{x}^T\} - \mathbf{m}\mathbf{m}^T.$$

Switching to the integral form, the first moment (mean) and raw second moment,  $\mathbf{M}$ , are defined as

$$\mathbf{m} = \int \mathbf{x}p(\mathbf{x}) d\mathbf{x} \quad \text{and} \quad \mathbf{M} = \int \mathbf{x}\mathbf{x}^T p(\mathbf{x}) d\mathbf{x}. \quad (4.6)$$

Thus, the covariance is computed from Eq. (4.6) by

$$\mathbf{P} = \mathbf{M} - \mathbf{m}\mathbf{m}^T.$$

If there exists a function such that

$$\tilde{p}(\mathbf{x}) = \pi(\mathbf{x})p(\mathbf{x})$$

then it can be substituted into Eq. (4.6) which yields

$$\tilde{\mathbf{m}} = \int \mathbf{x}\pi(\mathbf{x})p(\mathbf{x}) d\mathbf{x} \quad \text{and} \quad \tilde{\mathbf{M}} = \int \mathbf{x}\mathbf{x}^T \pi(\mathbf{x})p(\mathbf{x}) d\mathbf{x}.$$

Knowing this, Monte Carlo integration [39] methods can be employed to approximate the mean and raw second moment by

$$\tilde{\mathbf{m}} = \frac{1}{N} \sum_{k=1}^n \pi(\mathbf{x}^{(k)}) \mathbf{x}^{(k)} \quad \text{and} \quad \tilde{\mathbf{M}} = \frac{1}{n} \sum_{k=1}^N \pi(\mathbf{x}^{(k)}) \mathbf{x}^{(k)} \mathbf{x}^{(k)T}.$$

A couple of reasons for this approach are that first, the noise is assumed independent from the property calculation and therefore their joint distribution can be found by

$$f_{\mathcal{A}}(\mathbf{x}, \mathbf{v}) = f_{\mathcal{A}}(\mathbf{x})f(\mathbf{v}) = p(\mathbf{x})\pi(\mathbf{x}),$$

which directly ties into  $\tilde{p}(\mathbf{x})$ . The second reason being that the pdf  $\pi(\mathbf{x})$  does not have to have the same distribution as  $\tilde{p}(\mathbf{x})$  even though the mean and covariance of  $\pi(\mathbf{x})$  are obtained from it. This means that if the distribution of  $\tilde{p}(\mathbf{x})$  and  $p(\mathbf{x})$  is known, and it is a joint distribution made up of two independent pdfs, then one can determine the distribution of  $\pi(\mathbf{x})$ , which in this case is the distribution of errors in an image. Here, finding  $\pi(\mathbf{x})$  is not the focus. Instead, it is sought to accurately determine  $\tilde{p}(\mathbf{x}) = f_{\mathcal{A}}(\mathbf{x}, \mathbf{v})$ , as well as the other triangle properties.

The simulation used here to determine how the errors in an image can affect the star-set properties is straightforward. The algorithm for approximating the mean and variance for this process is summarized in Algorithm 1.

---

**Algorithm 1** *Monte Carlo Simulation on Star-Set Properties.*

---

- 1: Generate an image with errors at position  $\mathbf{r}_i$ , based on the icosahedron sphere.
  - 2: Determine the centroid of the three brightest stars using the UMM method.
  - 3: Calculate the triangle properties from Table 4.2.
  - 4: Repeat steps two and three  $N$  number of times.
  - 5: Determine the mean and uncertainty of the set of  $N$  runs.
  - 6: Generate an image with errors at position  $\mathbf{r}_{i+1}$ .
  - 7: Repeat the process until the entire icosahedron sphere is covered.
- 

#### 4.6. MONTE CARLO ANALYSIS RESULTS

Recall the icosahedron sphere used to provide the pointing vectors that have a less than one degree change between positions. The sphere generates a total of 31,362 pointing vectors. Each pointing vector,  $\mathbf{r}_i$ , has 1,000 images simulated, and the steps carried forth continue as such in algorithm 1. Once the mean and variance for all five properties are determined from the 1,000 samples, at every pointing vector, the new catalog is built such that each page of the catalog is a property. Along with this new catalog, which will hence forth be called the Property Catalog, a second catalog is formed, the Star-Set Catalog that contains the stars associated to the corresponding row in the Property Catalog. The Property Catalog is formed such that each page is formatted like Table 4.3. The first column is simply indexing to show which pointing vector the row is associated to, and will be associated as the zeroth column. The first column of the catalog is the mean, then the standard deviation (square-root of variance), and the third is used to score stars that are being compared to the Property Catalog. In order to find a match, the Mahalanobis distance is found for a

Table 4.3. Property Catalog Format, Page One – Perimeter.

	Mean	Std	Test
Index 1	$\mu_{\mathcal{P}}^{(1)}$	$\sigma_{\mathcal{P}}^{(1)}$	$D_M^{(1)}$
2	$\mu_{\mathcal{P}}^{(2)}$	$\sigma_{\mathcal{P}}^{(2)}$	$D_M^{(2)}$
3	$\mu_{\mathcal{P}}^{(3)}$	$\sigma_{\mathcal{P}}^{(3)}$	$D_M^{(3)}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$

current image's star-set properties, whose general form is given by

$$D_M = \sqrt{(\mathbf{x} - \boldsymbol{\mu})^T \mathbf{P}^{-1} (\mathbf{x} - \boldsymbol{\mu})},$$

where  $\mathbf{x}$  is the measured properties,  $\boldsymbol{\mu}$  is the vector of means, and  $\mathbf{P}$  is the covariance matrix.

To demonstrate this process, assume the star tracker bore-sight is rotated such that

$$\alpha = 215.16^\circ \quad \text{and} \quad \delta = 36.19^\circ.$$

These angles form the axis

$$\mathbf{e} = \begin{bmatrix} -0.65986 \\ -0.46471 \\ 0.59046 \end{bmatrix}.$$



With an angle of rotation of  $90^\circ$ , the attitude, of the body frame with respect to the inertial frame, is

$$\bar{\mathbf{q}} = \begin{bmatrix} -0.55663 \\ -0.39201 \\ 0.49809 \\ 0.53703 \end{bmatrix},$$

shown in Figure 4.3. This attitude produces the image found in Figure 4.4. The

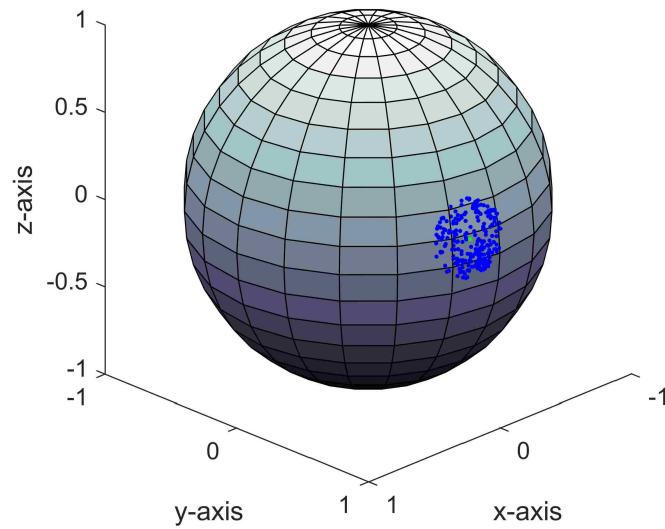


Figure 4.3. Field of view for  $\bar{\mathbf{q}}$ , blue dots indicating all the stars seen in the image, and the green dot represents the pointing vector.



Figure 4.4. Catalog matching example image.

centroids of the three brightest stars produce the properties

$$\begin{bmatrix} \mathcal{P} \\ \mathcal{A} \\ \mathcal{J} \\ \mathcal{S} \\ \mathcal{N} \end{bmatrix} = \begin{bmatrix} 379.69 \\ 6621.2 \\ 8.9583 \times 10^6 \\ 0.76572 \\ -13242 \end{bmatrix} \quad (4.7)$$

with units of pixels, pixels<sup>2</sup>, pixels<sup>4</sup>, unitless, and pixels<sup>2</sup>. Next, these properties are compared to the Property Catalog values and the Mahalanobis distance is calculated.

A glimpse of how the catalog looks like when comparing the properties is shown page by page in Tables 4.4 – 4.8. Looking at indices 176 and 177, they seem to provide similar scores for each property test. However, when summing up the scores for all pages, and then determining the minimum total  $D_M$  value over all pages, the star-set associated with index 177 provides the minimum summed  $D_M$ , making it a match to the three brightest stars in the image! This method shows that, though the Mahalanobis distance calculated might not always be the smallest for each property, the total score of an index is what matters. Now that the stars in an image are known, the next step is to back out the attitude of the spacecraft.

Table 4.4. Monte Carlo Example – Perimeter.

	Mean	Std	Test
⋮	⋮	⋮	⋮
176	378.758	0.04986	0.20712
177	379.760	0.05347	0.01721
178	621.080	0.07051	64.0996
⋮	⋮	⋮	⋮

Table 4.5. Monte Carlo Example – Area.

	Mean	Std	Test
⋮	⋮	⋮	⋮
176	6587.135	1.8976	46.986
177	6616.053	2.0030	7.346
178	6456.041	3.0050	286.385
⋮	⋮	⋮	⋮

Table 4.6. Monte Carlo Example – Polar Moment.

	Mean	Std	Test
⋮	⋮	⋮	⋮
176	$8.8697 \times 10^6$	4798.4	$6.138 \times 10^6$
177	$8.9581 \times 10^6$	5146.6	15287
178	$2.6366 \times 10^7$	15300	$2.1533 \times 10^9$
⋮	⋮	⋮	⋮

Table 4.7. Monte Carlo Example – Side-Length Ratio.

	Mean	Std	Test
⋮	⋮	⋮	⋮
176	0.7634	$1.7302 \times 10^{-4}$	$3.0131 \times 10^{-5}$
177	0.7612	$1.9300 \times 10^{-4}$	$6.3418 \times 10^{-5}$
178	0.3687	$6.6077 \times 10^{-5}$	$3.2280 \times 10^{-3}$
⋮	⋮	⋮	⋮

Table 4.8. Monte Carlo Example – Normal Coefficient.

	Mean	Std	Test
⋮	⋮	⋮	⋮
176	21588	6.2323	86954
177	21690	6.5682	89528
178	-17400	5.5255	9771.9
⋮	⋮	⋮	⋮

## 5. ATTITUDE DETERMINATION

Attitude determination, in the context of this work, refers to determining the spacecraft attitude based on two or more line-of-sight measurements from an image. The attitude of a body is quantified by some rotation away from a reference coordinate frame to the body coordinate frame. The attitude can be represented using several different parameter sets [2, 13, 40], and are usually taken to have three, four, or nine parameters. Since attitude only has three degrees of freedom, the four and nine parameter representations have one and six constraints, respectively. The most common three parameter set are Euler angles. Four-parameter sets include angle-axis and quaternion representations, and the nine-parameter representation is the attitude matrix.

The attitude determination techniques examined in this section are all considered “point” methods, versus those that take spacecraft dynamics into consideration. A point method is considered to be any method which uses unit vector measurements taken at a single time. This means that the methods do not use prior knowledge of the spacecraft’s dynamics or location to determine an attitude, therefore providing a “lost in space” scenario. The section starts with a survey of the more common attitude determination methods.

