

# Designing Star Trackers to Meet Micro-satellite Requirements

by

KARA M. HUFFMAN

B.S. Aeronautical and Astronautical Engineering  
University of Illinois at Urbana-Champaign, May 2004

SUBMITTED TO THE DEPARTMENT OF  
AERONAUTICAL AND ASTRONAUTICAL ENGINEERING  
IN PARTIAL FULLFILLMENT OF THE DEGREE OF

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 26, 2006

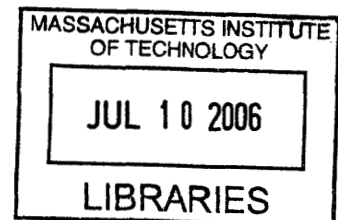
[June 2006]

© 2006 Massachusetts Institute of Technology  
All rights reserved

Signature of Author.....  
Department of Aeronautics and Astronautics  
May 26, 2006

Certified by.....  
Dr. Raymond J. Sedwick  
Department of Aeronautics and Astronautics  
Thesis Supervisor

Accepted by.....  
Professor Jaime Peraire  
Department of Aeronautics and Astronautics  
Chair, Committee on Graduate Student



ARCHIVES





# **Designing Star Trackers to Meet Micro-satellite Requirements**

by

KARA M. HUFFMAN

Submitted to the Department of Aeronautics and Astronautics  
on May 26, 2006 in Partial Fulfillment of the  
Requirements for the Degree of Master of Science  
in Aeronautics and Astronautics  
at the Massachusetts Institute of Technology

## **Abstract**

Star trackers provide numerous advantages over other attitude sensors because of their ability to provide full, three-axis orientation information with high accuracy and flexibility to operate independently from other navigation tools. However, current star trackers are optimized to maximize accuracy, at the exclusion of all else. Although this produces extremely capable systems, the excessive mass, power consumption, and cost that result are often contradictory to the requirements of smaller space vehicles. Thus, it is of interest to design smaller, lower cost, albeit reduced capability star trackers that can provide adequate attitude and rate determination to small, highly maneuverable, low-cost spacecraft. This thesis discusses the analysis used to select hardware and predict system performance, as well as the algorithms that have been employed to determine attitude information and rotation rates of the spacecraft. Finally, the performance of these algorithms using computer simulated images, nighttime photographs, and images captured directly by star tracker prototypes is presented.

Thesis Supervisor: Dr. Raymond J. Sedwick  
Title: Principle Research Scientist



# ACKNOWLEDGMENTS

To Mom and Dad, for teaching me to reach for the stars,

To Jonathan, for being my biggest fan,

and

To my fiancé, and soon-to-be husband, Jon,

for the new life we start tomorrow!!



# TABLE OF CONTENTS

Abstract.....	3
Acknowledgments.....	5
Table of Contents.....	7
List of Figures.....	13
List of Tables.....	19
Nomenclature.....	21
Acronyms.....	23
Chapter 1.....	25
1.1 Attitude Sensors for Satellites and Spacecraft.....	25
1.2 Types of Attitude Sensors.....	26
1.2.1 Earth Sensor.....	26
1.2.2 Sun Sensor.....	27
1.2.3 Magnetometers.....	28
1.2.4 Global Positioning System Receivers.....	29
1.2.5 Star Trackers.....	31
1.3 The Next Generation of Star Trackers.....	36
Chapter 2.....	41
2.1 Processor Selection.....	43

---

2.2 Imager Selection .....	44
2.2.1 CID Imagers.....	44
2.2.2 CCD Imagers .....	45
2.2.3 CMOS Imagers .....	46
2.3 Optics Design.....	49
2.4 Hardware Mounting.....	53
Chapter 3.....	55
3.1 Stellar Signal.....	55
3.2 Noise Variations.....	58
3.2.1 Signal Shot Noise.....	59
3.2.2 Dark Current Noise .....	59
3.2.3 Background Noise.....	59
3.2.4 Quantization Noise.....	60
3.2.5 Readout Noise.....	61
3.2.6 Combining Different Noise Types.....	62
3.3 Signal-to-Noise Ratio.....	63
Chapter 4.....	65
4.1 Image Thresholding .....	65
4.2 Image Centroiding .....	67

---

4.2.1 Formation of Star Clusters .....	67
4.2.2 LIST and FAR-MST Centroiding Techniques .....	69
Weighted Sum Technique.....	69
Measuring Centroid Uncertainty.....	71
Maximum Likelihood Estimator Technique.....	73
Performance Comparison Between WS and MLE Centroiders.....	77
4.2.3 Predictive Centroiding .....	84
Chapter 5.....	87
5.1 NASA Sky2000 Star Catalog.....	87
5.2 Stellar Matching Techniques .....	89
5.3 Single Star Matching.....	90
5.4 Pattern Matching.....	93
5.4.1 The Angle Method and Pivoting.....	93
5.4.2 Triangle Matching Algorithms .....	95
Spherical Triangle Pattern Matching .....	95
Planar Triangle Pattern Matching .....	97
5.4.3 Rate Matching.....	98
5.4.4 Selecting Stars to Process .....	98
5.5 LIST and FAR-MST Matching Techniques .....	99

---

5.5.1 Pair Match .....	100
5.5.2 Triangle Match .....	100
5.5.3 Rate Match .....	104
5.5.4 N-vertex Match .....	105
Creating a Reduced Star Catalog .....	105
Double Star Reduction .....	107
Projection of Stars onto the Celestial Sphere.....	108
The k-vector .....	109
The n-vertex Match Algorithm .....	111
Chapter 6 .....	115
6.1.1 Wahba's Loss Function.....	115
6.1.2 The Predictive Attitude Determination Algorithm .....	116
6.1.3 The q-method and the QUEST Algorithm.....	117
6.1.4 The Singular Value Decomposition Algorithm .....	122
6.1.5 The Fast Optimal Attitude Matrix Algorithm.....	124
Chapter 7 .....	127
7.1 Centroider Performance .....	129
7.1.1 Centroider Speed.....	129
7.1.2 Centroider Accuracy .....	133



---

7.1.3 Attitude Quaternion Uncertainty.....	144
7.2 Triangle Match Performance .....	147
7.3 Rate Match Performance.....	151
7.4 Attitude Quaternion Comparison Using n-vertex Match.....	159
Chapter 8.....	165
8.1 Conclusions.....	165
8.1.1 The Optimal Centroider .....	165
8.1.2 Pattern Matching.....	167
8.1.3 Attitude Determination .....	170
8.2 Future Work.....	170
8.2.1 Quantifying Multiple Star Pairs .....	170
8.2.2 Additional Benchmarking.....	171
8.2.3 Dynamic Thresholding.....	172
8.2.4 Predictive Centroiding .....	172
8.2.5 Reducing Error Propagation in Frame-to-Frame Pattern Matching .....	173
8.2.6 Attitude determination .....	173
Appendix A.....	175
Definition.....	175
Quaternion Multiplication.....	175

Quaternion Inverse..... 176

Frame Rotation vs. Vector Rotation ..... 176

References..... 179

# LIST OF FIGURES

Figure 1.1 Goodrich Corporation Multi-Mission Earth Sensor Model 13-410 <sup>[3]</sup> .....	27
Figure 1.2 AeroAstro, Incorporation Medium and Coarse Sun Sensors <sup>[5]</sup> .....	28
Figure 1.3 Voyager Spacecraft with Magnetometer Boom Extended <sup>[8]</sup> .....	29
Figure 1.4 GPS Satellite Configuration <sup>[10]</sup> .....	30
Figure 1.5 GPS Receiver Block Diagram <sup>[9]</sup> .....	31
Figure 1.6 Star Scanner Example - Galileo Spacecraft <sup>[15]</sup> .....	33
Figure 1.7 Gimbaled Star Tracker Example; Top Left – SR-71, Top Right – RC-135, Bottom – B-2 <sup>[16, 17, 18]</sup> .....	34
Figure 1.8 Astro-1 Observatory aboard STS-35 <sup>[20]</sup> .....	34
Figure 1.9 Comparison of LIST and FAR-MST Attitude Determination <sup>[22]</sup> .....	37
Figure 2.1 ASTROS Tracker Block Diagram <sup>[30]</sup> .....	42
Figure 2.2 LIST and FAR-MST Star Tracker Block Diagram <sup>[29]</sup> .....	43
Figure 2.3 Fillfactory IBIS5-A-1300 CMOS Chip <sup>[40]</sup> .....	49
Figure 2.4 Illustration of Defocusing Effects Due to Field Curvature .....	52
Figure 2.5 Implementing a Double Gauss Lens to Remove Field Curvature <sup>[46]</sup> .....	53
Figure 2.6 ASTROS Tracker Hardware Assembly <sup>[30]</sup> .....	54
Figure 2.7 LIST Mount <sup>[47]</sup> .....	54
Figure 3.1 Intensity Profile of the Airy Function <sup>[48]</sup> .....	55

---

Figure 3.2 Airy Disk with Outer Rings Digitally Enhanced to Make More Easily Visible <sup>[49]</sup> ....	56
Figure 3.3 Gaussian Function with Bivariate Normal Distribution.....	57
Figure 3.4 Standard Normal Distribution Depicting the Percentage that SNR Will Exceed $1\sigma$ , $2\sigma$ , and $3\sigma$ Amounts of Noise .....	63
Figure 4.1 Filtered Image with Threshold Set at SNR = 1 .....	66
Figure 4.2 Bright Star .....	68
Figure 4.3 Dim Star.....	69
Figure 4.4 Centroid Accuracy with No Noise .....	71
Figure 4.5 Centroid Accuracy for Various SNR.....	73
Figure 4.6 Simulated Noisy Image for $M_V = 1$ .....	78
Figure 4.7 Comparison of WS and MLE Centroiding Errors for $M_V = 1$ .....	79
Figure 4.8 Simulated Noisy Image for $M_V = 2$ .....	79
Figure 4.9 Comparison of WS and MLE Centroiding Errors for $M_V = 2$ .....	80
Figure 4.10 Simulated Noisy Image for $M_V = 3$ .....	80
Figure 4.11 Comparison of WS and MLE Centroiding Errors for $M_V = 3$ .....	81
Figure 4.12 Simulated Noisy Image for $M_V = 4$ .....	81
Figure 4.13 Comparison of WS and MLE Centroiding Errors for $M_V = 4$ .....	82
Figure 4.14 Simulated Noisy Image for $M_V = 5$ .....	82
Figure 4.15 Comparison of WS and MLE Centroiding Errors for $M_V = 5$ .....	83
Figure 5.1 Average Number of Stars for Given $M_V$ and FOV .....	88

---

Figure 5.2 Minimum Number of Stars for Given $M_V$ and FOV .....	88
Figure 5.3 Difference Between Assigned $M_V$ and Calculated $M_V$ .....	91
Figure 5.4 Stars to Match from Image 1 .....	102
Figure 5.5 Candidate Matching Stars from Image 2.....	102
Figure 5.6 Candidate Star Matrix.....	103
Figure 5.7 Row Reduction of Candidate Star Matrix .....	104
Figure 5.8 All Stars in NASA SKY2000 Master Star Catalog Brighter than $M_V6$ .....	106
Figure 5.9 Reduced Version of NASA SKY2000 Master Star Catalog .....	107
Figure 5.10 Plot of $A$ .....	109
Figure 5.11 Linear Approximation of Plot of $A$ .....	110
Figure 5.12 N-Vertex Match Example with 4 Stars .....	112
Figure 5.13 Results of N-Vertex Match Example .....	113
Figure 7.1 Centroiding Execution Times for Noise Threshold of $1\sigma_{\text{noise}}$ .....	129
Figure 7.2 Centroiding Execution Times for Noise Threshold of $1.5\sigma_{\text{noise}}$ .....	130
Figure 7.3 Centroiding Execution Times for Noise Threshold of $2\sigma_{\text{noise}}$ .....	130
Figure 7.4 Centroiding Execution Times for Noise Threshold of $3\sigma_{\text{noise}}$ .....	131
Figure 7.5 Centroiding Execution Times for Noise Threshold of $4\sigma_{\text{noise}}$ .....	131
Figure 7.6 Image 1 - starCentroid2 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	133
Figure 7.7 Image 1 – starCentroid3 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	134

---

Figure 7.8	Image 2 - starCentroid2 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	134
Figure 7.9	Image 2 – starCentroid3 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	135
Figure 7.10	Image 3 - starCentroid2 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	135
Figure 7.11	Image 3 – starCentroid3 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	136
Figure 7.12	Image 4 - starCentroid2 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	136
Figure 7.13	Image 4 – starCentroid3 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	137
Figure 7.14	Image 5 - starCentroid2 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	137
Figure 7.15	Image 5 – starCentroid3 Positioning Accuracy for Noise Threshold of $1\sigma_{\text{noise}}$ .....	138
Figure 7.16	Image 1 - starCentroid2 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	139
Figure 7.17	Image 1 – starCentroid3 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	139
Figure 7.18	Image 2 - starCentroid2 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	140
Figure 7.19	Image 2 – starCentroid3 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	140
Figure 7.20	Image 3 - starCentroid2 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	141
Figure 7.21	Image 3 – starCentroid3 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	141
Figure 7.22	Image 4 - starCentroid2 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	142
Figure 7.23	Image 4 – starCentroid3 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	142
Figure 7.24	Image 5 - starCentroid2 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	143
Figure 7.25	Image 5 – starCentroid3 Positioning Accuracy for Noise Threshold of $3\sigma_{\text{noise}}$ .....	143
Figure 7.26	Uncertainty in Image Attitude Quaternion for Noise Threshold of $1\sigma_{\text{noise}}$ .....	144

---

Figure 7.27	Uncertainty in Image Attitude Quaternion for Noise Threshold of $1.5\sigma_{\text{noise}}$ .....	145
Figure 7.28	Uncertainty in Image Attitude Quaternion for Noise Threshold of $2\sigma_{\text{noise}}$ .....	145
Figure 7.29	Uncertainty in Image Attitude Quaternion for Noise Threshold of $2.5\sigma_{\text{noise}}$ .....	146
Figure 7.30	Uncertainty in Image Attitude Quaternion for Noise Threshold of $3\sigma_{\text{noise}}$ .....	146
Figure 7.31	Triangles Selected for Matching Using the Triangle Match Algorithm.....	149
Figure 7.32	Maximum Attitude Quaternion Uncertainty Using Triangle Match Algorithm ....	150
Figure 7.33	Rate Match Between Images 2 and 3 .....	152
Figure 7.34	Rate Match Between Images 3 and 4 .....	152
Figure 7.35	Rate Match Between Images 4 and 5 .....	153
Figure 7.36	Rate Match Between Images 5 and 6 .....	153
Figure 7.37	Rate Match Between Images 6 and 7 .....	154
Figure 7.38	Rate Match Between Images 7 and 8 .....	154
Figure 7.39	Rate Match Between Images 8 and 9 .....	155
Figure 7.40	Rate Match Between Images 9 and 10 .....	155
Figure 7.41	Rate Match Between Images 10 and 11 .....	156
Figure 7.42	Rate Match Between Images 11 and 12 .....	156
Figure 7.43	Rate Match Between Images 12 and 13 .....	157
Figure 7.44	Rate Match Between Images 13 and 14 .....	157
Figure 7.45	Maximum Attitude Quaternion Uncertainty Using Rate Match Algorithm.....	158

---

Figure 7.46 Pentagon Selected for Matching Using the n-vertex Match Algorithm with $n = 5$	161
Figure 7.47 Uncertainty in Image Attitude Quaternion for n-vertex Match when $n = 3$ .....	162
Figure 7.48 Uncertainty in Image Attitude Quaternion for n-vertex Match when $n = 4$ .....	162
Figure 7.49 Uncertainty in Image Attitude Quaternion for n-vertex Match when $n = 5$ .....	163
Figure 8.1 Comparison of Triangle Match and Rate Match Quaternion Uncertainties in Roll Direction .....	168
Figure 8.2 Dynamic Implementation of Frame-to-Frame Pattern Matching Algorithms.....	170



# LIST OF TABLES

Table 1.1 Attitude Sensor Goals .....	25
Table 1.2 Comparison of Current Attitude Sensors.....	35
Table 1.3 LIST and FAR-MST Operational Goals <sup>[23, 24]</sup> .....	38
Table 2.1 Comparison of Existing Star Trackers and FAR-MST Star Tracker <sup>[25, 26, 27, 28, 29]</sup> .....	41
Table 2.2 Blackfin ADSP-BF533 Properties <sup>[31]</sup> .....	43
Table 2.3 Comparison of Various CMOS Imagers <sup>[,,,,,]</sup> .....	48
Table 2.4 Lens Comparison .....	50
Table 4.1 Centroider Comparison Image Parameters .....	77
Table 4.2 Computational Time Requirements for WS and MLE Centroiders .....	83
Table 5.1 Required $M_V$ for a Given Minimum Number of Stars and FOV .....	89
Table 5.2 Parameters for $M_V$ Trade Study.....	91
Table 5.3 Statistical $M_V$ Data for 500 Stars .....	92
Table 5.4 Results of S Matrix Construction.....	113
Table 7.1 Parameter Settings for Centroider Performance Tests.....	127
Table 7.2 Weighted Sum Centroider Comparison Statistics .....	132
Table 8.1 Optimal Centroider Based on Speed and Accuracy Performance Results .....	166
Table 8.2 Time Comparisons for Matching Algorithms.....	168