### 5.1. BACKGROUND AND OVERVIEW

Almost all point method algorithms are based on a problem presented in 1965 by Grace Wahba. Wahba’s problem finds the orthogonal matrix that minimizes the least-squares estimate of satellite attitude by the loss function [41]

$$L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^n a_i \|\mathbf{b}_i - \mathbf{A}\mathbf{r}_i\|^2, \quad (5.1)$$

where  $a_i$  are optional positive weights that depend on the confidence of a measurement,  $\mathbf{b}_i$  are unit vector observations,  $\mathbf{r}_i$  are the reference unit vectors,  $\mathbf{A}$  is the attitude matrix which transforms the reference vectors into the body frame, and the bars imply the Euclidean norm of the error inside. An example would be: let  $\mathbf{b}_1$  be a star's unit vector obtained from an image which is expressed in the spacecraft's body frame, and let  $\mathbf{r}_1$  be the reference unit vector from Earth to that star, which is expressed in the ECI J2000 frame. Using the loss function, the attitude matrix  $\mathbf{A}$  that maps the reference vector to the body frame observation can be determined, and in turn determines the current attitude of the body frame with respect to the ECI J2000 frame. The reference vectors are considered true, and due to noise the attitude matrix found will not perfectly map  $\mathbf{b}_1$  and  $\mathbf{r}_1$ , however it is considered the optimal solution by this loss function.

The purpose of using Wahba's problem is to attempt to find the square attitude matrix  $\mathbf{A}$  that will yield the minimum value for the loss function. Davenport's  $q$ -method is the basis of most current, applicable methods, where the optimal quaternion representation of the attitude matrix is found [42–45].

It should be noted that several methods use a Newton-Raphson iteration scheme to solve their corresponding characteristic equation, and it is beneficial to set the initial value as

$$\lambda_0 = \sum_{i=1}^n a_i. \quad (5.2)$$

As well, all the methods discussed use the  $\mathbf{B}$  matrix, which is defined as

$$\mathbf{B} = \sum_{i=1}^n a_i \mathbf{b}_i \mathbf{r}_i^T. \quad (5.3)$$

Another useful definition is given by the skew symmetric matrix given as

$$[\mathbf{e} \times] = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}. \quad (5.4)$$

Finally, it is also important to note that matrix norms used here are the Euclidean (or Frobenius, or Schur, or Hilbert-Schmidt) norm[44], such that

$$\|\mathbf{F}\|^2 \equiv \sum_{i=1}^n \sum_{j=1}^n F_{i,j}^2 = \text{tr}(\mathbf{F}\mathbf{F}^T), \quad (5.5)$$

where  $\text{tr}$  is the trace of a matrix.

The following sections of this section give a brief description of the various solutions to Wahba's problem. The Quaternion Estimator (QUEST), Estimator of the Optimal Quaternion (ESOQ), and Fast Optimal Attitude Matrix (FOAM) are the more common solutions used currently in space flight. Davenport's  $q$ -method and the Singular Value Decomposition, despite their principled robustness, are not actively implemented due to computational inefficiencies. As well, an in-depth derivation of the QUEST method and its covariance matrix are provided. The Quaternion Estimator is the most commonly used method due to the straightforward implementation and covariance [2]. To conclude the section, each method's accuracy and computational efficiency are analysed and compared against one another.

## 5.2. ATTITUDE DETERMINATION SURVEY

**5.2.1. Davenport's  $q$ -Method.** Paul Davenport provided the first usable solution to the least-squares estimate of attitude posed by Wahba during his time on the High Energy Astronomy Observation (HEAO-B) in 1978 [46]. While on the team, he developed a more robust way of finding the attitude. Davenport took the

loss function, a function of an orthogonal attitude matrix, and made it a function which depends on a quaternion attitude representation, where

$$\bar{\mathbf{q}} = \begin{bmatrix} \mathbf{q} \\ q \end{bmatrix} = \begin{bmatrix} \mathbf{e} \sin(\phi/2) \\ \cos(\phi/2) \end{bmatrix}. \quad (5.6)$$

where  $\mathbf{q}$  is the  $3 \times 1$  vector part of the quaternion, and  $q$  is the scalar. The variable  $\mathbf{e}$  defines an axis which the angle  $\phi$  rotates about. When converting the loss function into one dependent on the attitude quaternion, he concluded that Eq. (5.1) could be rewritten as

$$\begin{aligned} L(\mathbf{A}(\bar{\mathbf{q}})) &= \lambda_0 - \sum_{i=1}^n a_i \mathbf{b}_i^T \mathbf{A}(\bar{\mathbf{q}}) \mathbf{r}_i \\ &= \lambda_0 - \text{tr}(\mathbf{A}(\bar{\mathbf{q}}) \mathbf{B}^T), \end{aligned} \quad (5.7)$$

where he shows that [46]

$$\text{tr}(\mathbf{A}(\bar{\mathbf{q}}) \mathbf{B}^T) = \bar{\mathbf{q}}^T \mathbf{K} \bar{\mathbf{q}}. \quad (5.8)$$

Here, the  $\mathbf{K}$  matrix is defined as

$$\mathbf{K} = \left[ \begin{array}{c|c} \mathbf{S} - \sigma \mathbf{I}_3 & \mathbf{z} \\ \hline \mathbf{z}^T & \sigma \end{array} \right], \quad (5.9)$$

with

$$\mathbf{z} = \sum_{i=1}^n a_i (\mathbf{b}_i \times \mathbf{r}_i) \quad (5.10)$$

$$\mathbf{S} = \mathbf{B} + \mathbf{B}^T \quad (5.11)$$

$$\sigma = \text{tr} \mathbf{B}. \quad (5.12)$$



From Eqs. (5.7) and (5.8), it can be shown that the optimal quaternion,  $\bar{\mathbf{q}}_{\text{opt}}$ , is the eigenvector,  $\lambda_{\text{max}}$ , corresponding to the largest eigenvalue of the  $\mathbf{K}$  matrix, which implies that

$$\mathbf{K}\bar{\mathbf{q}}_{\text{opt}} = \lambda_{\text{max}}\bar{\mathbf{q}}_{\text{opt}}. \quad (5.13)$$

It should be noted that there is no unique solution if the two largest eigenvalues are equal, but this is not due to the algorithm. If this is the case, it means that there is an insufficient amount of data in order to determine a unique attitude solution. The four eigenvalues and the three other eigenvectors are not required for this method since simply knowing which vector is associated to the largest eigenvalue provides the necessary solution. There are many ways to solving the eigenvalue problem. In a paper by Keat, he provides a detailed approach for using the power method, or matrix iteration approach, in solving the eigenvalue problem for the  $q$ -method [46]. As mentioned before, the  $q$ -method is sound in principle, but is no longer implemented due to the difficulty in solving the eigenvalue/eigenvector problem.

**5.2.2. Quaternion Estimator (QUEST).** Due to the computational burden of solving for eigenvalues and eigenvectors, a more efficient and faster attitude determination algorithm was needed. First applied in 1979 on the MAGSAT mission, QUEST is one of the most widely used methods for solving Wahba's problem [44, 47]. The Quaternion Estimator gains its computational efficiency from replacing the eigenvalue and eigenvector calculations with a solution of the characteristic equation of  $\mathbf{K}$ . Solving this 4<sup>th</sup>-order equation calls for a root finding algorithm, the most common of which is the Newton-Raphson method. A detailed derivation of QUEST is provided in Section 5.3.

**5.2.3. Estimator of the Optimal Quaternion (ESOQ).** The Estimator of the Optimal Quaternion (ESOQ) was developed by Mortari in hopes of further

decreasing the computation time for providing an attitude estimate [45]. He comments that any of the methods produced previously only differ in their computational efficiency, but still achieve the same accuracy. Mortari developed ESOQ to be a singularity-free algorithm and provide a closed-form solution for  $\lambda_{\max}$  and  $\bar{\mathbf{q}}_{\text{opt}}$ .

Davenport's eigenvalue equation, Eq. (5.13), can be rewritten as

$$(\mathbf{K} - \lambda_{\max} \mathbf{I}_4) \bar{\mathbf{q}}_{\text{opt}} = \mathbf{H} \bar{\mathbf{q}}_{\text{opt}} = 0. \quad (5.14)$$

This implies that  $\bar{\mathbf{q}}_{\text{opt}}$  must be orthogonal to the four-row column vectors  $\mathbf{h}_i$  of  $\mathbf{H}$ , where  $\mathbf{H} = \mathbf{H}^T = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3 \ \mathbf{h}_4]$  such that  $\mathbf{h}_i^T \bar{\mathbf{q}}_{\text{opt}} = 0$  for  $i = 1, \dots, 4$ . In order for a unique solution to be determined, these four vectors must form a three-dimensional hyperplane. If they are all part of the same hyperplane, they are linearly correlated, meaning that the determinant of  $\mathbf{H}$  is zero, i.e.  $\det \mathbf{H} = 0$ . From this, it is determined that  $\bar{\mathbf{q}}_{\text{opt}}$  is the vector perpendicular to this hyperplane if  $\text{rank} \mathbf{H} = 3$ .

Mortari proves that the optimal quaternion can be computed by the generalized four-dimensional cross product of the three columns of  $\mathbf{H}$  [45]. To better understand this process, consider the adjoint of  $\mathbf{H}$ , for any value of  $\lambda$ ,

$$\begin{aligned} \text{adj} \mathbf{H} &= \text{adj} \left[ \sum_{k=1}^4 (\lambda_k - \lambda) \bar{\mathbf{q}}_k \bar{\mathbf{q}}_k^T \right] \\ &= \sum_{k=1}^4 (\lambda_2 - \lambda)(\lambda_3 - \lambda)(\lambda_4 - \lambda) \bar{\mathbf{q}}_k \bar{\mathbf{q}}_k^T, \end{aligned}$$

where  $\mathbf{K}$  is represented by its eigenvalues and eigenvectors,  $\lambda_{1-4}$  and  $\bar{\mathbf{q}}_k$ . By setting  $\lambda = \lambda_{\max}$ , the sum vanishes except for  $k = 1$ , and gives the result

$$\text{adj} \mathbf{H} = (\lambda_2 - \lambda_{\max})(\lambda_3 - \lambda_{\max})(\lambda_4 - \lambda_{\max}) \bar{\mathbf{q}}_{\text{opt}} \bar{\mathbf{q}}_{\text{opt}}^T. \quad (5.15)$$

From Eq. (5.15), it is determined that  $\bar{\mathbf{q}}_{\text{opt}}$  can be calculated by normalizing any non-zero column of  $\text{adj}\mathbf{H}$ , denoted with index  $k$ .

Denote  $\xi$  to be the  $k$ th element of the  $k$ th column of  $\mathbf{H}$  and  $\boldsymbol{\xi}_r$  and  $\boldsymbol{\xi}_c$  be the corresponding  $k$ th row and  $k$ th column. Let  $\mathbf{f}$  be the three-dimensional column vector achieved when  $\xi$  is deleted from  $\boldsymbol{\xi}_c$ , and let  $\mathbf{F}$  to be the square matrix left when  $\boldsymbol{\xi}_r$  and  $\boldsymbol{\xi}_b$  are deleted from  $\mathbf{H}$ . From here the  $k$ th element of the optimal quaternion is calculated as [44]

$$(\bar{\mathbf{q}}_{\text{opt}})_k = -c \det \mathbf{F}, \quad (5.16)$$

and the rest of the elements are

$$(\bar{\mathbf{q}}_{\text{opt}})_{1,\dots,k-1,k+1,\dots,4} = c(\text{adj}\mathbf{F})\mathbf{f}, \quad (5.17)$$

where  $c$  is a constant found by normalizing the quaternion. Typically  $c$  is not calculated, and is there to represent the normality constraint for the quaternion. It is ideal to choose the column with the maximum Euclidean norm, which in turn is the column containing the maximum value in the diagonal of  $\text{adj}\mathbf{H}$  due to  $\mathbf{H}$  being symmetric. It should be noted that the matrix  $\mathbf{F}$  depends on  $\lambda_{\text{max}}$ , which is found by solving the same characteristic equation as QUEST. Originally, ESOQ used the analytic solution of eigenvalues [45], but unfortunately this often gives complex roots, which is impossible for real, symmetric matrices [44].

**5.2.4. Singular Value Decomposition (SVD).** One of the first solutions to Wahba's problem leverages the singular value decomposition of the  $\mathbf{B}$  matrix [42]. This method was key in many early solutions of Wahba's problem; however, it is no longer a method used in practice due to the computational expense. Taking the singular value decomposition, while less expensive than finding the eigenvalues and

eigenvectors of  $\mathbf{K}$ , can still be costly. One advantage is that a robust method for determining the optimal attitude matrix can be formulated.

The singular value decomposition of the  $\mathbf{B}$  matrix is defined by

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^T = \mathbf{U}\text{diag}([s_1 \ s_2 \ s_3])\mathbf{V}^T,$$

where the  $\mathbf{U}$  and  $\mathbf{V}$  matrices are orthogonal, and with  $s_1 \geq s_2 \geq s_3$ . Since  $\mathbf{U}$  and  $\mathbf{V}$  do not necessarily have a determinant of +1, a new pair of rotation matrices are defined as

$$\mathbf{U}_+ \equiv \mathbf{U}\text{diag}([1 \ 1 \ \det \mathbf{U}]) \quad \text{and} \quad \mathbf{V}_+ \equiv \mathbf{V}\text{diag}([1 \ 1 \ \det \mathbf{V}]), \quad (5.18)$$

which ensures that  $\mathbf{U}_+$  and  $\mathbf{V}_+$  are proper orthogonal matrices. The diagonal matrix  $\mathbf{S}$  can be redefined by  $\mathbf{S}' = \text{diag}([s_1 \ s_2 \ ds_3])$ , where  $d = (\det \mathbf{U})(\det \mathbf{V}) = \pm 1$ . From here,  $\mathbf{B}$  can be defined as

$$\mathbf{B} = \mathbf{U}_+\mathbf{S}'\mathbf{V}_+. \quad (5.19)$$

Now, define a rotation matrix, and its angle-axis representation, where

$$\mathbf{W} = \mathbf{U}_+^T\mathbf{A}\mathbf{V}_+ \quad (5.20)$$

$$= (\cos \phi)\mathbf{I}_3 - \sin \phi[\mathbf{e}\times] + (1 - \cos \phi)\mathbf{e}\mathbf{e}^T, \quad (5.21)$$

where  $[\mathbf{e}\times]$  is the skew symmetric matrix, Eq. (5.4), of the axis, and  $\phi$  is the angle of rotation about that axis. Looking at Eq. (5.7), the trace can be rewritten using

the rotation matrix  $\mathbf{W}$ , such that [2]

$$\begin{aligned}\text{tr}(\mathbf{A}\mathbf{B}^T) &= \text{tr}(\mathbf{S}'\mathbf{W}) \\ &= \mathbf{e}^T \mathbf{S}' \mathbf{e} + \cos \phi [s_2 + ds_3 + (s_1 - s_2)e_2^2 + (s_1 - ds_3)e_3^2].\end{aligned}$$

It is clear than this the trace is as a maximum when  $\phi = 0$ , which provides  $\mathbf{W} = \mathbf{I}_3$  and therefore the optimal attitude matrix is

$$\mathbf{A}_{\text{opt}} = \mathbf{U}_+ \mathbf{V}_+^T = \mathbf{U}[\text{diag}(1 \ 1 \ d)]\mathbf{V}^T. \quad (5.22)$$

**5.2.5. Fast Optimal Attitude Matrix (FOAM).** Similar to QUEST, the Fast Optimal Attitude Matrix (FOAM) method determines the maximum eigenvalue from a characteristic equation, and as  $q$ -method is to QUEST, SVD is to FOAM. This is to say, SVD is a precursor to FOAM, and because of the SVD method's cumbersome computation time, FOAM was developed in order to provide a more efficient attitude matrix estimate [43].

It should be noted that FOAM does not require the SVD of  $\mathbf{B}$ , which was the goal of Markley introducing this method [43]. He shows that  $\mathbf{A}_{\text{opt}}$  can be computed using the  $\mathbf{S}$  matrix provided from the SVD method, but uses some algebra to give,

$$\mathbf{A}_{\text{opt}} = [(\kappa + \|\mathbf{B}\|^2)\mathbf{B} + \lambda_{\max}(\text{adj}(\mathbf{B}^T)) - \mathbf{B}\mathbf{B}^T\mathbf{B}]/\zeta, \quad (5.23)$$

where

$$\begin{aligned}\kappa &= \frac{1}{2}(\lambda_{\max}^2 - \|\mathbf{B}\|^2), \\ \zeta &= \kappa\lambda_{\max} - \det \mathbf{B},\end{aligned}$$

thereby providing an attitude matrix solution to Wahba's problem that does not depend on the singular value decomposition of  $\mathbf{B}$ . The maximum eigenvalue  $\lambda_{\max}$ , can be found from rearranging Eq. (5.7) to give [43]

$$\lambda_{\max} = \lambda_0 - L(\mathbf{A}_{\text{opt}}) = \text{tr}(\mathbf{A}_{\text{opt}}\mathbf{B}^T) \quad (5.24)$$

$$= (\kappa\lambda_{\max} - \det \mathbf{B})^{-1} \left[ (\kappa + \|\mathbf{B}\|^2) \|\mathbf{B}\|^2 + 3\lambda_{\max} \det \mathbf{B} - \text{tr}(\mathbf{B}\mathbf{B}^T\mathbf{B}\mathbf{B}^T) \right] \quad (5.25)$$

Equation (5.25) is rather inconvenient and can be reduced using matrix algebra to form the characteristic equation

$$0 = \phi(\lambda_{\max}) \equiv (\lambda_{\max}^2 - \|\mathbf{B}\|^2)^2 - 8\lambda_{\max} \det \mathbf{B} - 4\|\text{adj} \mathbf{B}\|^2. \quad (5.26)$$

As with some of the previous methods, FOAM uses an iterative root-finding process to find  $\lambda_{\max}$ . Markley does provide an analytic solution along with the numeric, but comments that the five transcendental function evaluations required result in a slower algorithm with no additional accuracy [43].

### 5.3. THE QUATERNION ESTIMATOR

The following section, based off of work by Shuster, gives a complete derivation of the QUEST attitude determination scheme [48].

**5.3.1. The Quaternion Eigenvalue Problem.** Recall Wahba's loss function in Eq. 5.1, where the goal is to find an orthogonal matrix  $\mathbf{A}_{\text{opt}}$  that minimizes this function. Again recall that the weights  $a_i, i = 1, \dots, n$  are a set of non-negative weights which sum to unity. Consider the gain function

$$G(\mathbf{A}) = 1 - L(\mathbf{A}) = \frac{1}{2} \sum_{i=1}^n a_i \mathbf{b}_i^T \mathbf{A} \mathbf{r}_i \quad (5.27)$$

which is formed from the loss function. Now,  $G(\mathbf{A})$  is at a maximum when  $L(\mathbf{A})$  is at a minimum. Moving forward, the focus is on finding an optimal matrix  $\mathbf{A}_{\text{opt}}$  that maximizes  $G(\mathbf{A})$ .

Maximizing the gain function is complicated due to the nine components of the attitude matrix, as well as the six constraints of this matrix. With this, it is advantageous to express the attitude by a quaternion. Since the quaternion is a four-component representation of attitude, there is only one constraint, which is that the quaternion must be unit-norm; i.e.,

$$\bar{\mathbf{q}}^T \bar{\mathbf{q}} = \|\mathbf{q}\|^2 + q^2 = 1. \quad (5.28)$$

In order to determine the attitude matrix that maximizes  $G(\mathbf{A})$  using the quaternion, the attitude matrix dependence needs to be replaced by a quaternion dependence. The attitude matrix is given in terms of the quaternion as

$$\mathbf{A}(\bar{\mathbf{q}}) = (q^2 - \mathbf{q} \cdot \mathbf{q})\mathbf{I}_3 + 2\mathbf{q}\mathbf{q}^T - 2q[\mathbf{q} \times]^T \quad (5.29)$$

Substituting Eq. (5.29) into Eq. (5.27) yields

$$\begin{aligned} G(\bar{\mathbf{q}}) &= \text{tr} \left[ ((q^2 - \mathbf{q} \cdot \mathbf{q})\mathbf{I}_3 + 2\mathbf{q}\mathbf{q}^T - 2q[\mathbf{q} \times]^T) \mathbf{B}^T \right] \\ &= (q^2 - \mathbf{q} \cdot \mathbf{q}) \text{tr} \mathbf{B} + 2 \text{tr} [\mathbf{q}\mathbf{q}^T \mathbf{B}^T] + 2q \text{tr} ([\mathbf{q} \times]^T \mathbf{B}^T). \end{aligned} \quad (5.30)$$

Equation (5.30) defines the gain function as a function of a quaternion and the previously defined  $\mathbf{B}$  matrix. Next, each of the three terms in Eq. 5.30 can be simplified using previously established definitions and relationships. The first term is trivial, since it has previously been shown that  $\text{tr} \mathbf{B} = \sigma$ . Taking a look at the second term

in Eq. (5.30), it follows that

$$2 \operatorname{tr}[\mathbf{q}\mathbf{q}^T \mathbf{B}^T] = 2 \operatorname{tr}[\mathbf{q}^T \mathbf{B}^T \mathbf{q}] = 2\mathbf{q}^T \mathbf{B}^T \mathbf{q} = \mathbf{q}^T \mathbf{S} \mathbf{q} \quad (5.31)$$

where  $\mathbf{S}$  is defined in Eq. (5.11). Turning to the third term, it is explicitly written as

$$\begin{aligned} 2q \operatorname{tr}([\mathbf{q} \times]^T \mathbf{B}^T) &= 2q \operatorname{tr} \left\{ \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} \right\} \\ &= 2q \operatorname{tr} \left\{ \begin{bmatrix} B_{21}q_3 - B_{31}q_2 & B_{22}q_3 - B_{32}q_2 & B_{23}q_3 - B_{33}q_2 \\ B_{31}q_1 - B_{11}q_3 & B_{32}q_1 - B_{12}q_3 & B_{33}q_1 - B_{13}q_3 \\ B_{11}q_2 - B_{21}q_1 & B_{12}q_2 - B_{22}q_1 & B_{13}q_2 - B_{23}q_1 \end{bmatrix} \right\}. \end{aligned}$$

Applying the trace and collecting terms gives

$$2q \operatorname{tr}([\mathbf{q} \times]^T \mathbf{B}^T) = 2q [q_1(B_{32} - B_{23}) + q_2(B_{13} - B_{31}) + q_3(B_{21} - B_{12})],$$

and therefore, the right-hand side can be rewritten using Eq. (5.10) as

$$2q \operatorname{tr}([\mathbf{q} \times]^T \mathbf{B}^T) = 2q(-\mathbf{q} \cdot \mathbf{z}). \quad (5.32)$$

The gain function is therefore simplified to

$$G(\bar{\mathbf{q}}) = (q^2 - \mathbf{q} \cdot \mathbf{q})\sigma + \mathbf{q}^T \mathbf{S} \mathbf{q} + 2q(\mathbf{q} \cdot \mathbf{z}).$$