# NOMENCLATURE

$a$	=	lens aperture (cm)
$DK$	=	number of dark current noise electrons (electron count)
$f$	=	lens focal length (mm)
$FF$	=	imager fill factor (%)
$FFSR$	=	product of fill factor and spectral response (A/W)
$fullWell$	=	maximum number of electrons that one imager pixel can contain (electron count)
$g$	=	image sensor gain (electron count/ADU)
$hFOV$	=	field of view in the horizontal direction (°)
$M_V$	=	visual magnitude
$\sigma_{star}$	=	width of Gaussian approximating the point spread function of a star (pixels)
$\sigma_{noise}$	=	standard deviation of the noise (electron counts)
$pp_x$	=	pixel pitch in the x-direction ( $\mu\text{m}$ )
$pp_y$	=	pixel pitch in the y-direction ( $\mu\text{m}$ )
$QE$	=	quantum efficiency (%)
$RO$	=	number of readout noise electrons (electron counts)
$SI$	=	number of stellar electrons (electron counts)
$t_{shutter}$	=	star tracker integration time (s)

---

$v_i$  = unit vector defining the position of the  $i^{\text{th}}$  star

$v_{FOV}$  = field of view in the vertical direction ( $^{\circ}$ )

# ACRONYMS

AD	=	analog-to-digital
ADC	=	analog-to-digital converter
ADU	=	analog-to-digital unit
CCD	=	charge-coupled device
CID	=	charge-induction device
CMOS	=	complementary metal-oxide-semiconductor
FAR-MST	=	fast angular rate miniature star tracker
FOV	=	field of view
GPS	=	Global Positioning System
LIS	=	lost-in-space
LIST	=	lightweight inexpensive star tracker
MMACS	=	Million Multiply Accumulate Cycles per Second
PSF	=	point spread function
SNR	=	signal-to-noise ratio



# Chapter 1

## BACKGROUND AND MOTIVATION

### 1.1 Attitude Sensors for Satellites and Spacecraft

All satellites and spacecraft require onboard attitude sensors to ensure maintenance of the projected trajectory. Errors in attitude sensing and/or tracking can result in significant damage, or in the worst case, complete loss of the spacecraft. Ideal attitude sensors can function independently from other onboard systems and can report current position and rate information instantaneously. An attitude sensor is generally designed to work in conjunction with an attitude controller. Once the attitude sensor detects an undesired deviation in course heading or speed, it signals the amount of deviation and the type of correction needed to the controller; the controller is then responsible for firing thrusters or other actuators to perform the corrective maneuver to return the spacecraft to its appropriate route.

A wide variety of attitude sensors exist, but their particular characteristics depend largely on the intended application. Nonetheless, the general design objectives of all attitude sensors are the same, as displayed in Table 1.1.

**Table 1.1 Attitude Sensor Goals**

<b>Maximize</b>	<b>Minimize</b>
accuracy	mass
reliability	size
wide-angle capture ability	power consumption
number of axes that can provide information	time delay in information relay
insensitivity to celestial bodies other than the one(s) it seeks	operating limits
lifetime	cost

Unfortunately, most attitude sensors have to prioritize between their desired success rates, computational speed and storage allotments, and hardware parameters. As a result, attitude

sensors that are most appropriate for one particular mission type, may be very different from attitude sensors applicable for a different mission type. However, because of the wide spread of satellite and spacecraft mission profiles in comparison to the number of attitude sensor types, individual missions are forced to rank which attitude sensor properties are most desirable and select one from those currently available.

## 1.2 Types of Attitude Sensors

There are numerous types of attitude sensors, but among the most popular are Earth sensors, Sun sensors, magnetometers, Global Positioning System receivers, and star trackers. Detailed below are brief introductions to these types of attitude sensors, including their respective purposes, hardware properties, and limitations.

### 1.2.1 Earth Sensor

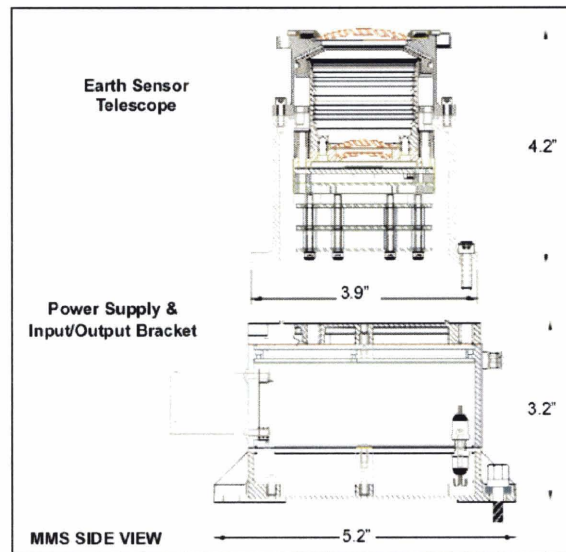
Earth sensors, also known as horizon scanners, determine spacecraft attitude relative to the Earth's horizon. They are generally used for space navigation, communication, and weather reports. Earth sensors have flown on the United States Mercury and Gemini manned spacecrafts and are currently implemented onboard various aircraft.

Earth sensors are appropriate for spacecraft and satellites within a reasonable proximity to the Earth. However, the crafts especially near the Earth can have 40% of their field of view (FOV) filled by the Earth itself, which makes it difficult to obtain accurate attitude information based on the Earth as a whole. Instead, these sensors locate the Earth's horizon and use it as a means of attitude determination.<sup>[1]</sup>

It has been shown that the position of Earth's horizon is the least ambiguous in the spectral region near  $15\mu\text{m}$  in the infrared. The spectral region of choice for most Earth sensors is the  $\text{CO}_2$  band because it has the most uniform intensity across the Earth.<sup>[2]</sup> One advantage of working in this realm is that the sensors can work just as effectively during the night as during the day and that they are naturally less susceptible to reflected light off of the craft than those sensors that work in the visible spectrum.



Earth sensors have four basic components: a scanning mechanism, an optical system, a radiance detector, and signal processing electronics.<sup>[1]</sup> An Earth sensor designed by Goodrich Corporation is displayed in Figure 1.1.<sup>[3]</sup>



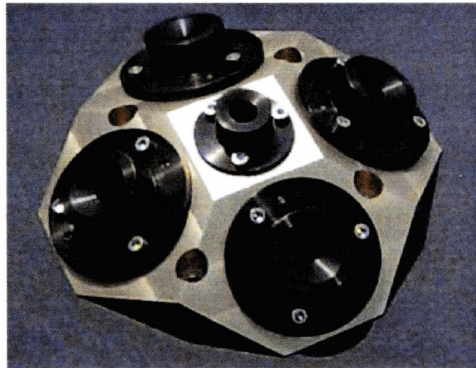
**Figure 1.1 Goodrich Corporation Multi-Mission Earth Sensor Model 13-410<sup>[3]</sup>**

Earth sensors do have several significant limitations. They have been known to perform poorly when high-altitude cold clouds are present or during solar interference, especially when close to the horizon.<sup>[4]</sup> Therefore, sufficient Sun rejection capability is a primary requirement for most Earth sensors. Additionally, uncertainties in temperature and altitude of the sensor are two other common sources of error.

## 1.2.2 Sun Sensor

Sun sensors are similar to Earth sensors, except they provide attitude information relative to the Sun, as opposed to the Earth. Apart from attitude determination, they are also used to protect delicate onboard equipment and position solar power arrays.<sup>[5]</sup> Solar sensors hold the advantage over Earth sensors in that the angular radius of the Sun is nearly Earth-orbit independent and small enough that for most missions a point-source approximation can be applied.<sup>[5]</sup>

There are three primary categories of Sun sensors: analog sensors, Sun presence sensors, and digital sensors. Due to the fact that Sun sensors are used in a wide range of applications, their fields of view (FOVs) range from several arc-minutes to a full 360°. An important advantage of Sun sensors is that they generally require no onboard power systems because their satellites or spacecraft use the Sun as their power source. Figure 1.2 shows an example of four medium Sun sensors surrounding a coarse Sun sensor designed by AeroAstro, Incorporated.<sup>[6]</sup>



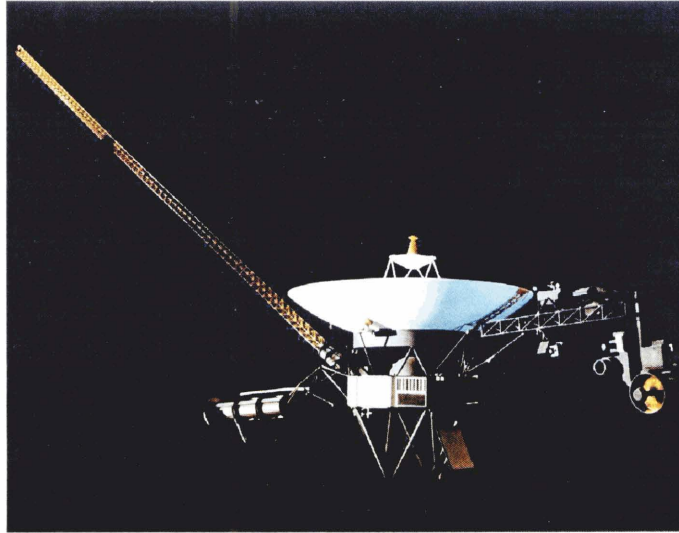
**Figure 1.2 AeroAstro, Incorporation Medium and Coarse Sun Sensors<sup>[5]</sup>**

The primary disadvantage of solar sensors is that satellites and spacecraft implementing them must continuously be concerned with the orientation and time evolution of the Sun vector in the body coordinate frame. In other words, the Sun must always remain in the FOV for the attitude sensor to be useful. This can pose strong limitations on mission flexibility. Another disadvantage is due to the natural occurrence of solar aging, which is caused by the persistent exposure to solar ultraviolet radiation. This can only be reduced through radiation shields and light baffles, which increase the mass requirements of the sensor.

### **1.2.3 Magnetometers**

Magnetometers are vector sensors that measure the orientation of satellites and spacecraft relative to a particular magnetic field, usually that of the Earth. These sensors run alternating currents through three mutually perpendicular coil-wound rods to magnetize the rods. The current is applied in one direction, and then in the opposite direction, and the orientation of the rods relative to the Earth's magnetic field is indicated by an imbalance in the alternating current output from the coils with respect to zero voltage.<sup>[4]</sup>

Magnetometers have two components: a magnetic sensor and an electronics unit that transforms the sensor measurements into a useable format.<sup>[7]</sup> Figure 1.3<sup>[8]</sup> depicts a magnetometer in use aboard the Voyager spacecraft, which studied the outer planets of Jupiter, Saturn, Uranus, and Neptune in the 1970s and 1980s.



**Figure 1.3 Voyager Spacecraft with Magnetometer Boom Extended<sup>[8]</sup>**

There are several significant limitations to magnetometers. First, due to the nature of how the rods are used to determine orientation, magnetometers can only determine attitude information about two axes; an additional sensor is necessary to obtain information about the third axis. Second, the use of the Earth's magnetic field becomes problematic because it is not completely known, and the models used to predict its magnitude and direction at the satellite or spacecraft's location are subject to substantial errors.<sup>[7]</sup> Additionally, because the Earth's magnetic field strength decreases with distance from the Earth as  $1/r^3$ , magnetometers are generally limited to operate below an altitude of 1000km.; above this altitude, residual craft magnetic biases tend to dominate the total magnetic field measurement.<sup>[7]</sup>

### **1.2.4 Global Positioning System Receivers**

The Global Positioning System (GPS) is a constellation of 27 Earth-orbiting satellites, 24 active and 3 spares, that was first designed and implemented by the United States Air Force in the 1970s as a means for navigation. The satellites are arranged in 6 orbital planes inclined at

55° with right ascension of the ascending nodes separated by 60°, as shown in Figure 1.4<sup>[9,10]</sup> These orbits are aligned so that at least 4 satellites are always within the line of sight from nearly any location on the Earth.

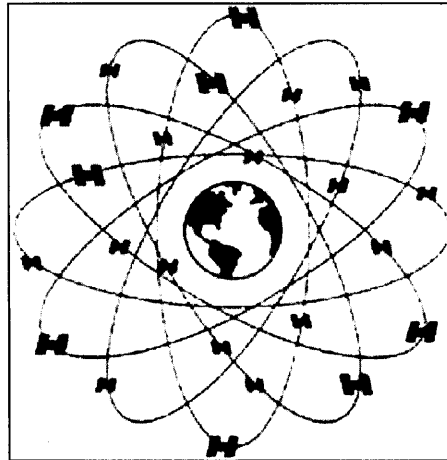


Figure 1.4 GPS Satellite Configuration<sup>[10]</sup>

GPS receivers are attitude sensors that calculate their current position, which includes latitude, longitude, and elevation, as well as the precise time aboard the satellites. The objective of a GPS receiver is to locate 4 of the GPS satellites, determine the distances to each, and then deduce its own location; this is accomplished using the mathematical principle known as *trilateration*, which is the process of determining relative positions of objects using the geometry of triangles.<sup>[11]</sup> GPS receivers contain the following components/functions: antenna, preamplifier, reference oscillator, frequency synthesizer, downconverter, intermediate frequency (IF) section, signal processing, and navigation processing, as displayed in Figure 1.5.<sup>[9]</sup> These elements work together to acquire signals from the satellites, filter noise and interference, and finally output the receiver's position.

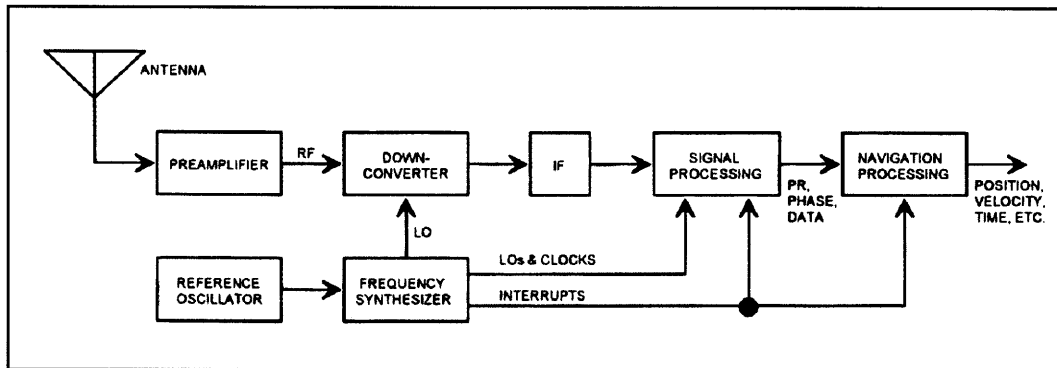


Figure 1.5 GPS Receiver Block Diagram<sup>[9]</sup>

Although GPS receivers have the potential for producing highly accurate positioning estimates, there are several disadvantages to using them. One of the largest contributors to receiver inaccuracies results from changing atmospheric conditions. Varying conditions unpredictably alter the speed of the GPS signals as they pass through the Earth's ionosphere, which is the atmospheric layer ionized by solar radiation.<sup>[12]</sup> The only means of alleviating this is to wait until satellites are more directly overhead, where the signal travels through less distance of the ionosphere. Signals can also be degraded because of multi-path reflections of the radio signals off of the ground or surrounding structures such as buildings or mountains.<sup>[12]</sup> With regards to spacecraft attitude sensing, current GPS receivers are limited in that they are currently restricted for use below altitudes of approximately 10,000km.

### 1.2.5 Star Trackers

Star trackers are composed of three elements: an optics system that incorporates a pinhole, single, or double lens that enables the capture of stellar photons, a detector system generally composed of a charge-coupled device (CCD), charge-injection device (CID), or complementary metal oxide semiconductor (CMOS) onto which the star light is defocused, and an electronics processing unit that both digitizes and analyzes the stellar data. Star trackers measure stellar coordinates and compare them to known coordinates from either an onboard star catalog or from previous images. This comparison results in attitude information within the star tracker body frame, which can then be translated into the spacecraft body frame or to an inertial reference frame. In general, star trackers are the most accurate attitude sensors because they have the potential of achieving accuracies within several arc-seconds. Also, since they do not

---

rely on the positions of the Earth, Sun, magnetic fields, or external satellites, they are highly flexible attitude sensors.

The earliest star trackers required initial, coarse attitude estimates from other onboard attitude sensors. In the 1940s and early 1950s, star trackers were used in aircraft and missile guidance to provide celestial reference for position and bearing determination that was available during the day and night.<sup>[13]</sup> In the 1950s and 1960s, star trackers were used in conjunction with gyro-stabilizing platforms to more accurately align rockets prior to launch. Star trackers continue to be used in the stabilization of Earth-orbiting, Lunar, and planetary satellites and spacecraft. Fortunately, with increasing duration and more complicated mission requirements, solid state imaging devices, such as the CCD, CID, and CMOS imagers, have been incorporated, which have drastically improved reliability and attitude information accuracy.