Rearranging terms gives the bilinear form

$$G(\bar{\mathbf{q}}) = \bar{\mathbf{q}}^T \mathbf{K} \bar{\mathbf{q}}, \quad (5.33)$$



where  $\mathbf{K}$  is defined in Eq. (5.9). The problem of finding the optimal attitude that maximizes the gain function has now been replaced by finding the optimal quaternion that maximizes the gain function. Unfortunately, Eq. (5.33) does not consider the unit norm constraint of the quaternion. By using Lagrange multipliers, the constraint is handled via a new function  $G'(\bar{\mathbf{q}})$ ,

$$G'(\bar{\mathbf{q}}) = \bar{\mathbf{q}}^T \mathbf{K} \bar{\mathbf{q}} - \lambda (\bar{\mathbf{q}}^T \bar{\mathbf{q}} - 1) \quad (5.34)$$

where  $\lambda$  is chosen such that the constraint is satisfied. To find a candidate optimum, Eq. (5.34) is differentiated with respect to  $\bar{\mathbf{q}}$ , giving

$$\begin{aligned} \frac{\partial}{\partial \bar{\mathbf{q}}} [G'(\bar{\mathbf{q}})] &= \frac{\partial}{\partial \bar{\mathbf{q}}} [\bar{\mathbf{q}}^T \mathbf{K} \bar{\mathbf{q}} - \lambda \bar{\mathbf{q}}^T \bar{\mathbf{q}}] \\ &= 2\bar{\mathbf{q}}^T \mathbf{K} - 2\lambda \bar{\mathbf{q}}, \end{aligned}$$

and setting this equal to zero, and knowing that  $\mathbf{K}$  is symmetric, it is found that

$$\mathbf{K} \bar{\mathbf{q}} = \lambda \bar{\mathbf{q}}.$$

Examination of this equation shows that the optimal quaternion,  $\bar{\mathbf{q}}_{\text{opt}}$  must be an eigenvector of  $\mathbf{K}$ . Knowing this,  $\lambda$  must be an eigenvalue of  $\mathbf{K}$ , and thus for each eigenvalue of  $\mathbf{K}$  is

$$G'(\bar{\mathbf{q}}) = \bar{\mathbf{q}}^T \mathbf{K} \bar{\mathbf{q}} = \lambda \bar{\mathbf{q}}^T \bar{\mathbf{q}} = \lambda.$$

With this, the gain function will be at a maximum if  $\lambda$  is chosen to be the maximum eigenvalue of  $\mathbf{K}$ , which gives  $\bar{\mathbf{q}}_{\text{opt}}$  as the corresponding eigenvector of  $\lambda_{\text{max}}$ . To clearly

express this,

$$\mathbf{K}\bar{\mathbf{q}}_{\text{opt}} = \lambda_{\text{max}}\bar{\mathbf{q}}_{\text{opt}}. \quad (5.35)$$

**5.3.2. QUEST Derivation.** A common representation of attitude takes the form of the Rodrigues parameters, developed by Rodrigues in 1840 [49]. Later, J. William Gibbs, inventor of modern vector notation, represented the Rodrigues parameters as the “vector semi-tangent of version,” thus giving it the name of the Gibbs vector [2, 50]. The vector,  $\mathbf{y}$ , is related to the quaternion and axis-angle representations via

$$\mathbf{y} = \frac{\mathbf{q}}{q} = \hat{\mathbf{e}} \tan\left(\frac{\theta}{2}\right). \quad (5.36)$$

The inverse of the Gibbs vectors allows the quaternion to be found, such that

$$\bar{\mathbf{q}} = \frac{1}{\sqrt{1 + |\mathbf{y}|^2}} \begin{bmatrix} \mathbf{y} \\ 1 \end{bmatrix}.$$

Rearrangement of Eq. (5.35) for any eigenvalue  $\lambda$  gives [48]

$$\mathbf{y} = [(\lambda + \sigma)\mathbf{I}_3 - \mathbf{S}]^{-1}\mathbf{z} \quad (5.37)$$

$$\lambda = \sigma + \mathbf{z} \cdot \mathbf{y}, \quad (5.38)$$

where  $\mathbf{y}$  is the Gibbs vector defined by Eq. (5.36). It can be seen that when the eigenvalue  $\lambda$  is equal to the maximum eigenvalue  $\lambda_{\text{max}}$ , both  $\mathbf{y}$  and  $\bar{\mathbf{q}}$  are representative of the optimal attitude solution for the gain function. Substituting Eq. (5.37) into (5.38) gives

$$\lambda = \sigma + \mathbf{z}^T [(\lambda + \sigma)\mathbf{I}_3 - \mathbf{S}]^{-1}\mathbf{z}, \quad (5.39)$$

which is equal to the implicit characteristic equation for the  $\mathbf{K}$  matrix in Eq. (5.9). The goal of developing QUEST is to avoid solving the  $\mathbf{K}$  matrix's eigenvalues explicitly. Note that

$$\lambda_{\max} = 1 - \frac{1}{2} \sum_{i=1}^n a_i \|\mathbf{b}_i - \mathbf{A}_{\text{opt}} \mathbf{r}_i\|^2$$

will be close to unity. Knowing this Shuster realized if  $\lambda_{\max} \approx 1$ , then substituting  $\lambda = 1$  into Eq. (5.37) provides a solution for the attitude which is second-order accurate in measurement errors, assuming that the denominator in the middle term of Eq. (5.39) is non-singular. Since Gibbs vectors are three-parameters representations of attitude, there is a singularity when the angle of rotation is  $180^\circ$ . Therefore, approximating the maximum eigenvalue by one is no longer useful when the angle of rotation is near  $180^\circ$ . The goal now is to develop a method that removes this singularity.

It should be noted that characteristic equation of any  $n \times n$  square matrix  $\mathbf{S}$ ,

$$\det |\mathbf{S} - \xi \mathbf{I}_n| = 0,$$

where  $\xi$  represents an eigenvalue of the matrix  $\mathbf{S}$ , takes the form, for  $n = 3$ ,

$$-\xi^3 + 2\sigma\xi^2 - \kappa\xi + \Delta = 0,$$

where

$$\sigma = \frac{1}{2} \text{tr} \mathbf{S}, \quad \kappa = \text{tr}(\text{adj} \mathbf{S}), \quad \text{and} \quad \Delta = \det \mathbf{S}.$$

According to the Cayley-Hamilton theorem, which states that every square matrix satisfies its own characteristic equation [51],

$$\mathbf{S}^3 = 2\sigma\mathbf{S}^2 - \kappa\mathbf{S} + \Delta\mathbf{I}_3. \quad (5.40)$$

Shuster finds that Eq. (5.40) can be rewritten to express any meromorphic function of  $\mathbf{S}$  as a quadratic in  $\mathbf{S}$ , specifically [48]

$$[(\omega + \sigma)\mathbf{I}_3 - \mathbf{S}]^{-1} = \gamma^{-1} (\alpha\mathbf{I}_3 + \beta\mathbf{S} + \mathbf{S}^2), \quad (5.41)$$

where

$$\alpha = \omega^2 - \sigma^2 + \kappa, \quad \beta = \omega - \sigma, \quad \text{and} \quad \gamma = (\omega + \sigma)\alpha - \Delta.$$

With simple rearrangement of Eq. (5.41), and substitution for  $\alpha, \beta$ , and  $\gamma$ , Eq. (5.40). From Eq. (5.41), Shuster realized that if  $\omega$  takes the value of  $\lambda_{\max}$  then

$$\mathbf{y}_{\text{opt}} = \frac{\mathbf{x}}{\gamma}, \quad (5.42)$$

where

$$\mathbf{x} = (\alpha\mathbf{I}_3 + \beta\mathbf{S} + \mathbf{S}^2)\mathbf{z}. \quad (5.43)$$

Recalling the quaternion form of Gibbs vector, Eq. (5.36), then Eq. (5.42) takes the form

$$\bar{\mathbf{q}}_{\text{opt}} = \frac{1}{\sqrt{\gamma^2 + |\mathbf{x}|^2}} \begin{bmatrix} \mathbf{x} \\ \gamma \end{bmatrix}. \quad (5.44)$$

Note that the Gibbs vector no longer plays a part in calculating the optimal attitude quaternion. Keep in mind that the goal is to find the maximum eigenvalue  $\lambda_{\max}$  of the  $\mathbf{K}$  matrix, avoiding any singularities or explicit solutions for eigenvalues. If Eq. (5.41) is substituted into Eq. (5.39), with some rearranging of variables, it can be shown that the characteristic equation takes the form

$$\lambda^4 - (a + b)\lambda^2 - c\lambda + (ab + c\sigma - d) = 0 \quad (5.45)$$

where

$$a = \sigma^2 - \kappa, \quad b = \sigma^2 + \mathbf{z}^T \mathbf{z}, \quad c = \Delta + \mathbf{z} \mathbf{S} \mathbf{z}, \quad \text{and} \quad d = \mathbf{z}^T \mathbf{S}^2 \mathbf{z}.$$

From here, finding  $\lambda_{\max}$  becomes a simple root finding process. In general,  $\lambda_{\max}$  is close to unity in order to maximize the gain function. Knowing this, the Newton-Raphson root finding method is ideal due to its rapid convergence with a strong initial guess. In practice, only one iteration can yield a practical result [44]. Thus, this analytic approach has a computational advantage when compared to the complete solution of the eigenvalue problem. It should be noted that solving the characteristic equation is, in principle, less reliable, and therefore, QUEST is less robust than Davenport's  $q$ -method. In practice however, solving the complete eigenvalue problem is more computationally expensive and yields a result that is no more accurate than QUEST [44].

#### 5.4. ANALYSIS OF SURVEY

The following section provides an analysis of the accuracy and speed of the five methods discussed. The comparison method is based off of work done by Markley and Mortari [44]. It should be noted that the  $q$ -method and SVD both utilize the built in MATLAB functions `eig` and `svd`, respectively. All other methods apply the formulas

given in this section. The methods are subjected to three line-of-sight measurements, as this is the case for the developed star tracker algorithm. In each run, the reference vectors,  $\mathbf{r}_i$ , are assumed to be normally distributed random unit vectors. These vectors are then converted to angles via

$$\begin{bmatrix} \theta_i \\ \phi_i \end{bmatrix} = \begin{bmatrix} \tan^{-1} \left( \frac{r_{y,i}}{r_{x,i}} \right) \\ \tan^{-1} \left( \frac{\sqrt{(r_{x,i}^2 + r_{y,i}^2)}}{r_{z,i}} \right) \end{bmatrix}.$$

Then the observation vectors,  $\mathbf{b}_i$ , are formed by

$$\mathbf{b}_i = \begin{bmatrix} \cos(\theta_i + \nu_{\theta_i}) \sin(\phi_i + \nu_{\phi_i}) \\ \sin(\theta_i + \nu_{\theta_i}) \sin(\phi_i + \nu_{\phi_i}) \\ \cos(\phi_i + \nu_{\phi_i}) \end{bmatrix}$$

where  $[\nu_{\theta_i} \nu_{\phi_i}]$  are Gaussian random noises with a standard deviation of 50 arcseconds. Weights are chosen such that they follow  $\mathcal{N}(0, 1)$ , and are forced to sum to one. The attitude is kept constant for all runs, since the only change is in the reference and observation vectors. Finally, to obtain an accurate representation of the accuracy and run time of each method, one million trials of three observations are given to each method.

Attitude errors are most useful when expressed in the spacecraft's body frame, and typically are parameterized by the first-order rotation vector  $\delta\boldsymbol{\theta}$ . The vector  $\delta\boldsymbol{\theta}$  defines the rotation from the estimated body frame  $\hat{\mathbf{B}}$  and the true body frame  $\mathbf{B}^*$ . Attitude errors can be calculated in various ways [2, 44]. When using quaternions to define the attitude, subtraction of two separate attitudes does not yield a useful

result. For this reason, attitude error is defined in a multiplicative sense [52], by

$$\begin{bmatrix} \frac{1}{2}\delta\boldsymbol{\theta} \\ 1 \end{bmatrix} = \bar{\mathbf{q}}_{\text{opt}} \otimes \bar{\mathbf{q}}^{-*} \quad (5.46)$$

where  $\delta\boldsymbol{\theta}$  is a three-parameter representation of the attitude error and  $\bar{\mathbf{q}}^{-*}$  is the inverse of the true attitude in the body frame with respect to the reference frame. These attitude errors are assumed to be small angles, and therefore, any singularities in the rotation vector representation are of no concern.