There are three types of star trackers: star scanners, which use the craft's rotation to dictate the searching and scanning function, gimbaleed star trackers, which search for and acquire stars via mechanical movement, and fixed head star trackers, which electronically search for and track stars over limited FOVs.<sup>[14]</sup> A star scanner was used onboard the Galileo spacecraft, in addition to Sun sensors and gyroscopes, to provide the most accurate and final attitude.<sup>[15]</sup> This is highlighted in Figure 1.6 by the red arrow.



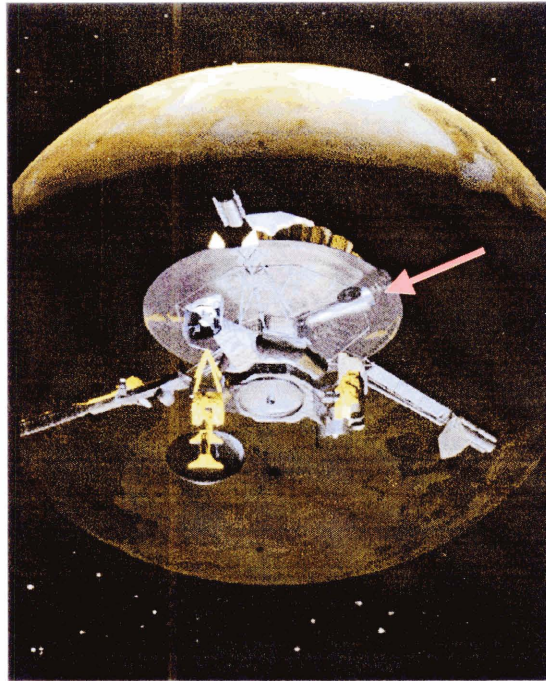


Figure 1.6 Star Scanner Example - Galileo Spacecraft<sup>[15]</sup>

The United States Air Force SR-71 Blackbird, RC-135 Rivet Joint surveillance aircraft, and the B-2 Spirit, displayed in Figure 1.7<sup>[16, 17, 18]</sup>, all implement gimballed star trackers.<sup>[19]</sup> In the B-2, the star tracker observations are made through a 7in window located in the left wing.



Figure 1.7 Gimbaled Star Tracker Example; Top Left – SR-71, Top Right – RC-135, Bottom – B-2<sup>[16, 17, 18]</sup>

The Astro-1 observatory aboard STS-35, depicted in Figure 1.8, relied on a fixed head star tracker (FHST).<sup>[20]</sup> The FHST was supposed to acquire, track, and identify stars of known visual magnitude, but the star tracker erroneously calculated the visual magnitudes to be lower than what they were in reality, severely hampering the outcome of the mission.<sup>[21]</sup>

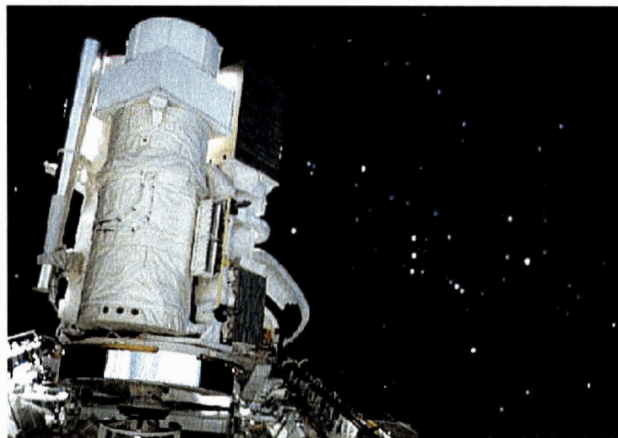


Figure 1.8 Astro-1 Observatory aboard STS-35<sup>[20]</sup>



As evident in these examples, star trackers have been implemented onboard some of the most sophisticated aircraft, spacecraft, and satellites that have accomplished impressive feats. Star trackers were selected as the primary attitude sensors in these missions because of their potential for high sensing and tracking accuracy. The unfortunate tradeoff for high precision is significant increases in mass, computational expense, and power consumption. Regardless, all of these missions were multi-million dollar ventures where the most advanced attitude sensors were not only appropriate, but were crucial, and so these tradeoffs were relatively inconsequential.

However, there are numerous current missions, many of which have a high potential for significant scientific advancement, that cannot afford the large increases in mass, computational expense or power consumption. Whether these missions emanate from small start-up companies or large government proposals, their objectives remain high. The pervasive requirement to drive down overall weight and volume of satellites and spacecraft in order to minimize propulsive expenses for launch will require attitude sensors that are small enough to accompany their reduced hardware sizes, while simultaneously maintaining their effectiveness in attitude determination. The second through fifth rows of Table 1.2 describe the five attitude sensors discussed above. It is clear from this table that current star trackers can perform under the most variable operating conditions and can provide the most accurate attitude information, but these advantages come at the expense of requiring the greatest size, mass, power consumption, and cost.

**Table 1.2 Comparison of Current Attitude Sensors**

Sensor	Num. Axes	Operating Angular Range	Orbits	Size/Mass/Power/Cost	Accuracy/Angular Rate
Earth Sensor	2	Narrow: Nadir $\pm \sim 10^\circ$	Circ., small alt. range	Medium	Medium/High
Sun Sensor	2	Narrow: Sun $\pm \sim 30^\circ$ None in eclipse	Any	Low	Low/High
Magnetometer	2	Full sphere, but need magnetic clean sat.	LEO only	Low	Low/High
Available GPS Attitude Receivers	3	Medium: Zenith $\pm \sim 90^\circ$	Alt < 10k km No GEO	Medium	Medium/High
Available Star Trackers	3	Wide: All except Earth/Sun	Any	High	High/Low
<b>New Opportunity Star Tracker</b>	<b>3</b>	<b>Wide: All except Earth/Sun</b>	<b>Any</b>	<b>Low</b>	<b>Medium/High</b>

The majority of missions are not at the same level as those of Galileo or Astro-1, and so a new method of attitude sensing that can provide low cost, high performance micro-satellites with appropriate means of attitude determination is currently desirable. This goal is best described by Junkins (2003):

“In order to accelerate the evolution of faster, better, cheaper spacecraft, it is evident that greatly enhanced general-purpose attitude determination methods are needed. There is a clear need for lightweight, accurate, reliable, and inexpensive systems for spacecraft attitude estimation. Moreover, the use of smaller satellites have more onboard processing as well as reduced ground computer support has encouraged the development of lightweight, autonomous star trackers. Such star trackers should be able to identify and track star patterns in arbitrary directions without prior information, additional sensor support, or ground processing.”<sup>[13]</sup>

To meet these requirements, new opportunity “micro” star trackers are currently being designed with properties specified in the final row of Table 1.2. These star trackers hope to provide medium to high attitude sensing and tracking accuracies in all three axes, while simultaneously minimizing their own size, mass, power consumption, and cost. Several existing micro star trackers include the Jet Propulsion Laboratory’s ASTROS star tracker, Ball Aerospace’s CT-600 series star trackers, Lockheed Martin’s Corning OCA improvement to the Lawrence Livermore National Laboratory Clementine star tracker, and Junkin’s DIGISTAR star tracker. Two specific star trackers, which are the focus of this paper, are described briefly below.

### **1.3 The Next Generation of Star Trackers**

Two next-generation star trackers, the Lightweight Inexpensive Star Tracker (LIST) and the Fast Angular Rate Miniature Star Tracker (FAR-MST), are currently under joint development by AeroAstro, Inc. and the Massachusetts Institute of Technology Space Systems Laboratory. The purpose of LIST and FAR-MST is to provide star tracker position and rate determination to small, highly maneuverable, low-cost spacecraft. Thus, LIST and FAR-MST propose a better balance between attitude determination accuracy and cost metrics, providing more affordable attitude sensors to smaller vehicles.

LIST and FAR-MST star trackers differ with respect to their attitude performance abilities, as described in Figure 1.9.<sup>[22]</sup> LIST is a simpler, minimum requirement star tracker capable of determining relative rate information. FAR-MST can achieve rate information during faster tumbles and also will provide lost-in-space (LIS) capability, by determining spacecraft orientation without prior attitude information

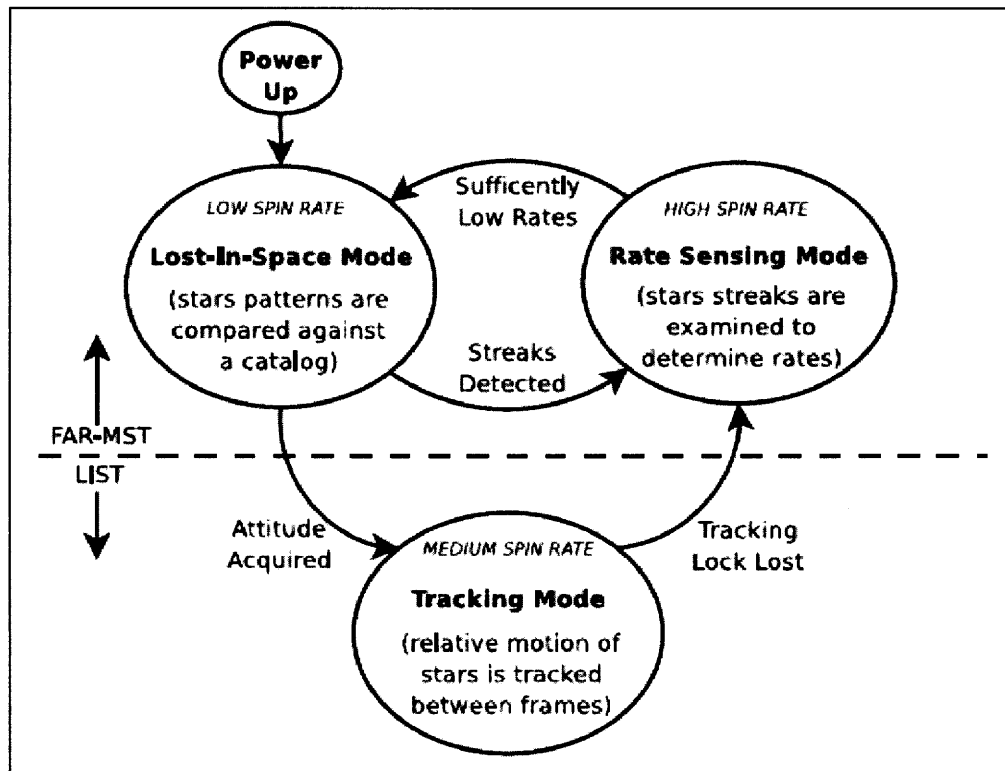


Figure 1.9 Comparison of LIST and FAR-MST Attitude Determination<sup>[22]</sup>

To meet physical requirements, appropriate constraints on optics, memory, and computational speed have been defined and included in the hardware selection and software design. The targeted hardware characteristics of FAR-MST include a mass below 1kg, dimensions smaller than 15cm x 8cm x 8cm, power consumption not exceeding 3W, and cost less than \$100k, as shown in Table 1.3.<sup>[23, 24]</sup> Since LIST has more lenient rate sensing requirements, the targeted characteristics for LIST are somewhat smaller than those for FAR-MST.

Table 1.3 LIST and FAR-MST Operational Goals<sup>[23, 24]</sup>

	LIST	FAR-MST
dimensions (cm <sup>3</sup> )	5.1 x 7.6 x 7.6	15 x 8 x 8
mass (kg)	0.3	0.9
power consumption (W)	< 1	3
cost (\$k)		100
limiting MV	4	4.5
stars tracked per frame	< 4	> 2
pitch/yaw accuracy (arcsec)	70	100
roll accuracy (arcsec)	1080	
max acquire rate (°/s)		4
max sensing rate (°/s)		15
max tracking rate (°/s)		10
update rate (Hz)	1	
FOV (°)	30	25 to 45
onboard star catalog (# stars)	none	< 500
supporting satellite weight (kg)	tens to hundreds	
mode of operation	relative tracking	LIS

Despite their restrictive specifications, LIST and FAR-MST star trackers will still be capable of performing attitude determination at 1Hz update rate with an accuracy of 100arcsec. LIST and FAR-MST will manage tumble rates up to 1°/s and 10°/s, respectively.

The primary software algorithms include functions to extract and centroid stars from image noise and to match star patterns between subsequent images or to a small on-board catalog. The current algorithms incorporate various ideas from open literature, but appropriate modifications and improvements have been made where applicable. Thresholding at low signal-to-noise is accomplished using a statistical pre-filter to address high noise content. Centroiding examines the stellar shape to optimally set shutter time at different rotation rates. Pattern matching between frames implements a ‘smart’ star selection technique to avoid wasted time on stars that might not appear in both images. Several algorithms, including the pair match, triangle match, and rate match, are used for frame-to-frame matching. Additionally, an n-vertex pattern match can be performed against a star catalog that has been reduced either based on the probability that the stars would be used or by existing attitude information. This variety in pattern matching algorithms enables the codes to be implemented in a dynamic manner based on

---

the number of stars present in the image and whether or not current rate information is available. This provides several modes of operation and graceful degradation of performance at higher angular rates. Finally, a recursive estimation technique is used to generate attitude quaternion solutions from the pattern matching results.



# Chapter 2

## STAR TRACKER HARDWARE

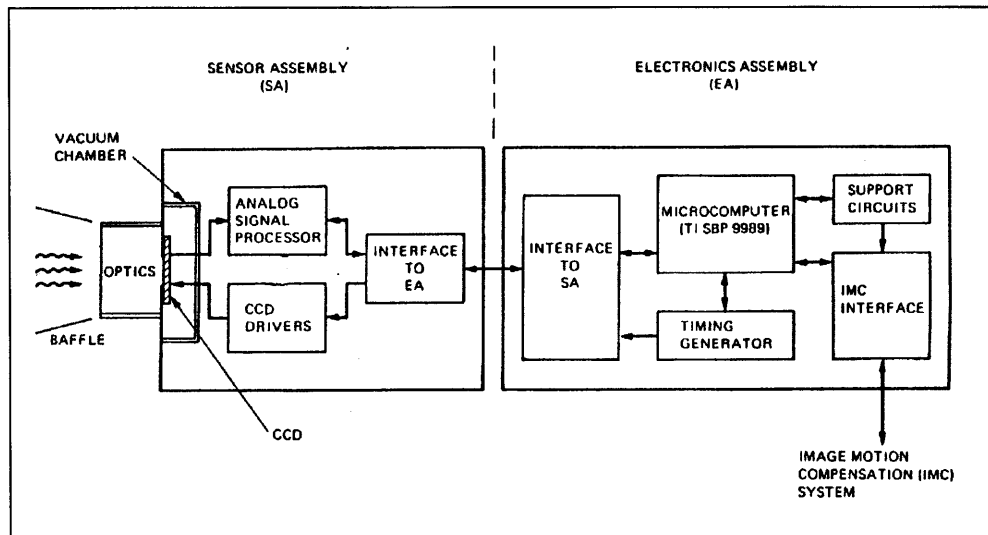
The principle goal of next-generation star trackers, such as LIST and FAR-MST, is to provide micro-satellites and spacecraft with accurate attitude information without the lofty mass, size, power consumption, and cost requirements that constrain most current star trackers. These goals are accomplished by selecting particular hardware components that are capable of meeting star tracker needs. Table 2.1 compares the FAR-MST star tracker with current star trackers designed by Goodrich, Ball Aerospace, SODERN, and Galileo.<sup>[25, 26, 27, 28, 29]</sup>

**Table 2.1 Comparison of Existing Star Trackers and FAR-MST Star Tracker**<sup>[25, 26, 27, 28, 29]</sup>

Star Tracker	Goodrich HD-1003	Ball CT-633	SODERN SED-16	Galileo AA-STR	AeroAstro FAR-MST
Size (cm)	41 x 16 x 11	34 x 19 x 19	29 x 17 x 16	18 x 12 x 12	<b>15 x 8 x 8</b>
Mass (kg)	3.54	2.85	3.0	1.1	<b>0.9</b>
Power (W)	10	10	12.8	3.8	<b>3</b>
Cost (\$k)	550	> 600	> 500	~ 300	<b>100</b>
Accuracy (arcsec 3 $\sigma$ )	25	40	70	98	<b>100</b>
Max Acquire Rate ( $^{\circ}$ /sec)	0.8	0.2	7.0	1.0	<b>4.0</b>
Max Tracking Rate ( $^{\circ}$ /sec)	1.4	10	20	N/A	<b>10</b>
Max Sensing Rate ( $^{\circ}$ /sec)	None	None	None	4	<b>10</b>

Star trackers are composed of three main hardware elements: optics, imager, and processor. The optics subsystem is used to collect and focus light onto the image plane. The imager digitizes the photons into electrons and passes them to the processor, which converts digitized electron signals into quantities that can be used in the pattern matching and attitude defining algorithms. These subsystems are assembled together in a manner similar to that shown below for the ASTROS star tracker in Figure 2.1.<sup>[30]</sup> The sensor assembly receives incoming light rays through a single lens, projects the photons onto the CCD, and generates the corresponding digital star-image pixel data. The electronics assembly receives the digitized data

and uses the processor and memory to run the image processing algorithms necessary for attitude determination.