Each method's error was calculated and averaged over one million runs where each was given the same reference vectors, observation vectors, and weights. To provide consistency between methods, any method using Newton-Raphson iteration was limited to three iterations to converge. The mean errors between the optimal quaternion and the true quaternion are presented in Table 5.1. The table also includes mean errors between each method and the  $q$ -method. The  $q$ -method and SVD are based on robust matrix analysis algorithms, and therefore should provide the truly optimal solution[44]. By comparing methods to the  $q$ -method, the errors prove the validity of the presented attitude determination alternatives. The comparison with

Table 5.1. Mean Attitude Errors in Arcseconds.

Algorithm	Mean (optimal-to- $q$ )			Mean (optimal-to-true)		
	$\delta\theta_x$	$\delta\theta_y$	$\delta\theta_z$	$\delta\theta_x$	$\delta\theta_y$	$\delta\theta_z$
$q$ -method	-	-	-	33.4615	33.4372	32.5445
QUEST	$5.85 \times 10^{-10}$	$5.98 \times 10^{-10}$	$2.63 \times 10^{-10}$	33.4615	33.4372	32.5445
ESOQ	$7.07 \times 10^{-10}$	$8.94 \times 10^{-10}$	$5.09 \times 10^{-10}$	33.4615	33.4372	32.5445
SVD	$7.97 \times 10^{-11}$	$8.30 \times 10^{-11}$	$8.16 \times 10^{-11}$	33.4615	33.4372	32.5445
FOAM	$2.00 \times 10^{-10}$	$2.47 \times 10^{-10}$	$2.01 \times 10^{-10}$	33.4615	33.4372	32.5445

the true optimal  $q$ -method solution provides an interesting look at how each method compares. SVD yields the most accurate solution for this comparison, which is as expected due to the robust matrix algorithm used. The FOAM method gives the second most accurate solution to the optimal attitude; this is likely due to the fact that the FOAM characteristic equation, Eq. 5.26, deals with  $\mathbf{B}$  directly, while other methods use the  $\mathbf{S}$  matrix and  $\mathbf{z}$  vector.

The mean optimal-to-true rotation vector for each method is approximately the same, and has a magnitude of 57.376 arcseconds. Extending the significant figures behind the decimal in Table 5.1 would show the slight variations in the solutions, but these are left out to emphasize that the methods are equally close to the true attitude. The better accuracy in the third rotation vector component is due to the way in which noise is introduced into the system, being that the  $z$  component of the observation vectors only has one added noise. It is reassuring that each of these methods produces the same accuracy in each attitude error parameter, showing a consistency throughout the methods. Due to this consistency, the standard deviation for each method is the same for the optimal-to-true case, such that

$$\sigma_{\delta\theta} = \begin{bmatrix} 29.818 \\ 29.507 \\ 27.358 \end{bmatrix},$$

where the units are in arcseconds. Considering the level of noise in the system, these values provide a reasonable uncertainty in the estimate.

For each of the developed methods, their creators had the goal of providing a faster attitude estimate, that required less operations to obtain. Of the investigated methods, ESOQ is believed to provide the fastest solution [45]. The methods are timed such that only the actual estimating algorithm computation time is measured,



since the absolute time of calculating the basis variables used for multiple methods is not critical for ground computations. These absolute times were more relevant in the past when attitude solutions had to be performed by slower machines, which is why QUEST was such a large turning point in attitude estimation [44].

All runs were on a Windows 7 machine with the specifications below. To collect

Table 5.2. Computer Specifications.

Windows 7 Enterprise	
Processor:	Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50 GHz (2x)
RAM:	192 GB
System Type:	64-bit Operating System

the measurements, the MATLAB functions `tic` and `toc` are used to track the execution time of each method. In the past, there existed a function in MATLAB that counted the number of floating point operations (FLOPS) and allowed a more in-depth look at the number of computations for each method [44, 53]. Unfortunately, recent versions of Matlab have removed this function. Each method was subjected to a million runs with the same format as the accuracy test. The average and relative averages of these runs are shown in Figures 5.1 and 5.2 for varying numbers of measurements. The averages in Figure 5.1 give a confident look at what is expected by each method. As Mortari claims, ESOQ provides the fastest speed of the coded algorithms, with the QUEST computational time close to it. The claim that SVD and the  $q$ -method are unfit for practical use, due to computation time, might seem incorrect based on these plots. Their run times are significantly faster than the others, but this is primarily due to the use of the built in functions of MATLAB, which are no doubt optimized by Mathworks to be extremely efficient. These trends follow those of Cheng and

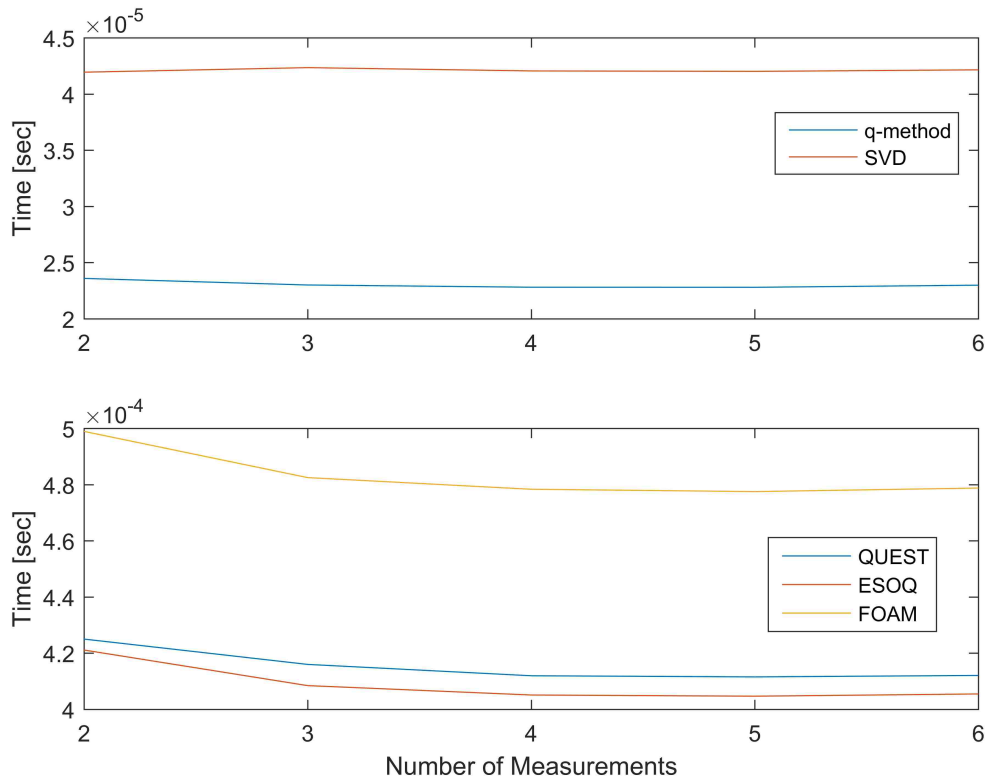


Figure 5.1. The average computation time for each attitude estimation algorithm.

Shuster's work, who compare times and FLOPS for several types of file structures [53], and show that even though computation times are unreliable, the number of FLOPS for SVD and  $q$ -method show their cumbersome calculations. These results are corroborated by the work done by Markley and Mortari, who also illustrate the higher number of FLOPS for the more robust methods [44].

The relative times in Figure 5.2 illustrate how the methods operate as they receive more measurements. The averages are made relative to the two measurement runs to see how they perform in comparison. The methods see a consistent decrease in computation times as they receive more measurements, which is possibly due to the nature of MATLAB function allocation which increases efficiency as functions are called repeatedly. What's interesting is the increase between five and six

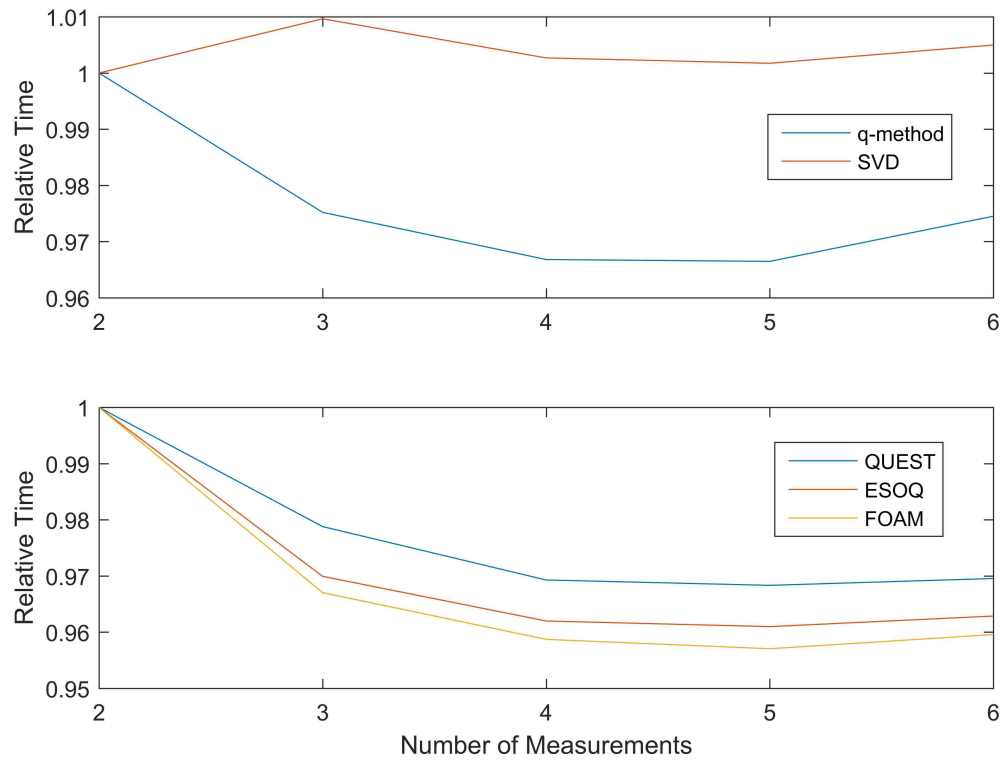


Figure 5.2. The relative average computation time which relates how the methods perform when given more measurements.

measurements. Presumably, this indicates that five measurements provide the ideal circumstance for the estimation algorithms to run efficiently, while also having an increase in accuracy.

## 6. SIMULATION AND RESULTS

### 6.1. SIMULATION CONFIGURATION

To demonstrate the complete star tracker algorithm, including the image formation algorithm, a Monte Carlo simulation is performed. The simulation takes the Tycho-2 star catalog, which has been broken down into a smaller catalog consisting only of stars equal to, or below, the visual spectrum magnitude of 6.5. No other editing of the stars within the catalog was performed in order to produce a realistic image. The image sensor whose properties the images are based on is the KAF-16803 CCD sensor from ON Semiconductor(R)<sup>2</sup>. The relative information from this sensor, with respect to image formation, is found in Table 6.1 along with additional parameter values used. The  $f$ -number and integration time are chosen depending on hardware characteristics, and the star radius and buffer size are chosen depending on image resolution and determine spread of each star in an image.