**Figure 2.1 ASTROS Tracker Block Diagram<sup>[30]</sup>**

Figure 2.2 shows a simple schematic of how the hardware functions en bloc to produce attitude information for the satellite.<sup>[29]</sup> Stellar light rays enter the optics system and are focused on the imager. The processor then uses the image, pattern recognition software, and onboard star catalog to output the roll, pitch, and yaw position as attitude information.



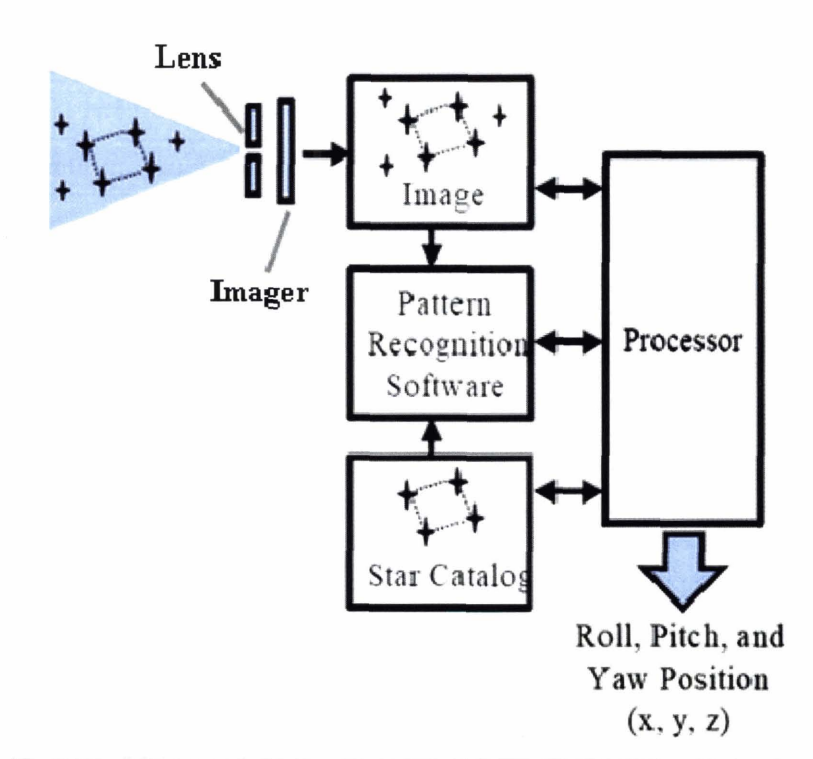


Figure 2.2 LIST and FAR-MST Star Tracker Block Diagram <sup>[29]</sup>

## 2.1 Processor Selection

A star tracker processor is used to convert the image stored by the sensor into information useful to the software that eventually outputs attitude information. The parameters of importance for LIST and FAR-MST star trackers are clock speed and power consumption. Several processors have been considered, including the Hitachi SH7065 processor and Analog Devices, Inc.'s Blackfin ADSP-BF533. The Blackfin BF533 has been selected for FAR-MST and its parameters of importance are displayed in Table 2.2.<sup>[31]</sup>

Table 2.2 Blackfin ADSP-BF533 Properties<sup>[31]</sup>

max clock speed (MHz)	750
max MMACS	1500
RAM (kBytes)	148
external memory bus (# bit)	16
core voltage (V)	0.8 to 1.26

## 2.2 Imager Selection

Star tracker imagers are responsible for the digital conversion of the photons collected by the lens into electrons that can be read by the processing unit. Current star trackers generally implement either charge-injection devices (CIDs), charge coupled devices (CCDs), or complementary metal oxide semiconductors (CMOS) detectors. The primary difference between these three types of imagers is the variation in signal readout methods, as is explained below.

### 2.2.1 CID Imagers

General Electric Company initiated the concept of charge-injection devices (CIDs). In 1972, General Electric's scientists used the photosensitive properties of silicon to develop a simple array of photosensitive capacitor elements, which eventually evolved into the first CID camera. Since the 1970s, CIDs have evolved to provide high resolution with blooming resistance, broad spectral response capabilities, and asynchronous operation modes for detection and measurement.

CIDs are composed of a two-dimensional array of charge-coupled MOS storage capacitors. They measure signal charge *in situ*<sup>[32]</sup>, meaning electron charges are not removed from the original pixels during readout. Although this type of reading is difficult, the charge remains intact following readout, thus preventing the data from being erased. Signal reading is generally accomplished by “sequential-injection” or “parallel-injection.”<sup>[33]</sup> When a clear pixel map is desired for a new image, electrodes in each pixel are temporarily switched to ground, “injecting” the charge into the silicon.<sup>[34]</sup>

Because of this unique data-readout process, CID imagers hold several important technical advantages over other imaging devices. CCDs and CMOS detectors suffer from low fill factors. However, since CIDs have a contiguous pixel structure, their fill factors are always 100%, which translates to higher image resolution. Because the readout process is nondestructive, CIDs have large degrees of exposure control. This means that low-light static images can be captured, and images can be exposed for long time periods, until an optimal exposure develops.<sup>[34]</sup> Solid state imaging, such as that provided by CCDs, is often challenged

by pixels “blooming” and “smearing,” caused by individual pixels reaching their maximum fullwell saturation levels and spilling their electrons into adjacent pixels during longer exposures at the brighter intensity levels. However, CIDs better tolerate bright intensity levels because electrons are confined to the pixels in which they were originally placed, since charge does not move from pixel to pixel in these imagers.

CIDs can implement several different types of progressive scanning modes. When the fastest capture rate is desired over full resolution, the image frame can be downsized. Also, particular regions of the FOV can be isolated to more quickly update specific windows of interest to the user. CIDs also have the ability to act in a “freeze-frame” mode to enable images to be viewed for shorter or longer periods of time.<sup>[34]</sup>

### 2.2.2 CCD Imagers

The charge-coupled device (CCD) was first developed by Willard Boyle and George Smith at AT&T Bell Labs in 1969. Further advances in CCD technology were made by Fairchild Semiconductor, RCA, and Texas Instruments, and the first commercial device was manufactured in 1974 by Fairchild Semiconductor. CCDs are known for their fast speed, high dynamic range, low noise, and excellent resolution capabilities. However, the primary disadvantage of CCD imagers is their high power consumption.

A charge-coupled device is composed of an array of photosensitive coupled capacitors embedded in an integrated circuit, where each capacitor, or pixel, accumulates and stores electric charge. Prior to digitization, charge must be moved from the original pixel locations to the end columns in their row where they are sent through an analog-to-digital converter (ADC) to be read. Readout times are small in comparison to integration times to prevent image smearing.

The CCD is commonly implemented in one of three different architectures: full-frame, frame-transfer, and interline.<sup>[35]</sup> The full-frame architecture uses a mechanical shutter instead of an electronic one, and all of its image area is active. The frame-transfer architecture is more expensive than the full-frame architecture because half of the silicon area on the imager is covered by an opaque mask. This opaque area provides an additional storage area for readout

while a new image can be simultaneously integrated in the active area. The primary disadvantage of a full-frame CCD is the increased power requirements of the mechanical shutter. The interline architecture covers every-other column of the imager with the opaque mask. This architecture is advantageous because it only requires charge to move one pixel over to reach a storage and readout region. The unfortunate tradeoff is that the overall fill factor and quantum efficiency are decreased to 50%. Modifications have been made to the interline architecture to improve the imager's fill factor by incorporating micro-lenses to direct light away from the opaque areas and focus it onto the active ones. In general, applications requiring maximum light capture, regardless of cost, power or processing time, should implement full-frame CCDs. However, applications requiring lower costs and lower power consumption should utilize interline CCDs.<sup>[35]</sup>

### 2.2.3 CMOS Imagers

Complementary metal-oxide-semiconductor (CMOS) circuits were developed by Frank Wanlass at Fairchild Semiconductor in 1963. CMOS imagers read image data through the use of transistors, located at each pixel, that amplify and transfer the charge. In terms of the reading process, CMOS imagers are more flexible than CCD imagers because each pixel in a CMOS imager can be individually read.<sup>[36]</sup> This enables a window-of-interest, or “windowing,” capability that is useful in image compression, motion detection, and target tracking.<sup>[37]</sup>

There are two types of CMOS imagers: ones with passive pixel sensors and ones with active pixel sensors. Passive pixel sensors are the oldest CMOS imagers and were first used in the 1960s. Photosites convert photons into electrical charges. The main drawback of passive pixel sensors is that high background noise is generated in the image. Active pixel sensors were designed to reduce the noise associated with passive pixel sensors. This is accomplished by implementing “active circuitry” to determine and cancel out the noise level at each pixel. Active pixel sensors allow for larger image arrays and enable higher resolution due to this noise cancellation ability while still consuming considerably less power than CCDs.

In general, the first CMOS imagers were prone to lower resolution and lower sensitivity in comparison to other solid state imagers. However, significant improvement has been made to

---

bring them almost up to par with CCD imagers. Although CMOS imagers still do not provide quite as much accuracy and flexibility as CCD imagers, CMOS imagers are significantly less expensive and are beginning to show capabilities comparable to CCD imagers.

One of the largest problems with CCDs is the fact that their hardware characteristics require them to be manufactured using specialized and expensive processes, which are used exclusively for CCD production.<sup>[38]</sup> However, CMOS imagers use the same processes and equipment as computer processing and memory chips, which are mass-produced all over the world. This standard fabrication ability is the foremost reason CMOS imagers are much less expensive than CCD imagers.

There are two disadvantages of CMOS detectors. First, CMOS chips have fill factors lower than CCDs, making them less sensitive to light and therefore requiring longer exposure times to capture the same number of photons. Second, they have higher noise levels than CCDs, requiring longer processing times to reduce the noise.

Despite these few shortcomings of CMOS imagers, their low cost, low power consumption, imbedded analog-to-digital (AD) conversion and amplification, and windowing capabilities make them the imager of choice for micro star trackers like LIST and FAR-MST. Table 2.3 compares six different CMOS detectors manufactured by three different companies.

Table 2.3 Comparison of Various CMOS Imagers<sup>[39, 40, 41, 42, 43, 44]</sup>

	Fillfactory		Micron		Kodak	
	IBIS4-1300	IBIS5-A-1300	MT9M001	MT9M413	KAC-9638	KAC-01301
active image area (mm <sup>2</sup> )			6.66 x 5.32	15.36 x 12.29	6.27 x 7.81	3.47 x 2.78
active array format (# pixels)	1280 x 1024	1280 x 1024	1280 x 1024	1280 x 1024	1288 x 1032	1284 x 1028
pixel size (μm <sup>2</sup> )	7 x 7	6.7 x 6.7	5.2 x 5.2	12 x 12	6 x 6	2.7 x 2.7
frame rate (fps)	7 (on chip ADC), < 23 (analog output)	< 27.5	30	< 500	18	16
shutter mode	electronic rolling	rolling, snapshot	electronic rolling	trueSNAP freeze-frame	rolling reset	continuous, single frame rolling
ADC	10 bit	10 bit	10 bit	10 bit	8 and 10 bit	10 bit
conversion gain (μV/e <sup>-</sup> )	< 16	17.6	< 15	13		
FF (%)	60	50		40	49	
FFSR (A/W)	0.165	0.16				
QE (%)	> 50	> 60	< 55	< 25		TBD
peak QE * FF (%)	> 30	30%				
sensitivity (V/Lx-s)	7	< 8.46	2.1	1600 bits/Lx-s		TBD
fullWell (# e <sup>-</sup> )	70000			63000		
dynamic range (dB)	< 100	< 100	68.2	59	57	49
dark current (pA/cm <sup>2</sup> )	344					
dark current rate (# of e <sup>-</sup> /s)	1055	410		50 mV/s		
power dissipation (mW)	535	175	363	< 500 at 500 fps, < 150 at 60 fps	150	< 100
operating temperature (°C)		0 to 60	0 to 70	-5 to +60	-10 to +50	-30 to +70

The imager selected for LIST and FAR-MST is Fillfactory's IBIS5-A-1300, which is displayed in Figure 2.3. This imager was chosen because of its low power consumption, high sensitivity, fast frame rate and large QE, as shown in Table 2.3.

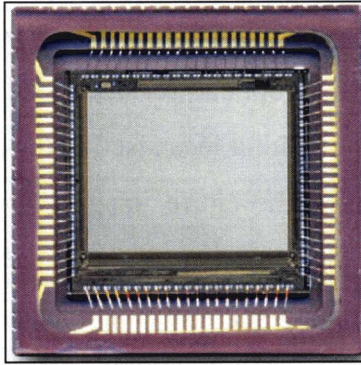


Figure 2.3 Fillfactory IBIS5-A-1300 CMOS Chip<sup>[40]</sup>

## 2.3 Optics Design

The primary function of a lens is to collect and focus light rays. The important parameters for consideration in imaging lenses are aperture, focal length, f-number, format, and resolution.

A lens' aperture is its diameter. Larger apertures correspond to larger lenses, and therefore have higher light collecting capabilities. The downside of large apertures is increased mass. In general for star trackers, the largest lens that does not exceed hardware constraints is desired to maximize light collection, allowing for dimmer stars to be processed at faster rates. A lens' focal length is the distance between the lens and the location of the focused image. Long focal lengths correspond to high magnifications and narrow FOVs, whereas short focal lengths correspond to low magnifications and wide FOVs.<sup>[45]</sup> The focal length and FOV of the lens and imager plane are related by

$$FOV = 2 \tan^{-1} \left( \frac{w}{2f} \right) \quad (2.1)$$

where  $w$  is the width of the imager and  $f$  is the focal length. A lens' f-number, or focal ratio, is the ratio of the lens' focal length to its aperture. Smaller f-numbers correspond to greater light collection ability while larger f-numbers correspond to greater latitude for focus.<sup>[45]</sup>

When selecting the proper star tracker optical and imaging hardware, a lens' format and resolution are important factors to consider when matching these subsystems together. The size

of the imager plane helps dictate the necessary size of the lens' format and vice-versa. For example, an imager having a ¼ inch optical format, which corresponds to a diagonal of approximately 4 mm, should be paired with a lens also having a ¼ inch format.<sup>[45]</sup> This ensures that the star tracker will make use of the entire imaging array, including the corners. The resolution of the lens should match the resolution of the imager, which can be measured by the number of microns contained in a single pixel. For example, an imager containing pixels that are 5 microns wide should be matched with a lens that can resolve 5 micron wide features.<sup>[45]</sup> Improperly matched optical and imaging subsystems can result in blurry images and, consequently, inaccurate attitude information.

Star trackers have the flexibility to implement several different lens types. Three lenses have been considered for LIST and FAR-MST, which include a pinhole, single lens, and double lens. Their respective properties are displayed in Table 2.4.

**Table 2.4 Lens Comparison**

Type	Advantages	Disadvantages
<b>Pinhole</b>	simple	small light gathering area → low light capturing ability
	no additional mass	→ longer exposure times required → higher SNR required
	no radiation yellowing	→ less capacity in roll direction
		open to radiation
		open to contaminants
<b>Single Lens</b>	larger light gathering area → higher light capturing ability → shorter exposure times required → lower SNR allowed → higher capacity in roll direction	additional mass required
	sensor protected	lens radiation yellowing
		curvature of fields effect
<b>Double Lens</b>	larger light gathering area → higher light capturing ability → shorter exposure times required → lower SNR allowed → higher capacity in roll direction	additional mass required
	sensor protected	lens radiation yellowing
	removal of curvature of fields effect	



---

The pinhole was initially considered for LIST because of its simplicity and low cost. However, testing showed that the sensing and tracking capabilities did not meet the performance requirements in roll for dim visual magnitudes. The single lens does meet performance requirements, but is problematic due to the bending of the light as it passes through the lens, which results in a field curvature.

When light enters a lens, it is refracted, and all light entering perpendicular to the lens is focused to a single point, known as the lens focal point, at a distance equal to the focal length. Light entering at angles offset from the perpendicular is focused to a point also at a distance equal to the focal length, but due to the angle offset, this point is not in line with the image plane, as displayed in Figure 2.4. Therefore, light rays entering the lens at any angles other than perpendicular to the lens are projected onto a curved image plane. This results in defocused images on the actual flat imager. For star trackers, this means that stars appear larger or smaller, depending on where they fall on the imager, which can have serious effects on pattern matching and image processing algorithms. The two-dimensional curvature-of-fields effect is illustrated in Figure 2.4.

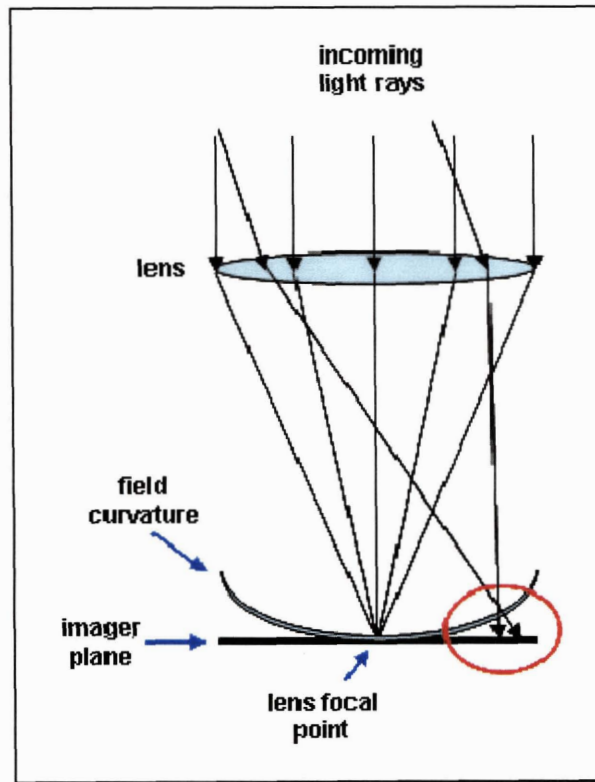


Figure 2.4 Illustration of Defocusing Effects Due to Field Curvature

A double lens system can be used to eliminate the field curvature. For example, a double Gauss lens refracts non-perpendicular light rays back by the appropriate angles so that all light is focused onto a flat image plane, which, therefore, prevents stellar distortion. This is illustrated in Figure 2.5.<sup>[46]</sup>

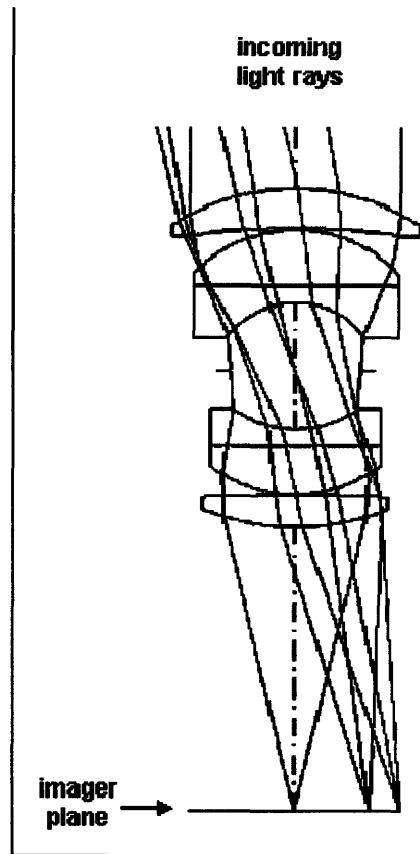


Figure 2.5 Implementing a Double Gauss Lens to Remove Field Curvature<sup>[46]</sup>

The primary constraint limiting the use of a double lens for LIST and FAR-MST star trackers is the resulting increased mass and size of the hardware system. Testing has shown that LIST star trackers can achieve their required accuracy using a single lens, and so a commercial off-the-shelf (COTS) single lens will be utilized in LIST. However, due to more rigid FAR-MST goals, a double lens will most likely be considered for FAR-MST star trackers.

## 2.4 Hardware Mounting

The processor, imager, and optics must be mounted together as a single unit to be attached to the satellite or spacecraft. The optics and imager are generally assembled using a C, S, CS, or X mount; the difference between these mounts is the number of threads and the back flange-to-image distance. An example of the imager and electronic assembly of the ASTROS star tracker is shown in Figure 2.6.<sup>[30]</sup> The ASTROS star tracker implements Texas Instrument's SBP 9989 microprocessor, RCA's 501 CCD imager, and a single lens optics system.

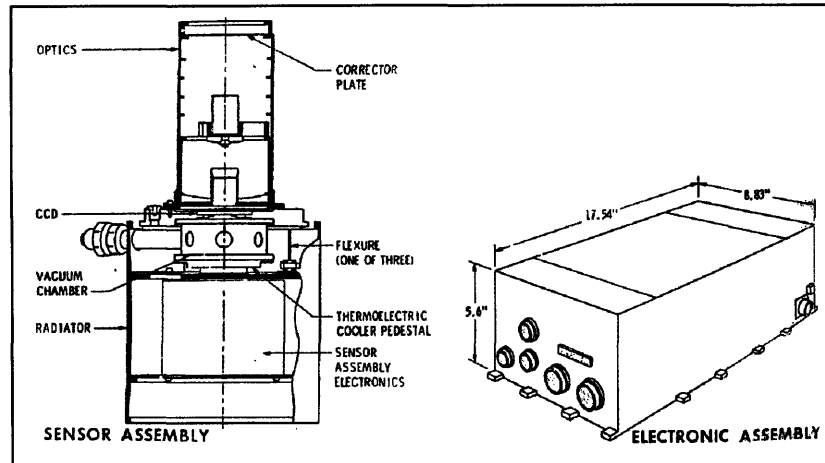


Figure 2.6 ASTROS Tracker Hardware Assembly<sup>[30]</sup>

Once the star tracker is assembled, it must be mounted onto the star tracker. The mount for the LIST star tracker is outlined in Figure 2.7.<sup>[47]</sup> The overall dimensions of the mount are 10.2cm x 8.9cm x 5.3cm.

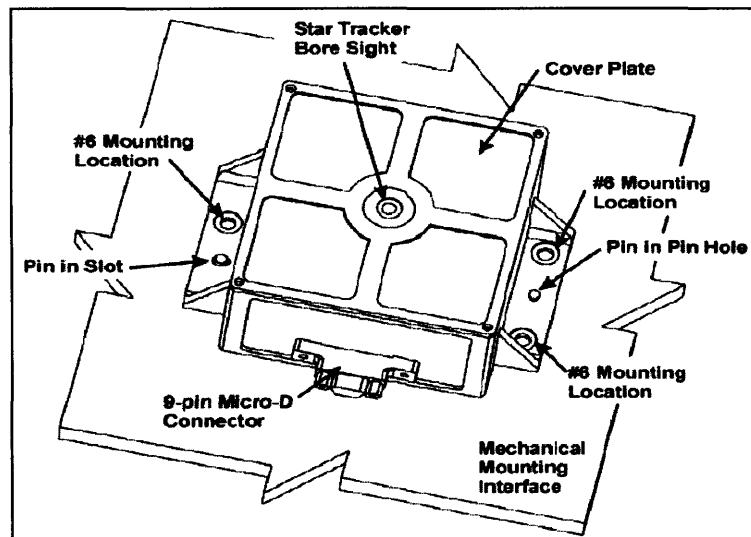


Figure 2.7 LIST Mount<sup>[47]</sup>

# Chapter 3

## STELLAR SIGNAL AND NOISE

In the subsequent algorithms described in Chapters 4, 5, and 6, several assumptions have been made regarding stellar signal and noise variations. The signal itself is assumed to be modeled by a two dimensional Gaussian. The stellar shot, dark current, and readout noise variations are significant electron contributions that must be considered in the image processing. Properties of the stellar signal and noise electrons are defined and discussed in this section.

### 3.1 Stellar Signal

As light is gathered by the star tracker, the optics system defocuses the photons on the detector. Each star that falls on the detector maps out a two-dimensional Airy function, as displayed in Figure 3.1<sup>[48]</sup> and Figure 3.2.<sup>[49]</sup> The central portion of the Airy pattern, which is known as the Airy Disk, comprises approximately 86% of the total number of electrons that make up the star.

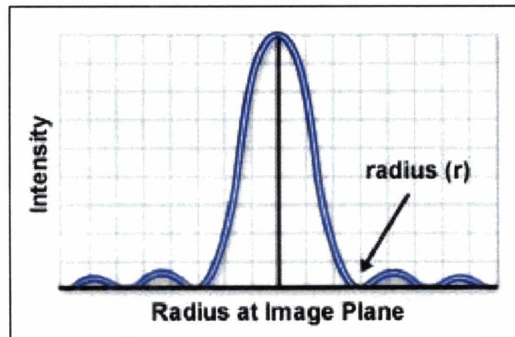


Figure 3.1 Intensity Profile of the Airy Function<sup>[48]</sup>



**Figure 3.2 Airy Disk with Outer Rings Digitally Enhanced to Make More Easily Visible<sup>[49]</sup>**

The Airy function,  $Ai(z)$ , is defined as<sup>[50]</sup>

$$Ai(z) \triangleq \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\left(\frac{zt+t^3}{3}\right)} dt \quad (3.1)$$

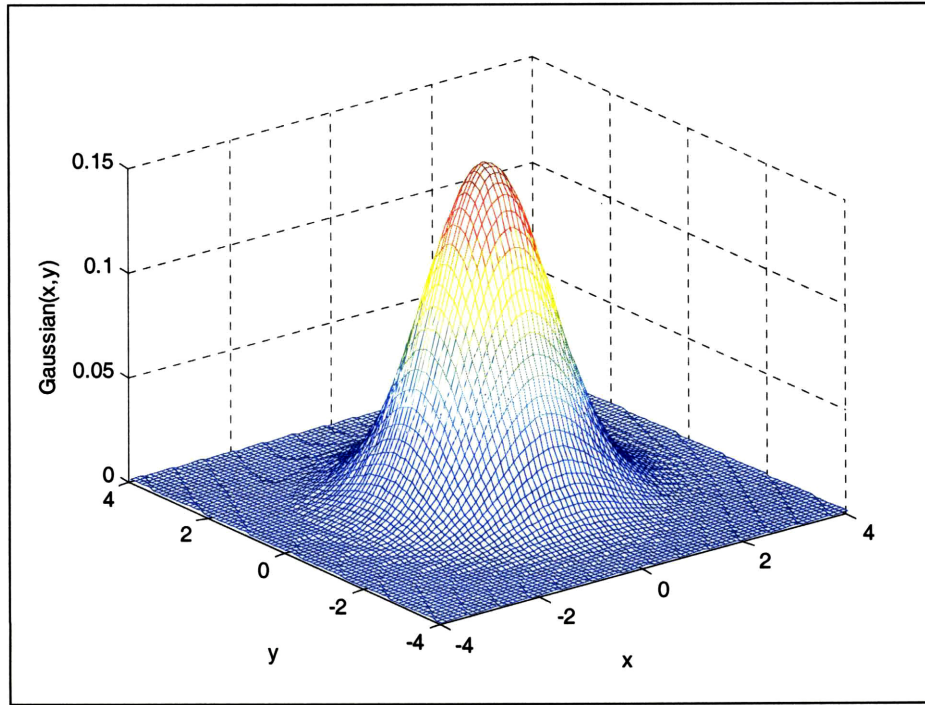
For real values, the Airy function is written as<sup>[51]</sup>

$$Ai(z) \triangleq \frac{1}{\pi} \int_0^{\infty} \cos\left(\frac{t}{3} + zt\right) dt \quad (3.2)$$

The two-dimensional Airy function is fairly accurately approximated out to the first zero by the two-dimensional Gaussian function. The Gaussian shape is assumed to approximate the number of photons that fall on the imager. The two-dimensional Gaussian is defined as<sup>[52]</sup>

$$f(x, y) \triangleq \frac{1}{2\pi\sigma_x\sigma_y} e^{-\left[\frac{(x-\mu_x)^2}{2\sigma_x^2} + \frac{(y-\mu_y)^2}{2\sigma_y^2}\right]} \quad (3.3)$$

where  $x$  and  $y$  are pixel coordinates,  $\mu_i$  is the mean centroid location, and  $\sigma_i$  is the standard deviation from the mean. A plot of the two-dimensional Gaussian function, having bivariate normal distribution with a mean of 0 and a variance of 1, is shown below in Figure 3.3.



**Figure 3.3 Gaussian Function with Bivariate Normal Distribution**

On star trackers, the stellar signal is the entity that is to be measured. In the case of CMOS and CCD imagers, this signal is generally measured in terms of the number of electrons that are generated on the imager. The number of electrons,  $eNum$ , for a given star is given as

$$eNum = (a)(FFSR)(t_{shutter})(g)\left(10^{\frac{(28-M_V)}{2.5}}\right) \quad (3.4)$$

where  $a$  is the lens area,  $FFSR$  is the product of the fill factor and the spectral response of the imager,  $t_{shutter}$  is the length of time in which the shutter is open to collect stellar light,  $g$  is the gain of the imager measured by the number of electrons represented by each analog-to-digital unit (ADU), and  $M_V$  is the visual magnitude. The fill factor is the percentage of an individual pixel that is photo-responsive, and the spectral response is the average value of the responsiveness of the imaging element to the characteristic spectrum in units of  $A/W$ .

## 3.2 Noise Variations

Ideally, the only photons passed through the imager are those from the stellar signal. However, all imagers contain uncertainties caused by the injection of various types of noise that are digitized along with the actual image. The most common of these are signal shot, dark current, and background noises. Additionally, there are pseudo-noise sources introduced during the process of digitizing the stellar signal, which include readout and quantization noises.

To fully comprehend digital signal and noise processing, several fundamental aspects of probability theory and statistics are required. Recall the generic formulas for one dimensional (1D) Gaussian and Poisson distributions. A 1D Gaussian distribution is modeled by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.5)$$

where  $x$  is a random variable,  $\mu$  is the mean,  $\sigma$  is the standard deviation, and  $\sigma^2$  is the variance. A 1D Poisson Distribution is modeled by

$$f(x) = \frac{\lambda^x e^{-\lambda}}{x!} \quad (3.6)$$

where  $x$  is a random variable and  $\lambda$  is both the mean and variance.

If  $X_1$  has Poisson distribution with parameter  $\lambda_1$ , then  $X_1 + X_2 + \dots + X_r$  has Poisson distribution with parameter  $\lambda_1 + \lambda_2 + \dots + \lambda_r$ . This means that the means and variances are additive under a summation of Poisson variables. This same property holds for a summation of Gaussian variables, but in general does not hold for the summation of a mixture of the two. However, as the Poisson parameter increases, the Poisson distribution can be approximated more and more accurately by a Gaussian, and in the limit that the parameter is large, the general rule of mean and variance sums does hold between the two distributions. This concept is important in that it allows different types of noises to be added together.



### 3.2.1 Signal Shot Noise

There is an inherent signal noise, known as stellar shot noise, which results solely from the stellar signal falling on the imager; the noise itself is due to the uncertainty in signal measurement. Since the electron arrival is a Poisson process, the variance is equal to the mean value, and therefore the standard deviation of the signal noise is

$$\sigma_{shot} = \sqrt{eNum} \quad (3.7)$$

### 3.2.2 Dark Current Noise

Dark current noise is a result of thermally generated electrons, which are present even in the absence of light. The majority of these electrons are created in the “forbidden” energy gap between the valence and conduction band, where photons excite electrons between interface energy levels.<sup>[53]</sup> The amount of dark current noise can be reduced by operating the imager in inversion, where holes from the channel prevent electron flow into the interface states. In space, dark current noise can be further reduced by directly connecting the imager to a passive radiator or active cooling device. Dark current noise is a Poisson process with

$$\sigma_{dark} = \sqrt{D} = \sqrt{d \cdot t_{collect}} \quad (3.8)$$

where  $D$  is the number of dark current electrons,  $d$  is the dark current rate, and  $t_{collect}$  is the time over which the dark current electrons are collected.

### 3.2.3 Background Noise

Background noise accounts for all other noise additives that contribute to an uncertainty in signal strength, which often includes noise from the satellite or the surrounding environment. There are numerous types of background noise, but each is generally very specific to application and, therefore, often difficult to model. Background noise originating from the surrounding environment is usually quite small when compared to the lower magnitude stars generally used for navigation; it would certainly be dominated by other potential sources, such as glint from the sun as it reflects off of other surfaces of the spacecraft. The background noise will be assumed

to follow the same Poisson statistics as the signal shot noise, given that its source is external to the imager and defined by the same process of photon arrival rate. It is considered to accumulate as specified by a rate, in the same way that the dark current is specified, but rather than being modeled, it will instead have its value bounded by the performance requirements of the imager.

### 3.2.4 Quantization Noise

To understand quantization noise, an understanding of imager gain is first necessary. The imager itself generates ‘counts’ from the individual image pixels, where the counts are proportional to the total number of electrons acquired. To model the uncertainty in actual numbers of electrons per pixel, consider the imager’s gain,  $\Delta$ , which quantifies the proportionality relating the actual electron count to the output counts measured in ADUs. Although the gain can often be variable, the maximum gain is generally defined as

$$\Delta = \frac{\text{fullWell}}{ADU_{\max}} \quad (3.9)$$

where *fullWell* is the maximum number of electrons that a single pixel can hold, and  $ADU_{\max}$  is two raised to the number of bits available to the analog-to-digital converter (ADC). Increasing the gain beyond the value given in (3.9) simply results in a loss of resolution, since the maximum ADU extends beyond the capacity of the fullWell. As an example, the IBIS5-A-1300 has a fullWell of approximately 120,000 electrons and a maximum ADU value of  $2^{10} = 1024$  counts.<sup>[40]</sup> Therefore, the maximum useful gain setting for this imager is

$$\Delta = \frac{120,000}{1024} \cong 117 \quad (3.10)$$

This means the stellar intensity can be resolved to approximately 117 electrons per ADU. Consequently, for example, anywhere from 0 to 58 electrons will be read in as 0 electrons, and anywhere from 59 to 176 electrons will be read in as 117 electrons. To increase the intensity resolution, the user can set the imager to a lower gain or select an imager with a higher maximum ADU value (which contains more ADC bits).