To begin, a pointing vector is formed from two uniform random angles,  $\theta$  and  $\phi$ , like those of Eq. (2.1), which relate to right-ascension and declination. From here, the algorithm performs as described by this thesis, starting with image formation and ending with an attitude estimate based on the unit vectors from an image. The line-of-sight measurements in the body frame of the camera are determined from an image by [54]

$$\mathbf{u}_i = \frac{1}{\sqrt{1 + \tan^2 \alpha_i + \tan^2 \beta_i}} \begin{bmatrix} \tan \alpha_i \\ \tan \beta_i \\ 1 \end{bmatrix}, \quad (6.1)$$

---

<sup>2</sup><http://www.onsemi.com/pub/Collateral/KAF-16803-D.PDF>

Table 6.1. Image Creation Parameters.

KAF16803 CCD Image Sensor				
Parameter	Min	Typical	Max	Units
Number of Active Pixels	-	4096 × 4096	-	pixels
Pixel Size	-	9.0 (H) × 9.0 (V)	-	μm
Saturation Signal	-	100,000	-	e <sup>-</sup>
Quantum Efficiency	-	60	-	%
Read Noise	-	9	15	e <sup>-</sup>
Dark Signal	-	3	15	e <sup>-</sup> /pix/sec
Photoresponse Non-Uniformity	-	1	-	%
Misc. Parameters				
Parameter	Min	Typical	Max	Units
Field of View (Circular)	-	25	-	degrees
<i>f</i> -number	-	f4	-	-
Integration Time	-	2.225	-	sec
Star Radius	-	1.6	-	pixels
Buffer Size	-	10	-	pixels <sup>2</sup>

where the angles  $\alpha_i$  and  $\beta_i$  are determined by [12]

$$\tan \alpha_i = 2(o_x - x_i) \frac{\tan(w_x/2)}{N_c} \quad \text{and} \quad \tan \beta_i = 2(o_y - y_i) \frac{\tan(w_y/2)}{N_r}. \quad (6.2)$$

Here,  $o_x$  and  $o_y$  are the pixel coordinates of the center of the image array,  $w_x$  and  $w_y$  are the horizontal and vertical pixel dimensions of the FOV, and  $N_c$  and  $N_r$  are the width and height of the image in pixels. To ensure that an accurate attitude solution is provided, the three brightest stars in each image are recorded and set aside to determine if the catalog matching scheme provides the correct solution. The reference vectors of the correct star triad are later passed to the attitude determination scheme, regardless of if the catalog matching algorithm provides the correct star triad. Based

on the analysis of Section 5.4, QUEST is used as the attitude determination scheme in the present analysis.

The entire process is subjected to a one million run Monte Carlo analysis to evaluate the performance of this star tracker algorithm. To quantify the performance, five characteristics are measured:

- time: the time the algorithm takes to run from receiving an image to providing an attitude estimate,
- iterations: the total number of thresholding iterations needed to find only the three brightest stars in an image,
- successful matches: the number of successful matches of triangles from an image to the Property Catalog,
- attitude errors: the attitude error characterized by  $\delta\theta$ , and
- attitude covariance: the covariance provided by the QUEST algorithm.

**6.1.1. Simulation Time Analysis.** All simulations are run on the following computer setup: where parallel computing is utilized for this Monte Carlo simulation

Table 6.2. Computer Specifications.

Windows 7 Enterprise	
Processor:	Intel(R) Xeon(R) CPU E5-2680 v3 @ 2.50 GHz (2x)
RAM:	192 GB
System Type:	64-bit Operating System

with 12 local workers. Computational time is relative depending on the system used, and due to the nature of MATLAB, the time at which a solution is found is arguably

frivolous. This being said, knowing the relative performance of the algorithm can help allow for better efficiency in the future. A histogram of the times can be found in Figure 6.1, where each bin is 1% of the one million trials. Runs that took longer than 1.0 seconds are considered to be outliers and are therefore omitted from the histogram. Times are based on the time the algorithm takes to provide an attitude measurement, and do not incorporate the time needed to construct the image since real-time image capturing is an extremely brief process for most current CCD sensors.

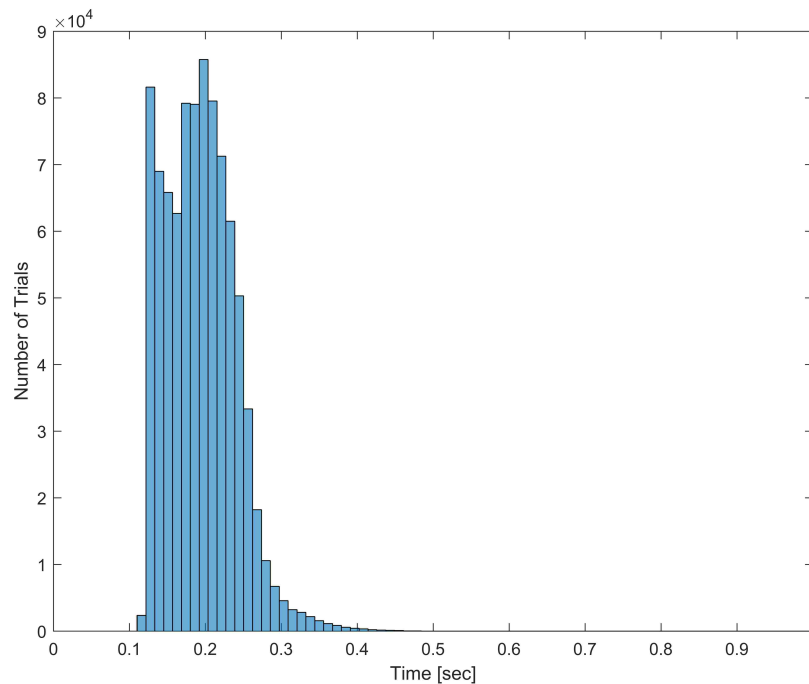


Figure 6.1. A histogram of the Monte Carlo trial times.

It seems that, when looking at the detailed time breakdown provided by MATLAB, the centroiding method, and specifically the iterative method used by the thresholding technique, accounts for the vast majority of computation time, which

is expected. The thresholding technique must evaluate the entire picture before it determines if enough stars are found, and then adjusts accordingly. In the cases that exactly three stars are not found, the thresholding scheme is flagged for that trial, and another image is simulated.

**6.1.2. Simulation Centroiding Analysis.** When examining the centroiding technique, two questions must be answered: did the process converge, and how many iterations did it take to converge on a solution? There are two reasons as to why the thresholding and centroiding processes would not converge. The first is due a lack of bright stars in the image, which would cause the centroiding method to only detect one or two bright stars, or one bright star and at least three dimmer stars, that fell below the threshold. Examples of these phenomena are shown in Figures 6.3 and 6.4. During these cases, the thresholding hits the limit of maximum iteration attempts, which is set at  $i_{\max} = 100$ , and the current image is discarded and a new image is simulated. These cases are classified as a failed thresholding attempt, and occurred in 1.2% of the trials. On average, the number of iterations the thresholding technique needed to converge was 11. A histogram of these values is shown in Figure 6.2, and it is clear that a large portion of the images only require one attempt at thresholding to find a solution. The histogram shows that the maximum number of iterations when convergence is achieved is 26, indicating that choosing  $i_{\max} = 100$  may be too high, but the selection of  $i_{\max} = 100$  does provide a suitable limit to indicate any statistical outliers.

**6.1.3. Simulation Catalog Matching Analysis.** The key aspect of the star tracker algorithm, the catalog matching scheme, successfully matched stars to the catalog 12.56% of the time. With this success rate, there is evidence to suggest that the matching algorithm can be improved. It can be concluded from the results that the method does work, but that it is not entirely reliable. There are several places that are potentially causing issues, one being the triangle properties themselves. Some values,



such as the polar moment, possess significantly large values compared to others, such as the side-length-ratio, which is bounded, i.e.  $\mathcal{S} \in (0, 1)$ . Additionally, the normal coefficient calculation has proved to be troublesome. Due to the orientation of the triangle, the normal coefficient can be positive or negative. If an incorrect value of  $\gamma$  is assigned, the value that is determined from the orientation of the triangle, then the value will be doubled. The normal coefficient phenomenon and the large polar moment values cause the Mahalanobis distance to be too large, even if the rest of the properties are correctly matched. Luckily, this is not always the case, as shown in the example in Section 4.6. Another cause for an unsuccessful match is potentially due to the underlying issue of creating a customized star catalog. The process itself eliminates stars from consideration by the algorithm, therefore providing fewer star triads for recognition. The last possible cause is the Mahalanobis distance itself. The method uses only the standard deviation of each property, and therefore the covariance matrices are simply diagonal. Without the full covariance matrix, any correlations that may help to better estimate the minimum distance between the data are lost. The consequence of using the correlations is that a 15 element symmetric covariance and five-state mean would have to be saved for each and every star-set, versus the five-state mean and five standard deviations.

**6.1.4. Simulation Attitude Determination Analysis.** Even if the catalog match is incorrect, the attitude is still estimated using the three brightest stars with respect to the image. The reference vectors of the three brightest stars, i.e. the line-of-sight vectors obtained from the reference catalog, are saved during image formation and are used in QUEST instead of the star triad provided by the catalog matching scheme. The observed vectors of the star triad are calculated using Eqs. (6.1) and (6.2). QUEST is applied to the reference and observed line-of-sight vectors and produces an estimate of the attitude via the quaternion,  $\hat{\mathbf{q}}$ . This estimate is compared to the true attitude of the star tracker to produce an attitude error, given

by Eq. (5.46). A histogram for each angle of the error is given in Figure 6.5, where the red lines are the  $3\sigma$  uncertainty intervals of the Monte Carlo data and the blue lines are the average  $3\sigma$  uncertainty intervals provided by QUEST. The intervals suggest, under the assumption of Gaussian statistics, that approximately 99% of all data should land between them. This assumption clearly does not hold for these results, but the mean and the standard deviations across all trials are

$$\mathbf{m}_{\delta\theta} = \begin{bmatrix} 0.01948 \\ 0.11866 \\ -0.35524 \end{bmatrix}, \quad \sigma_{\text{MCS}} = \begin{bmatrix} 1.080 \\ 1.327 \\ 14.794 \end{bmatrix}, \quad \text{and} \quad \sigma_{\text{QUEST}} = \begin{bmatrix} 1.1836 \\ 1.1919 \\ 8.6668 \end{bmatrix},$$

where the units are degrees, and where  $\sigma_{\text{MCS}}$  and  $\sigma_{\text{QUEST}}$  are the standard deviations of the Monte Carlo data and the average standard deviation produced by the QUEST algorithm, respectively. Though the mean is high for typical star trackers, approximately 81% of the error magnitudes were less than  $1^\circ$  and 42% of the error magnitudes were less than 50 arcseconds. University level star trackers usually have an accuracy in the 20 – 50 arcsecond accuracy range [12]. The standard deviations are large, and it seems that the QUEST algorithm provides overestimated measurements. It is apparent that the highest inaccuracy and uncertainty is in  $m_{\delta\theta_3}$ , which is the rotation about the  $z$ -axis. The high level of uncertainty is assumed to be due to the circular FOV, which incorporates the same information regardless of any rotation about the  $z$ -axis. Therefore, the suspected reason for these considerable errors lies in the simulated images. Along with simulating a circular FOV, instead of a typical rectangular FOV, the images presented are also of very low quality, around 0.14 megapixels. Due to the low image quality, the centroiding algorithm cannot provide an accurate enough centroid measurement to be used. It is believed that if higher resolution images are used that the errors would drastically decrease.