Quantization noise is a result of uncertainty in the true value of the electron count, as described above, caused in the ADC. Quantization noise can be modeled as a flat distribution, meaning that the data is distributed so that all values between two limits are equally likely to occur. With no intrinsic bias, the distribution also has zero mean. Thus, the variance of quantized noise is found by determining the expected value of the mean-squared error,

$$E\left(\frac{(x-\mu)^2}{N}\right) = \frac{1}{N} E(x^2), \quad (3.11)$$

or more explicitly<sup>[54]</sup>

$$\sigma_{\text{quantization}}^2 = \frac{1}{N} E(x^2) = \frac{1}{\Delta} \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} x^2 dx = \frac{1}{\Delta} \left. \frac{x^3}{3} \right|_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} = \frac{1}{\Delta} \left[ \frac{\left(\frac{\Delta}{2}\right)^3}{3} - \frac{\left(-\frac{\Delta}{2}\right)^3}{3} \right] = \frac{1}{\Delta} \left[ \frac{\Delta^3}{24} - \frac{-\Delta^3}{24} \right] = \frac{\Delta^2}{12}. \quad (3.12)$$

Therefore, the standard deviation of the quantization noise is

$$\sigma_{\text{quantization}} = \frac{\Delta}{\sqrt{12}}. \quad (3.13)$$

### 3.2.5 Readout Noise

Readout noise is associated with the inexact conversion of electrons from the signal falling on the imager to the readout of the imager. It is mainly caused by the on-chip output amplifier, where electrons are converted into analog voltage. The standard deviation of readout noise is given by

$$\sigma_{\text{readout}} = \frac{R}{\Delta} \quad (3.14)$$

where  $R$  is the number of electrons caused by the readout noise and  $\Delta$  is the gain in electrons/ADU, as used above for quantization noise. In contrast to the quantization error, which increases with higher gain, the readout error, being a fixed number of electrons per readout, will

decrease in comparison with the signal as the gain is increased. Both readout and quantization noise are limited by the construction of the imager itself and therefore cannot be improved upon by the user or image processing techniques.

### 3.2.6 Combining Different Noise Types

The total amount of noise falling on the imager has been shown to consist of several sources, each with a distinct statistical dependence. By statistical definition, the variance is the mean square of the difference in actual value and average value. When a random variable can be expressed as the sum of  $M$  random variables, the following relationship applies:

$$\begin{aligned}
 \sigma_{sample}^2 &= \frac{1}{N} \sum_i^N (x^i - \mu)^2 = \frac{1}{N} \sum_i^N \left( \sum_j^M (x_j^i - \mu_j) \right)^2 \\
 &= \frac{1}{N} \sum_i^N \left[ (x_1^i - \mu_1)^2 + (x_2^i - \mu_2)^2 + \dots + (x_1^i - \mu_1)(x_2^i - \mu_2) + \dots \right] \\
 &= \frac{1}{N} \sum_i^N (x_1^i - \mu_1)^2 + \frac{1}{N} \sum_i^N (x_2^i - \mu_2)^2 + \dots + \frac{1}{N} \sum_i^N (x_1^i - \mu_1)(x_2^i - \mu_2) + \dots \\
 &= \sum_j^M \sigma_j^2 + \frac{1}{N} \sum_i^N (x_1^i - \mu_1)(x_2^i - \mu_2) + \dots
 \end{aligned} \tag{3.15}$$

As the sample size becomes large, the sample variance of each variable approaches the statistical variance, and the cross-correlations approach zero, since each variable is distributed evenly about its mean value, and each is assumed to be totally uncorrelated with any other. For the specific cases of Poisson and Gaussian distributions, the assumption of large sample sizes is not required for this to hold, as was discussed earlier. Thus, for large sample sizes, the variance for the noise electrons is

$$\sigma_N^2 = \sigma_S^2 + \sigma_D^2 + \sigma_R^2 + \sigma_Q^2 + \sigma_B^2 \tag{3.16}$$

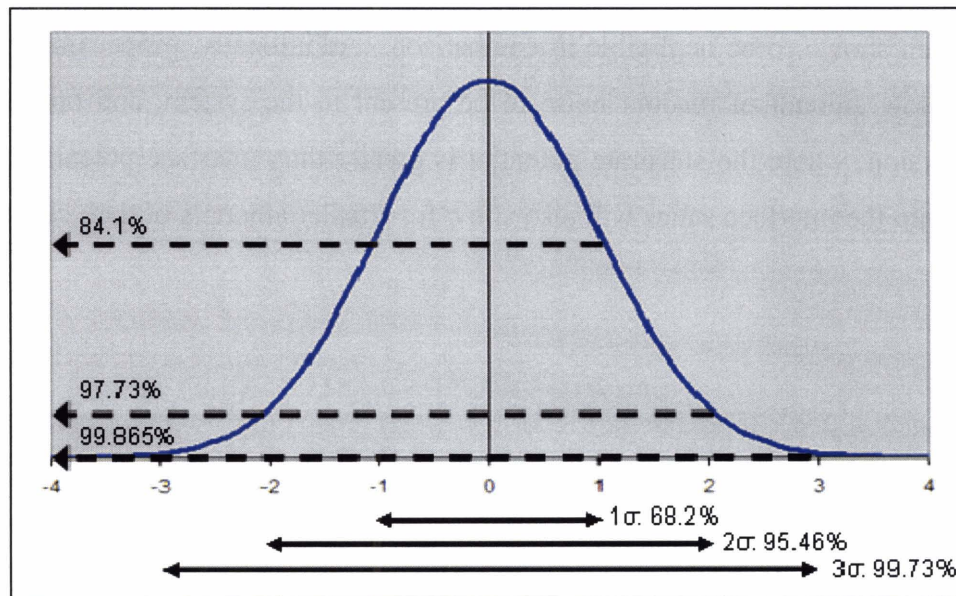
where  $S$ ,  $D$ ,  $R$ ,  $Q$ , and  $B$  represent the signal, dark current, readout, quantization, and background noises, respectively.

### 3.3 Signal-to-Noise Ratio

The signal-to-noise ratio (SNR) is the measure of signal quality at a given pixel, which is expressed as the ratio of measured signal to incorporated noise. It is defined as the ratio of the signal electron count to one standard deviation of the total noise electron count, which, given in terms of the derived variances discussed above, results in

$$SNR = \frac{S}{\sqrt{S + D + R^2 + \frac{A^2}{12} + B}} \quad (3.17)$$

Equation (3.17) can be used to determine the percentage of times that the signal will exceed the noise by this ratio, as displayed in Figure 3.4. By the central limit theorem, the distribution of the SNR is normal with mean  $S$  and standard deviation as given by the square-root of (3.16).<sup>[55]</sup> The solid arrows below the curve represent the percentage of noise within the  $\pm 1\sigma$ ,  $\pm 2\sigma$ , and  $\pm 3\sigma$  bands. The dashed arrows represent the amount of time in which the signal will exceed the noise by the ratio given in (3.17) for  $1\sigma$ ,  $2\sigma$ , and  $3\sigma$  amounts of noise.



**Figure 3.4 Standard Normal Distribution Depicting the Percentage that SNR Will Exceed  $1\sigma$ ,  $2\sigma$ , and  $3\sigma$  Amounts of Noise**

In terms of stellar simulations, the SNR can be calculated by

$$SNR = \frac{eNum}{2\pi\sigma_{star}^2\sigma_{noise}} \quad (3.18)$$

where  $eNum$  is the number of electrons in a given star,  $\sigma_{star}$  is the width of the point spread function of the star which approximates a Gaussian, and  $\sigma_{noise}$  is the standard deviation of noise on the imager.

SNR is an important parameter in image processing. It will be shown in Chapter 4 that the values of SNR directly affect the centroider's ability to locate precise centers of the stars, which automatically dictates the accuracy of the attitude quaternion discussed in Chapter 5. It will be verified that higher values of SNR produce more accurate centroid locations.

Due to the nature of the stars trackers, it will at times be necessary to process stars of relatively dim visual magnitude in order to meet the micro-satellite attitude acquisition requirements. Therefore, it is important to understand the need to minimize the amount of noise in the system to maintain the largest SNR possible. The image processing techniques in this thesis account for signal shot, dark current, and readout noises, as quantization and background noises have been shown to be negligible in comparison. Additionally, proper imager selection has enabled a low amount of readout noise to be present in the system, and operation of the imager in inversion, where the substrate potential is greater than interface potential, to prevent electron flow into the interface states will also allow for smaller amounts of dark current noise to accumulate.

# Chapter 4

## THRESHOLDING AND CENTROIDDING

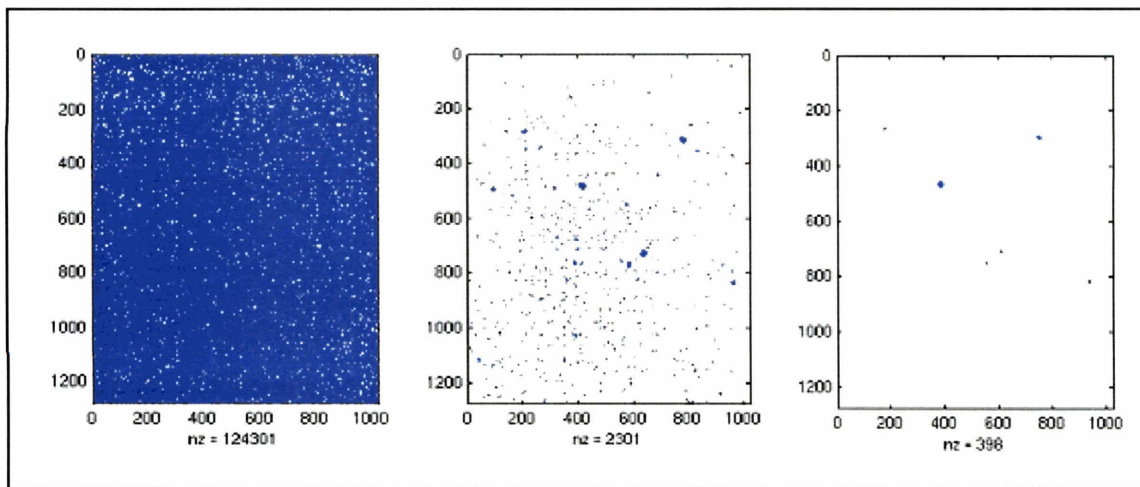
### 4.1 Image Thresholding

Given the total number of photons that fall onto the imager surface in a particular image, the thresholding function allows stellar photons to be accurately distinguished from the noise electrons. In general, readout noise is constant across the imager plane, and thus shot noise and dark current are the main elements that need to be distinguished from the stellar signal for accurate centroiding. This is accomplished by setting a noise threshold, which allows only those pixels that contain electron counts higher than this minimum value to be considered as potential star pixels. There are advantages and disadvantages of setting particular threshold values. Setting a high noise threshold drastically reduces the chances of “false stars,” noise spikes caused by abnormally high dark current counts, but prevents the imager from finding dim stars and portions of stars whose signal strengths fall below this threshold. On the other hand, setting a low threshold enables the processing of dimmer stars but prevents discernment between noisy pixels and actual stars.

Star trackers discussed in the literature often calculate a background intensity level to determine a minimum threshold level.<sup>[30, 56]</sup> For example, the micro star tracker DIGISTAR uses a look-up table, that was created based on empirical results from nighttime sky images, to set a minimum threshold level. The average background intensity level of an incoming image is examined to estimate an overall background level by selecting and examining five spatially distributed 10pixel x 10pixel windows devoid of stars. Once this level is determined, the look-up table automatically sets the minimum threshold level to ensure that the threshold is not at or below the background level, which could result in every pixel becoming a potential star.

It is possible for the SNR to be close to unity, especially during fast rotation rates that only allow short integration times. Thus, each pixel on the imager is filtered twice: first based

upon the pixel's relation to the threshold value and second based upon its location relative to other pixels of similar intensity. Figure 4.1(left) depicts a stellar image, with dark image subtracted, whose threshold was set at an SNR of unity. The dark image was defined as the median of the content of each pixel in the pixel matrix across several images in a sequence. In Figure 4.1(left) approximately 124,000 pixels (10% of the pixels in the image) exceed this threshold. Based on the SNR description in Chapter 3, one would expect this percentage to be closer to 16%, but it is speculated that the 6% difference is due to the fact that the dark image, with which only the median value in each pixel was considered, was subtracted out. In the vicinity of a star, roughly half of the pixels contained in the star should exceed this threshold. This knowledge leads to a method by which stellar and noise pixels can be statistically distinguished. Two pre-filters have been implemented. The first examines a 5-pixel by 5-pixel region centered on each pixel that exceeds the stated threshold, and determines if at least 9 of the 25 pixels in this region also exceed the threshold. If at least 9 pixels are found, the area is regarded as being a possible star, as the likelihood of this occurring due to noise is small.



**Figure 4.1 Filtered Image with Threshold Set at SNR = 1**

However, at an SNR of unity, there are still many pixels that will statistically form groups of 5 or more, as seen in the center image of Figure 4.1. The second filter examines the proximity of each pixel to those in its local neighborhood and eliminates the proposed stars from the first filtering that do not contain pixels with sufficient electrons in closely packed groups. A



final filtering discards any groups that do not include at least 3 pixels as statistically rare noise events.

## 4.2 Image Centroiding

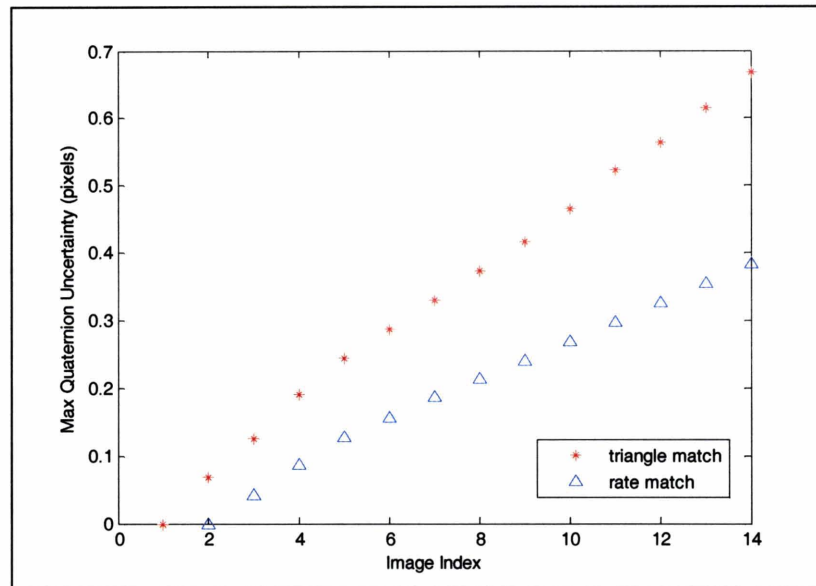
Image centroiding is a vital process to increase the accuracy of the attitude data set.<sup>[57]</sup> Without accurate centroiders, precise attitude information through image processing cannot be achieved. Precise centroiding is difficult in the presence of noise because pixels contain inaccurate numbers of electrons, which can cause the calculated centroid positions to be different from actual locations. However, even in the presence of no noise, there is a limit to the centroiding precision that can be achieved due to issues of discretization.

Centroiding methods favor a defocusing of the stars as they fall on the imager to improve the precision of centroid predictions. The amount of defocusing, along with the manner that best weighs the pixels to locate the center of the star, is the primary focus of centroiding algorithms. The performance ability of any centroider is directly related to the strength of the stellar signal, the presence of noise variations, and the image spot size as it falls on the imager. Originally two centroiding methods were created to be used consecutively; a weighted sum (WS) technique would approximate the centroid positions to single pixel resolution and a maximum likelihood estimator (MLE) technique further refined centroids to subpixel resolution. However, based on the analysis and performance of the MLE as described below, the WS centroider is the only centroider used for image processing in LIST and FAR-MST star trackers. Two versions of the WS centroider have been created; the centroider of choice is based upon the current noise level.

### 4.2.1 Formation of Star Clusters

Before images can be successfully centroided, pixels must be appropriately grouped into stellar clusters. Once the noise threshold level is set, pixels that exceed this threshold can be considered potential star pixels. Both the ASTROS and DIGISTAR star trackers form pixel clusters by selecting the brightest pixels on the image as “core pixels” of stars. All pixels exceeding the threshold that have at least one adjacent pixel with the core pixel are then assigned

The triangle match and rate match produce accumulating uncertainty errors over consecutive images, and the uncertainties were the largest in the roll direction. Figure 8.1 compares the uncertainties of each image's attitude quaternion for the triangle match and rate match. Comparisons of the pitch and yaw uncertainties produce similar results.



**Figure 8.1 Comparison of Triangle Match and Rate Match Quaternion Uncertainties in Roll Direction**

The predictive ability of the rate match to find all of the matchable stars proved to be advantageous because the uncertainty errors of the rate match accumulated at a slower rate than those of the triangle match, as seen by the rate match's smaller slope.

Apart from accuracy, computational requirements for the pattern matching algorithms were considered. Table 8.2 compares the time requirements of each algorithm to threshold, centroid, and pattern or rate match the stars and output an attitude quaternion. These times were collected using Matlab versions of the algorithms; the actual algorithms will be run in C, which will produce much faster times than those listed below. The Matlab times have been included to show the relative time requirements of the different matching algorithms.

**Table 8.2 Time Comparisons for Matching Algorithms**

algorithm:	triangle match	rate match	3-vertex match	4-vertex match	5-vertex match
time (s):	9.06	15.12	8.30	8.27	8.33

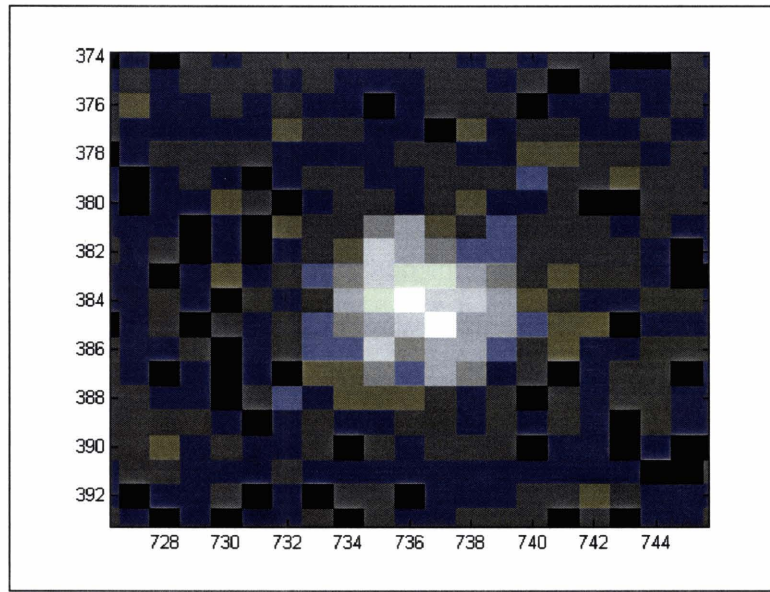


Figure 4.3 Dim Star

## 4.2.2 LIST and FAR-MST Centroiding Techniques

### *Weighted Sum Technique*

Many star trackers implement a mass moment or center of gravity technique to centroid stars. This is similar to a weighted sum technique, in which the brightest pixel in the immediate neighborhood is localized and centroiding is performed via computation of the “center of gravity” in the specific region of pixels surrounding the brightest pixel.<sup>[58]</sup>

Two weighted sum centroiding algorithms have been implemented to determine the stars’ centers in a given image. For each star on a particular image, each of these algorithms determines the centroid x-y position on the imager, overall brightness in number of electrons, and aspect ratio for each star in the image. The aspect ratio is the ratio of second moments and is used to determine whether or not the star is creating a streak on the image, which is an indication as to whether or not the appropriate integration time is being used. The difference between these three algorithms is the type of filtering used to eliminate likely noise spikes and whether or not current rate information is available to predict centroid locations. The value of the SNR and the availability of current attitude information determine which centroider should be implemented.

The first centroider, `starCentroid2`, uses a pre-filter that scans the image for statistically relevant, elevated count levels to aid in the reduction of the number of random noise spikes. These statistics are evaluated over 5pixel x 5pixel squares that are centered over each pixel found to exceed the noise threshold. If more than half of the 25 pixels in the square exceed this threshold, the pixel under examination is considered to be part of a star; otherwise, the pixel is dismissed as noise. A second filter eliminates stars that contain less than three pixels, which is most likely a noise spike. The second centroider, `starCentroid3`, uses a pre-filter that is applied to regularly spaced 5pixel x 5pixel regions on the pixel matrix to accelerate the execution time. This filter enables faster execution times than `starCentroid2` if more than 1/25 of the total pixels exceed the threshold. Therefore, this centroider is used for images with low SNR values, when a low noise threshold must be selected to extract as many stars as possible from noisy images. Performance trade studies, completed in Chapter 7, have explained the conditions under which each centroider performs best.

Each centroiding algorithm first defines all of the pixels that belong to each individual star that exceeds the desired noise threshold. It then uses the number of stellar photons in pixels as the weighting function, which allows for quick centroiding:

$$position = \frac{\sum_{i=1}^n (SI_i * position_i)}{\sum_{i=1}^n (SI_i)} \quad (4.1)$$

where  $i$  is the current pixel,  $n$  is the total number of pixels in the current star,  $SI$  is the number of stellar photons in the current pixel, and  $position$  represents the coordinates of the current pixel on the imager. The weighted sum centroiding algorithm is disadvantageous in that it is unable to take noise variations into account; the noise is assumed constant across each pixel on the imager. Post filtering is applied to the centroider to deal with noise spikes and hot pixels. Any stars that are spread over fewer than four pixels are discarded as false stars.

### Measuring Centroid Uncertainty

One measure of performance for the centroiding algorithm is the accuracy of the centroiding for each individual star. The primary driver in centroiding accuracy is aperture size and integration time; larger lens diameters and longer integration times allow more photons to fall on the imager, thus increasing the SNR in each pixel. However, longer integration times require much slower rotations to avoid stellar streaking, which decreases the ability of the star tracker to perform at higher tumble rates. The uncertainty in each centroid position was estimated using an empirical curve fit derived from a Monte Carlo simulation and is given by

$$psfWidth = 0.11 + 0.5SNR^{-1.3} \quad (4.2)$$

Accurate centroiding is vital to successful image processing; the tighter the centroider, the higher the performance of the pattern matching algorithms discussed in section 4.0. When noise variations are ignored, the centroiding accuracy improves with decreasing PSF width, as depicted in Figure 4.4.

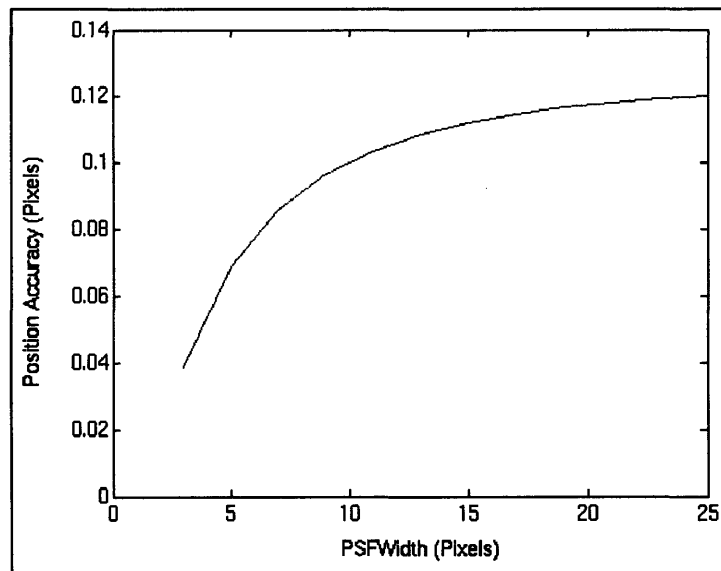


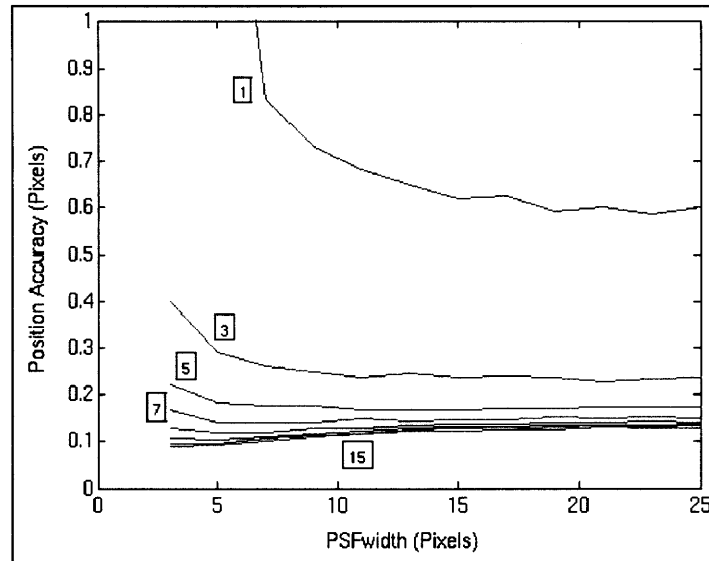
Figure 4.4 Centroid Accuracy with No Noise

Upon examination of Figure 4.4, it can be inferred that the centroid accuracy has an asymptote at approximately 0.12 pixels, which is due to two opposing effects. The first effect results because the quantization of the stellar intensity always assumes that the content of the

pixel is located in the center of the pixel. This notion can cause considerable biases in the vicinity of steep gradients, and consequently decreases in the accuracy of the calculated centroid. If the star is perfectly centered on a pixel, then these biases cancel one another. However, if the star is not perfectly centered, each pixel contained in the star adds a portion of bias to the centroid position. In general, stars spread across more pixels have greater uncertainties in centroided positions, although this increase in uncertainty becomes less significant for PSF widths greater than approximately 15 pixels.

The second effect is that when considering a star spread over  $N$  pixels, the uncertainty in centroid position decreases by  $1/\sqrt{N}$ . For all star sizes, each pixel introduces a particular amount of uncertainty in centroid position. A large fraction of the pixels contained in small stars are located on the periphery of the star, where the SNR is lowest, which logically produces the largest amounts of centroiding uncertainty. However, as the PSF width of the star increases, the fraction of pixels located along the edge of the star decreases. Therefore, the off-center bias that causes a decline in centroid accuracy by  $\sqrt{N}$  for larger PSF widths is balanced by the fact that there is a  $1/\sqrt{N}$  improvement in centroid accuracy for larger PSF widths due to smaller ratios of edge pixels to central pixels.

However, when noise is present in the system each pixel has an additional uncertainty associated with it. Therefore, as depicted in Figure 4.5, SNR has a sizable impact on positioning accuracy. The boxed numbers in the figure represent the various SNR values.



**Figure 4.5 Centroid Accuracy for Various SNR**

On any star, the edge pixels contain the lowest SNR values; therefore, the fewer the pixels in a given star, the worse the knowledge of the true centroid becomes, except in the limit where the SNR of the star is very high. Given the data presented in Figure 4.5, the ideal PSF width of a star having a given SNR is the PSF width located at the knee of the curve. For example, at an SNR of 1, the ideal PSF width is approximately 9pixels, and at an SNR of 3, the ideal PSF width is approximately 5pixels.

### ***Maximum Likelihood Estimator Technique***

The idea behind the maximum likelihood estimator (MLE) comes from the derivation of a two-dimensional Cramér-Rao lower bound, which helps to develop a lower limit for the mean-squared error of an unbiased position estimator.<sup>[59]</sup> Throughout the analysis performed by Winick, identical square pixels, an image spot intensity profile represented by an Airy function and approximated by a Gaussian curve, and stellar shot and dark current noise contributions defined by Poisson processes were assumed.

Image spot size can be adjusted by varying the focal length of the optical subsystem, which is equivalent to varying the PSF width in simulated images. Winick argued that a particular pixel-to-image spot-size ratio exists that minimizes the position-estimation error. Image spot sizes that are too large mean that the image is spread over numerous pixels, which

forces a lower electron count in the each pixel, making it difficult to distinguish pixels containing pure noise. On the other hand, image spot sizes that are too small may contain the majority of their electrons within a single pixel making accurate sub-pixel resolution impossible. The goal of Winick's work was to determine the best pixel-to-image spot-size ratio.

According to Winick, the maximum likelihood estimator, or the most likely two-dimensional centroid, can be derived based upon the derivation of the two-dimensional standard Cramér-Rao bound

$$\mathbf{E}\left[(\hat{\epsilon}_x(\mathbf{c}) - \epsilon_x)^2\right] \geq \left( \frac{\mathbf{E}\left\{\left[\frac{\partial \ln(P(\mathbf{c}|\epsilon_x, \epsilon_y))}{\partial \epsilon_y}\right]^2\right\}}{\mathbf{E}\left\{\left[\frac{\partial \ln(P(\mathbf{c}|\epsilon_x, \epsilon_y))}{\partial \epsilon_x}\right]^2\right\} \mathbf{E}\left\{\left[\frac{\partial \ln(P(\mathbf{c}|\epsilon_x, \epsilon_y))}{\partial \epsilon_y}\right]^2\right\} - \left\{\mathbf{E}\left[\frac{\partial \ln(P(\mathbf{c}|\epsilon_x, \epsilon_y))}{\partial \epsilon_x} \frac{\partial \ln(P(\mathbf{c}|\epsilon_x, \epsilon_y))}{\partial \epsilon_y}\right]\right\}^2}\right) \quad (4.3)$$

knowing

$$\frac{\partial}{\partial \epsilon_y} \int [\hat{\epsilon}_x(\mathbf{c}) - \epsilon_x] P(\mathbf{c}|\epsilon_x, \epsilon_y) d\mathbf{c} = 0 \quad (4.4)$$

where  $\epsilon_x$  and  $\epsilon_y$  are the x and y centroid position estimators, respectively,  $\mathbf{c}$  is an observed vector quantity statistically related to  $\epsilon_x$ ,  $\hat{\epsilon}_x(\mathbf{c})$  is the unbiased estimator of  $\epsilon_x$ , and  $P(\cdot)$  is the probability of the enclosed argument.

Using this information, Winick found that the MLE is the location where  $Q_x$  and  $Q_y$  simultaneously equal zero.  $Q_x$  and  $Q_y$  are defined as

$$Q_x = \sum_{ij} c_{ij} \frac{\lambda_s g'_i(\epsilon_x) g_j(\epsilon_y)}{\lambda_s g_i(\epsilon_x) g_j(\epsilon_y) + \lambda_N} \quad (4.5)$$

$$Q_y = \sum_{ij} c_{ij} \frac{\lambda_s g_i(\epsilon_x) g'_j(\epsilon_y)}{\lambda_s g_i(\epsilon_x) g_j(\epsilon_y) + \lambda_N} \quad (4.6)$$



where

$$g_i(\varepsilon_x) \triangleq \int_{x_i - \frac{\Delta x}{2}}^{x_i + \frac{\Delta x}{2}} S(x, \varepsilon_x) dx \quad (4.7)$$

$$g_j(\varepsilon_y) \triangleq \int_{y_j - \frac{\Delta x}{2}}^{y_j + \frac{\Delta x}{2}} S(y, \varepsilon_y) dy \quad (4.8)$$

with

$$g_{ij}(\varepsilon_x, \varepsilon_y) = \lambda_s g_i(\varepsilon_x) g_j(\varepsilon_y) \quad (4.9)$$

being the average number of electrons produced by a stellar image spot. In (4.5) and (4.6)  $c_{ij}$  is the number of electrons per pixel,  $\lambda_s$  is the total number of electrons over the entire star, respectively,  $\lambda_N$  is the standard deviation of the dark current and readout noise,  $x_i$  and  $y_j$  are the center of the x-y pixel,  $\Delta x$  is the width of a pixel, and  $\varepsilon_x$  and  $\varepsilon_y$  are the x and y centroid position estimators, respectively. Additionally,  $S(x, y, \varepsilon_x, \varepsilon_y)$  is the Gaussian shaped intensity profile defined by

$$S(x, y, \varepsilon_x, \varepsilon_y) = (2\pi\sigma_s^2)^{-1} \exp\left(-\frac{(x-\varepsilon_x)^2}{2\sigma_s^2}\right) \exp\left(-\frac{(y-\varepsilon_y)^2}{2\sigma_s^2}\right) \quad (4.10)$$

which can be decomposed into

$$S(x, \varepsilon_x) = (2\pi\sigma_s^2)^{-1/2} \exp\left(-\frac{(x-\varepsilon_x)^2}{2\sigma_s^2}\right) \quad (4.11)$$

$$S(y, \varepsilon_y) = (2\pi\sigma_s^2)^{-1/2} \exp\left(-\frac{(y-\varepsilon_y)^2}{2\sigma_s^2}\right) \quad (4.12)$$

where  $\sigma_s$  is the 1-sigma width of the Gaussian shaped intensity curve. The best case scenario is approximated by

$$\sigma_s = \frac{psfWidth}{4}, \quad (4.13)$$

which is derived here. As first discussed in Chapter 3, let a symmetric two-dimensional Gaussian normalized to one with mean zero and variance  $\sigma^2$  be defined as

$$f(x, y) \triangleq \frac{1}{2\pi\sigma^2} e^{-[r^2/2\sigma^2]} \quad (4.14)$$

where  $r$  is the radius of the PSF that extends from the center of the star. The relationship in (4.13) is found by double integration of (4.14)

$$= \int_0^{2\pi} \int_0^r \frac{1}{2\pi\sigma^2} e^{-\tilde{r}^2/2\sigma^2} \tilde{r} d\tilde{r} d\theta \quad (4.15)$$

$$= 2\pi \int_0^r \frac{1}{2\pi\sigma^2} e^{-\tilde{r}^2/2\sigma^2} \tilde{r} d\tilde{r} \quad (4.16)$$

$$= \frac{1}{\sigma^2} \int_0^r e^{-\tilde{r}^2/2\sigma^2} \tilde{r} d\tilde{r} \quad (4.17)$$

$$= \frac{1}{\sigma^2} \left[ -e^{-\tilde{r}^2/2\sigma^2} \sigma^2 \right]_0^r \quad (4.18)$$

$$= 1 - e^{-r^2/2\sigma^2}. \quad (4.19)$$

Recall that the actual pattern of starlight maps out an Airy function on the image plane, where the Airy disk contains 86% of the total number of electrons of the star. By superimposing a Gaussian on the Airy disk, it is desirable that the Gaussian also comprises 86% of the total number of electrons

$$1 - e^{-r^2/2\sigma^2} = 0.86. \quad (4.20)$$

Solving (4.20) gives

$$\frac{r}{\sigma} = 2. \quad (4.21)$$

Since the PSF width is equal to twice the radius  $r$ ,

$$\frac{psfWidth}{\sigma} = 4 \quad (4.22)$$

which is the same as (4.13).