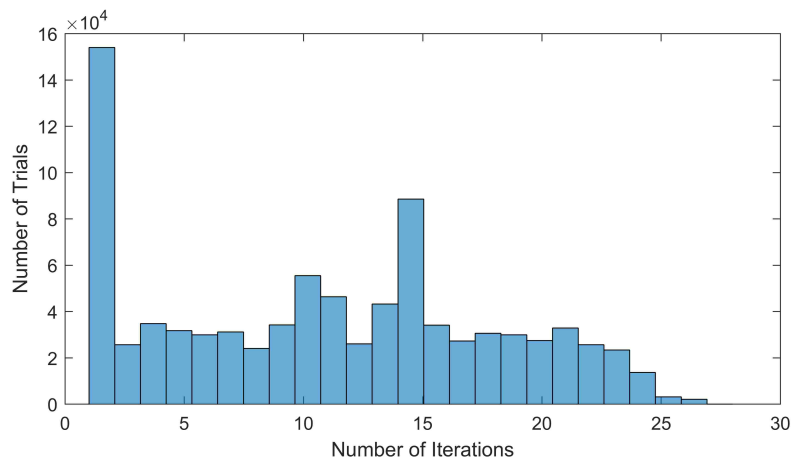


Figure 6.2. A histogram of the number of iterations needed to converge.

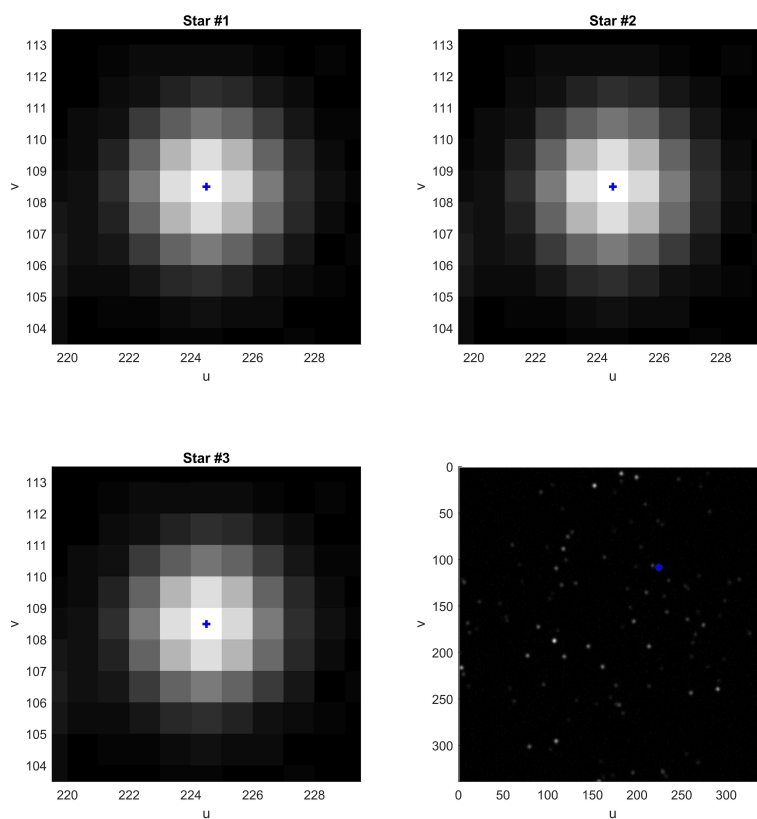


Figure 6.3. An example of unsuccessful thresholding. This case is when there are not enough distinguished bright stars in an image to find a set of three. Instead, the centroiding algorithm could only find the one brightest star in the image, and assumed that all three stars were this one star.

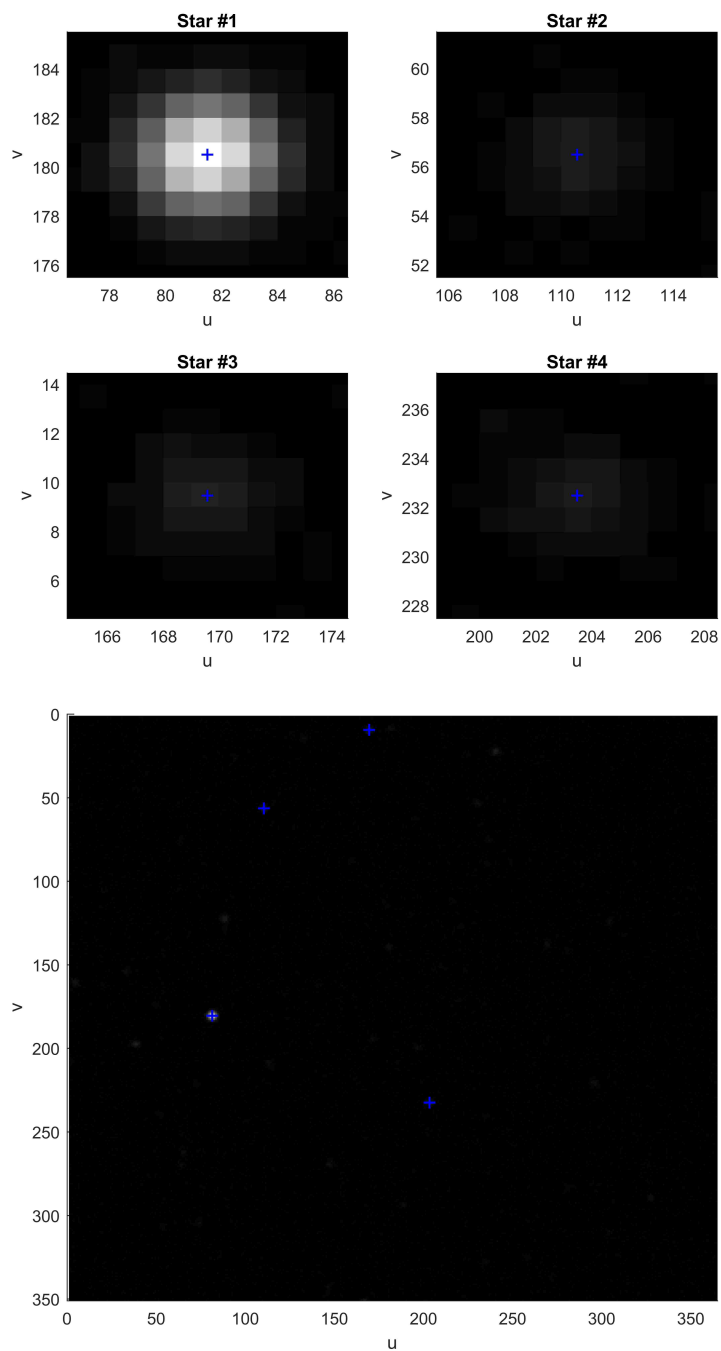


Figure 6.4. Another example of unsuccessful thresholding. Here, the method found one bright star (top left) and three dimmer stars that were all of similar brightness, and therefore the thresholding technique could not distinguish between the top three.

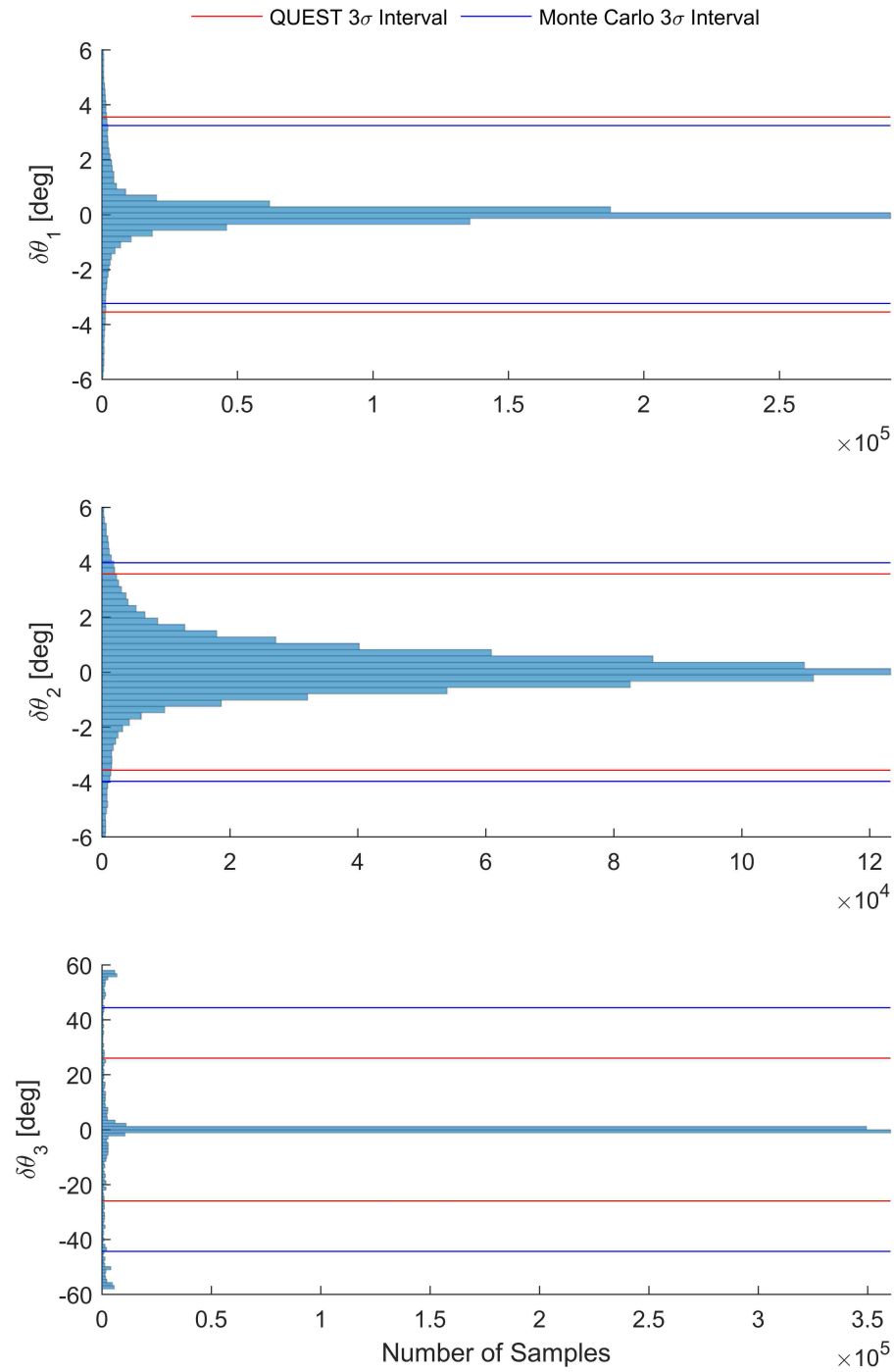


Figure 6.5. The attitude errors from the complete algorithm Monte Carlo simulation. The red lines indicate the  $3\sigma$  uncertainty intervals of the Monte Carlo data and blue lines indicate the average  $3\sigma$  uncertainty intervals provided by QUEST.

## 7. CONCLUSIONS

A star tracker algorithm has been developed and used to match the stars within a simulated image using a planar triangle properties technique to a known catalog. These images and algorithms were simulated without physical hardware and only depended on a reference star catalog and information found on data sheets of applicable hardware. The images incorporate straightforward star magnitude to electron conversion, detailed CCD sensor errors, and apply the formalized camera matrix in order to provide as realistic an image as possible. The implementation of the uniform mixture model technique has allowed for fast and accurate centroiding, and the planar triangle catalog matching scheme has shown that it can provide an accurate match from an image to the reference star catalog. To obtain the star triad, the Mahalanobis distance was used to score each star with the observed properties. Using the QUEST algorithm, reference-catalog stars were compared against those observed in the simulated images in order to produce a measurement. It is shown that the algorithm has the capabilities to successfully match observed stars with those in the Property Catalog and produce an accurate attitude measurement. The results of the simulations shed light on the areas of improvement for the algorithm, none of which degrade the validity of the approach taken for developing the star tracker software.

Even though the success rate was lower than expected, it is important to keep in mind that the developed algorithm still holds. It is possible to simulate a star tracker image, successfully match the stars found in the image to a reference catalog, and provide an accurate attitude estimate. These scenarios address the lost-in-space operational mode of a star tracker, which is considered to be the more difficult mode compared to the tracking mode. Although attitude errors are large and successful

matches are less frequent than desired, the path to improving these aspects is set forth and clearly explained.

Simulating a higher resolution image can potentially allow the algorithm to provide more accurate measurements. The only down-side with high resolution images appears in computation time. Custom image processing in MATLAB is non-efficient when compared to other hardware-level languages. Forming high resolution images will take longer to process, but since this is not an aspect of a star tracker, the consequence is longer simulations for ground testing. The thresholding technique can be sped up several ways, though. Certain calculations can be relocated outside the threshold loop, and the maximum number of iterations to find a solution can be reduced. Alternatively, the scheme can be designed to simply run once with a stagnant threshold value, and only the three brightest stars can be kept, regardless of the number of stars found. The catalog matching can be greatly improved with incorporation of the correlations of the covariance matrix. This will help better relate the observed properties with the pre-generated statistics of the properties. As well, scaling the triangle properties' values such that they are not as influential in the Mahalanobis distance calculation may increase the success rate. Additionally, other scoring techniques may be carried out to provide more comparative results. The QUEST algorithm is widely used, and very little can be done to improve upon it or any other attitude determination scheme available. However, a different form of, or approach to, the covariance calculation might provide a more accurate estimate of the attitude. With all of these improvements, it is highly probable that the outcome will be a more realistic, efficient, and accurate star tracker algorithm.

## BIBLIOGRAPHY

- [1] James R. Wertz. *Spacecraft Attitude Determination and Control*, volume 73. D. Reidel Publishing Company, 1978.
- [2] F. Landis Markley and John L. Crassidis. *Fundamentals of spacecraft attitude determination and control*, volume 33. Springer, 2014.
- [3] Matthew W. Knutson. Fast star tracker centroid algorithm for high performance cubesat with air bearing validation, 2010.
- [4] Malak Samaan, Stephen R. Steffes, and Stephan Theil. Star tracker real-time hardware in the loop testing using optical star simulator. *Spaceflight Mechanics*, 140, 2011.
- [5] Jack A. Tappe. Development of star tracker system for accurate estimation of spacecraft attitude, December 2009.
- [6] Cutris Padgett and Kenneth Kreutz-Delgado. A grid algorithm for autonomous star identification. *IEEE Transactions on Aerospace and Electronic Systems*, 33(1), January 1997.
- [7] Carl Christian Liebe. Accuracy performance of star trackers—a tutorial. *IEEE Transactions on aerospace and electronic systems*, 38(2):587–599, 2002.
- [8] Nathan Houtz and Carolin Frueh. Spacecraft attitude simulation to improve the efficiency of star trackers. In *AIAA/AAS Astrodynamics Specialist Conference*, number 15–779 in AAS, 2015.
- [9] Yan Feng, J. Goree, and Bin Liu. Accurate particle position measurement from images. *Review of scientific instruments*, 78(5):053704, 2007.
- [10] Stephen M. Anthony and Steve Granick. Image analysis with rapid and accurate two-dimensional gaussian fitting. *Langmuir*, 25(14):8152–8160, 2009.
- [11] Nathan Hagen and Eustace L. Dereniak. Gaussian profile estimation in two dimensions. *Applied Optics*, 47(36):6842–6851, 2008.
- [12] Jacob E. Darling, Nathan Houtz, Carolin Frueh, and Kyle J. DeMars. Recursive filtering of star tracker data. In *AIAA/AAS Astrodynamics Specialist Conference*, page 5672, 2016.
- [13] Jacob E. Darling. Bayesian inference for dynamic pose estimation using directional statistics, May 2016.
- [14] Thomas E. Strikwerda and John L. Junkins. Star pattern recognition and spacecraft attitude determination, 1981.



- [15] Carl Christian Liebe. Pattern recognition of star constellations for spacecraft applications. *IEEE Aerospace and Electronic Systems Magazine*, 7(6):34–41, 1992.
- [16] Daniele Mortari. A fast on-board autonomous attitude determination system based on a new star-ID technique for a wide FOV star tracker. *Advances in the Astronautical Sciences*, 93:893–904, 1996.
- [17] Daniele Mortari. Search-less algorithm for star pattern recognition. *J ASTRO-NAUT SCI*, 45(2):179–194, 1997.
- [18] Daniele Mortari. SP-search: A new algorithm for star pattern recognition. *Advances in the Astronautical Sciences*, 102(2):1165–1174, 1999.
- [19] Daniele Mortari, Malak A. Samaan, Christian Bruccoleri, and John L. Junkins. The pyramid star identification technique. *Navigation*, 51(3):171–183, 2004.
- [20] Craig L. Cole and John L. Crassidis. *Fast star pattern recognition using spherical triangles*, volume 5389. State University of New York at Buffalo, 2004.
- [21] Craig L. Cole and John L. Crassidis. Fast star pattern recognition using planar triangles. *Journal of Guidance, Control, and Dynamics*, 29(1):64–71, 2006.
- [22] E. Schilbach, Wenjing Jin, M. Cr ez e, P. D. Hemenway, I. I. Kumkova, I. K. Platais, S. R oser, C. Turon, and J. J. Wang. Commission 24: Photographic astrometry:(astrometrie photographique). *Transactions of the International Astronomical Union*, 24(1):48–59, 2000.
- [23] R. Dumont and A. C. L evasseur-Regourd. Zodiacal light and space observation of faint objects. *Advanced Space Research*, 1:127–130, August 1981.
- [24] A. C. L evasseur-Regourd and R. Dumont. Absolute photometry of zodiacal light. *Astronomy and Astrophysics*, 84:277–279, April 1980.
- [25] Thomas J. Fellers and Michael W. Davidson. Concepts in digital imaging technology, 2014. URL <http://hamamatsu.magnet.fsu.edu/articles/ccdsnr.html>.
- [26] K. Irie, A. E. McKinnon, K. Unsworth, and I. M. Woodhead. A model for measurement of noise in ccd digital-video cameras. *Measurement Science and Technology*, 19(4):045207, 2008.
- [27] Junichi Nakamura. *Image Sensor and Signal Processing for Digital Still Cameras*. Taylor and Francis Group, 2006.
- [28] Francois M. Klaassen. Characterization of low 1/f noise in MOS transistors. *IEEE Transactions of Electron Devices*, 18(10):887–891, October 1971.
- [29] Francisco J. Gallegos-Funes and Alberto J. Rosales-Silva. *Rank M-type Filters for Image Denoising*. InTech, December 2009.

- [30] O. Bendinelli, G. Parmeggiani, A. Piccioni, and F. Zavatti. The newton–gauss regularized method: Application to point–spread–function determination in CCD frames. *The Astronomical Journal*, 94(4), October 1987.
- [31] G. Lamy au Rousseau, J. Bostel, and B. Mazeri. Star recognition algorithm for aps star tracker: Oriented triangles. *IEEE Aerospace and Electrical Systems*, 20(2):27–31, February 2005.
- [32] Keith Robinson. *Starlight: An Introduction to Stellar Physics for Amateurs*. Springer, 2009.
- [33] John D. Vedder. Star trackers, star catalogs, and attitude determination: Probabilistic aspects of system design. *Journal of Guidance, Control, and Dynamics*, 16(3), May–June 1993.
- [34] Robert Bauer. Distribution of points on a sphere with applications to star catalogs. *Journal of Guidance, Control, and Dynamics*, 23(1), January–February 2000.
- [35] Malak A. Samaan, Christian Bruccoleri, Daniele Mortari, and John L. Junkins. Novel techniques for creating nearly uniform star catalog. *Advances in the Astronautical Sciences*, 116:1691–1703, 2004.
- [36] Muhammad Arif Saifudin, Bib Paruhum Silalahi, and Imas Sukaesih Sitanggang. Star catalog generation for satellite attitude navigation using density based clustering. *Journal of Computer Science*, January 2016.
- [37] E. Hog, C. Fabricius, V.V. Makarov, S. Urban, T. Corbin, G. Wycoff, U. Bastian, P. Schwkendiek, and A. Wicenec. *Guide to the Tycho–2 Catalogue*. Copenhagen University Observatory, United States Navel Observatory, Astronomisches Rechen–Institut, European Southern Observatory, 2000.
- [38] Thomas A. Cruse. *Reliability Based Mechanical Design*, volume 108. CRC Press, 1997.
- [39] Kyle J. DeMars and James S. McCabe, editors. *Multi–Sensor Data Fusion in Non–Gaussian Orbit Determination*, August 2014.
- [40] Malcolm D. Shuster. A survey of attitude representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, October–December 1993.
- [41] Grace Wahba. A least squares estimate of satellite attitude. *SIAM Review*, 8(3):384–386, July 1966.
- [42] F. Landis Markley. Attitude determination using vector observation and the singular value decomposition. *The Journal of the Astronautical Sciences*, 38(3):245–258, 1986.

- [43] F. Landis Markley. Attitude determination using vector observations: A fast optimal matrix algorithm. *The Journal of Astronautical Science*, 41:261 – 280, April - June 1993.
- [44] F. Landis Markley and Daniele Mortari. Quaternion attitude estimation using vector observations. *The Journal of the Astronautical Sciences*, 48(2 and 3):359 – 380, April - September 2000.
- [45] Daniele Mortari. Esoq: A closed-form solution of the wahba problem. *The Journal of the Astronautical Sciences*, 45(2):195–204, April–June 1997.
- [46] J. Keat. Analysis of least-squares attitude determination routine DOAOP, February 1977.
- [47] M. Shuster. The quest for better attitudes. *The Journal of Astronautical Sciences*, 54(3 and 4):657–683, July–December 2006.
- [48] Malcom D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, 4(1):70 – 77, January - February 1981.
- [49] M. Olinde Rodrigues. Des lois gomtriques qui rgissent les dplacements dun systme solide dans l'espace, et de la variation des coordonnes provenant de ces dplacements considrs indpendamment des causes qui peuvent les produire. *Journal De Mathematiques Pures et Appliquees*, 5:380–440, December 1840.
- [50] Edwin Bidwell Wilson and Josiah Willard Gibbs. *Vector analysis: a text-book for the use of students of mathematics and physics*. Scribner's Sons, 1902.
- [51] Wilson J. Rugh. *Linear system theory*, volume 2. Prentice Hall Upper Saddle River, NJ, 1996.
- [52] Kyle J. DeMars. Nonlinear orbit uncertainty prediction and rectification for space situational awareness, 2010.
- [53] Yang Cheng and Malcom D. Shuster. Speed testing of attitude estimation algorithms. *The Journal of the Astronautical Sciences*, 2008.
- [54] M. Shuster. Maximum likelihood estimation of spacecraft attitude. *The Journal of the Astronautical Sciences*, 37(1):79–88, January–March 1989.

## VITA

Casey Smith was born in Ferguson, MO and spent his formative years between Florissant, MO and Chesterfield, MO. Unsure of what to do with his life, he followed his best friend to Missouri University of Science and Technology in the fall of 2011. After being part of the rocket design team and then the satellite research team, he decided to take a crack at obtaining his Master's degree in aerospace engineering. Later, graduating with his B.S. in the spring of 2015, Casey obtained his first internship with NASA that summer, and then with the Air Force Research Lab in the following summer. With the help of Dr. DeMars and his fellow lab mates and friends, Casey received his M.S. in aerospace engineering in December of 2017 and went on to bigger and better things, but always remembered his humble beginnings.