### ***Performance Comparison Between WS and MLE Centroiders***

A comparison between the weighted sum and maximum likelihood estimator centroiders was performed to determine the best centroider. In this trade study, thirty stars of the same visual magnitude were randomly placed on an image, noise was added, and then these simulated images were run through the WS centroider, using starCentroid3, and the MLE centroiders. For a given PSF width and noise threshold, a series of visual magnitudes were examined. Several additional trade studies were completed for other values of PSF width and noise thresholds. Table 4.1 shows a list of the parameters used in creating the simulated images for the first trade study.

**Table 4.1 Centroider Comparison Image Parameters**

<b>Parameter</b>	<b>Value</b>
# of stars on image	30
gain	10
FFSR (A/W)	0.12
$t_{\text{shutter}}$ (s)	0.1
DK (# of photoelectrons)	41
RO (# of photoelectrons)	40
$a$ (m <sup>2</sup> )	3.14E-04
psfWidth (pixels)	10
noise threshold (# of $\sigma$ )	5

Shown below are the results for the centroid comparison; they are best understood when examined as a group rather than individually. Figure 4.6, Figure 4.8, Figure 4.10, Figure 4.12, and Figure 4.14 show the noisy pixel matrix of the simulated image. As the values of  $M_V$

increase, the stars become appropriately dimmer. Figure 4.7, Figure 4.9, Figure 4.11, Figure 4.13, and Figure 4.15 show the differences in actual centroid locations and that which was calculated by each of the centroiders.

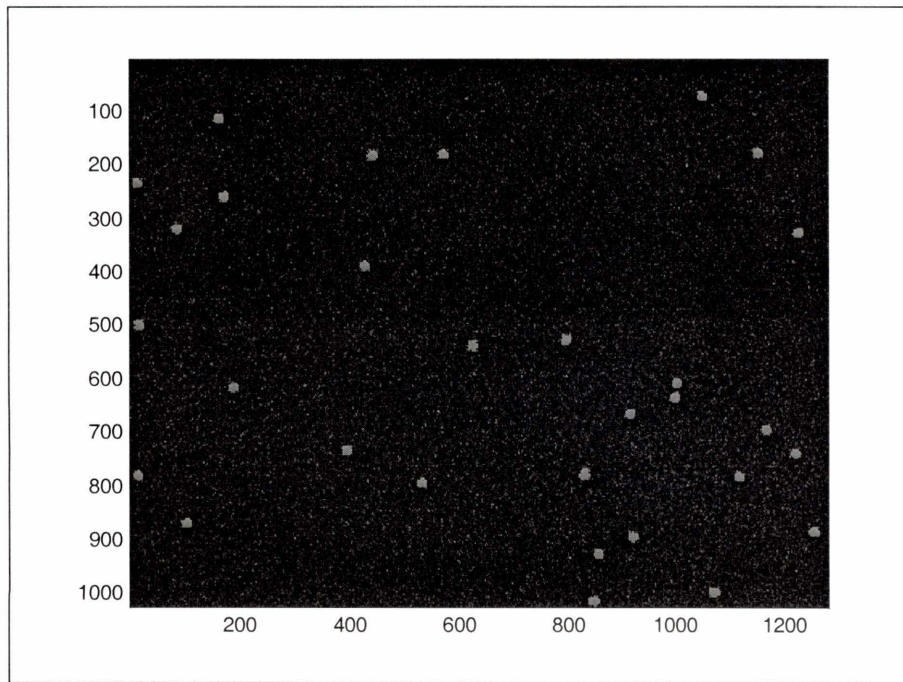


Figure 4.6 Simulated Noisy Image for  $M_V = 1$

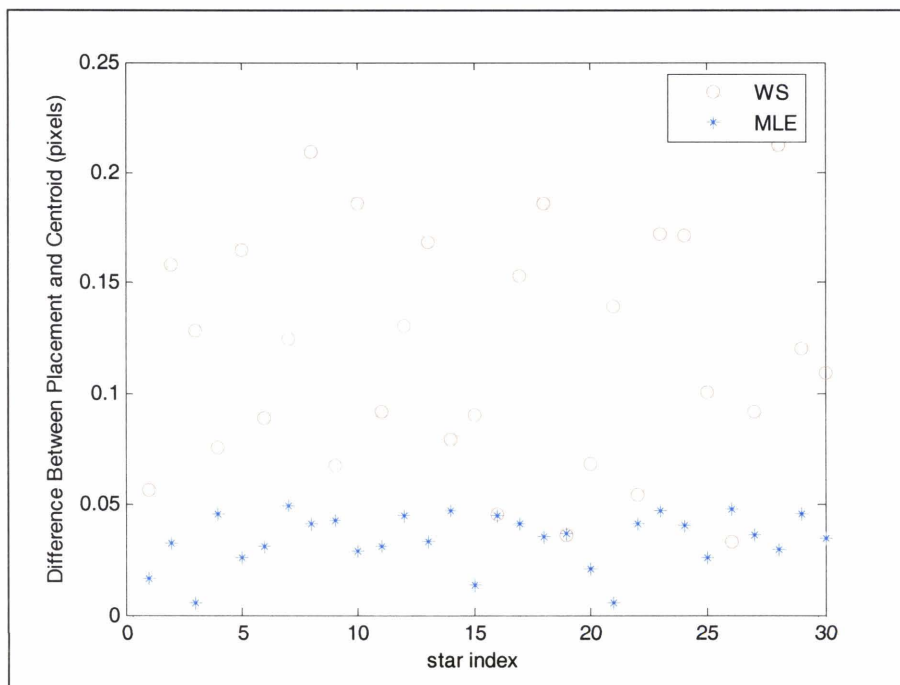


Figure 4.7 Comparison of WS and MLE Centroiding Errors for  $M_V = 1$

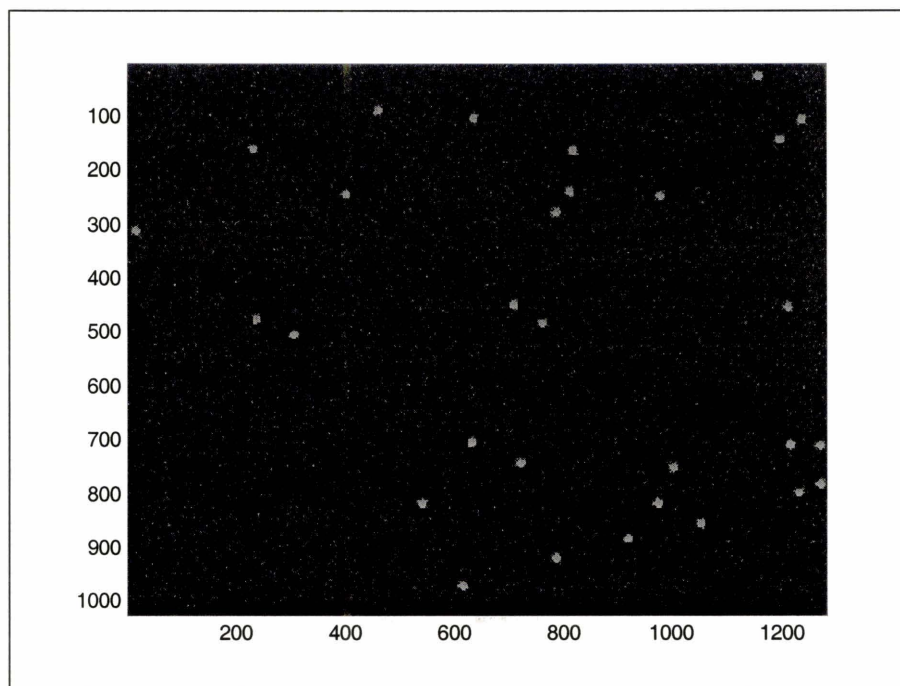


Figure 4.8 Simulated Noisy Image for  $M_V = 2$

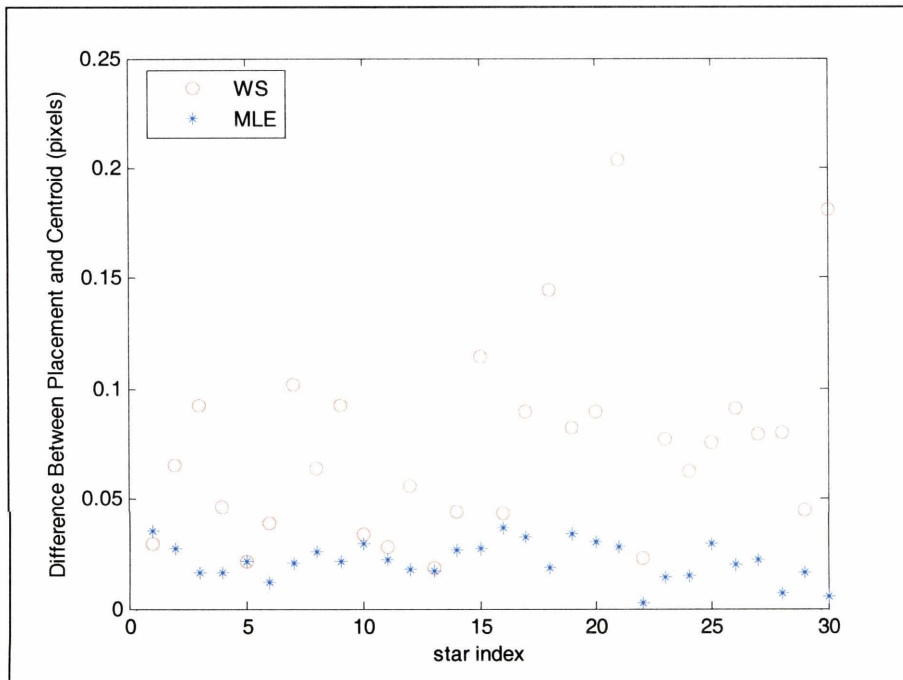


Figure 4.9 Comparison of WS and MLE Centroiding Errors for  $M_v = 2$

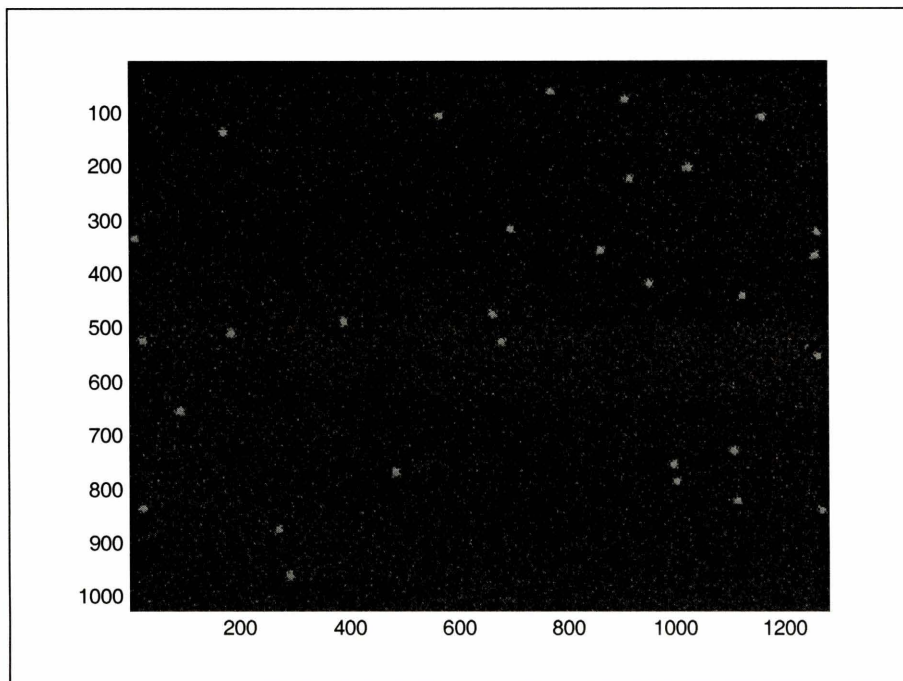


Figure 4.10 Simulated Noisy Image for  $M_v = 3$

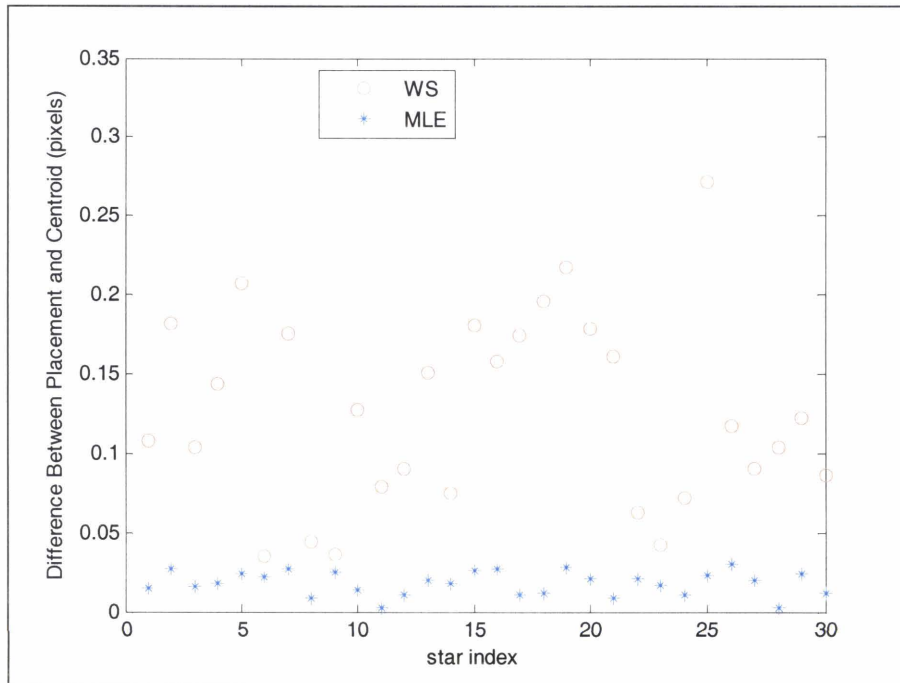


Figure 4.11 Comparison of WS and MLE Centroiding Errors for  $M_V = 3$

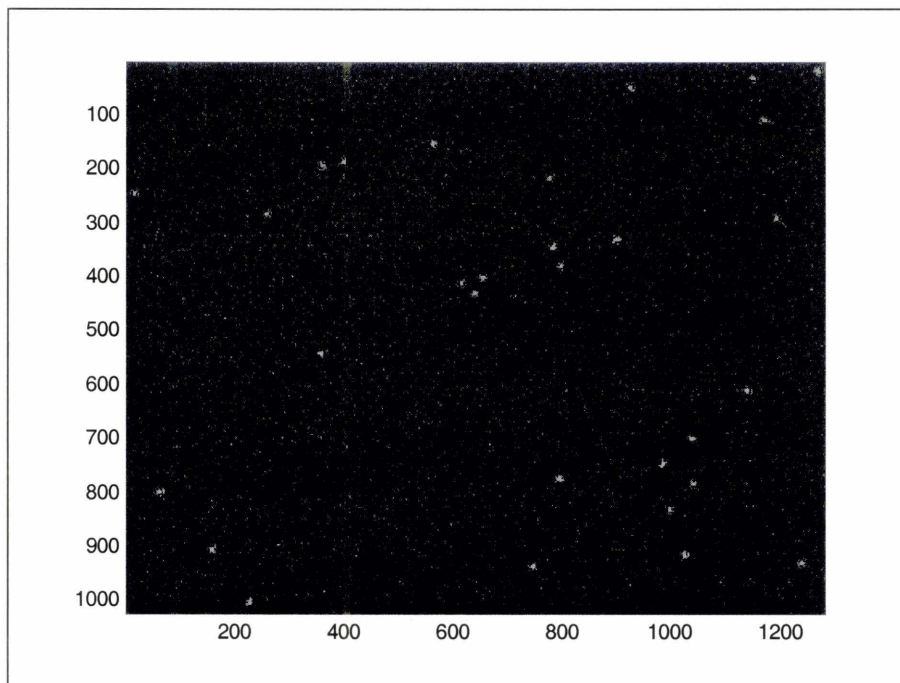


Figure 4.12 Simulated Noisy Image for  $M_V = 4$

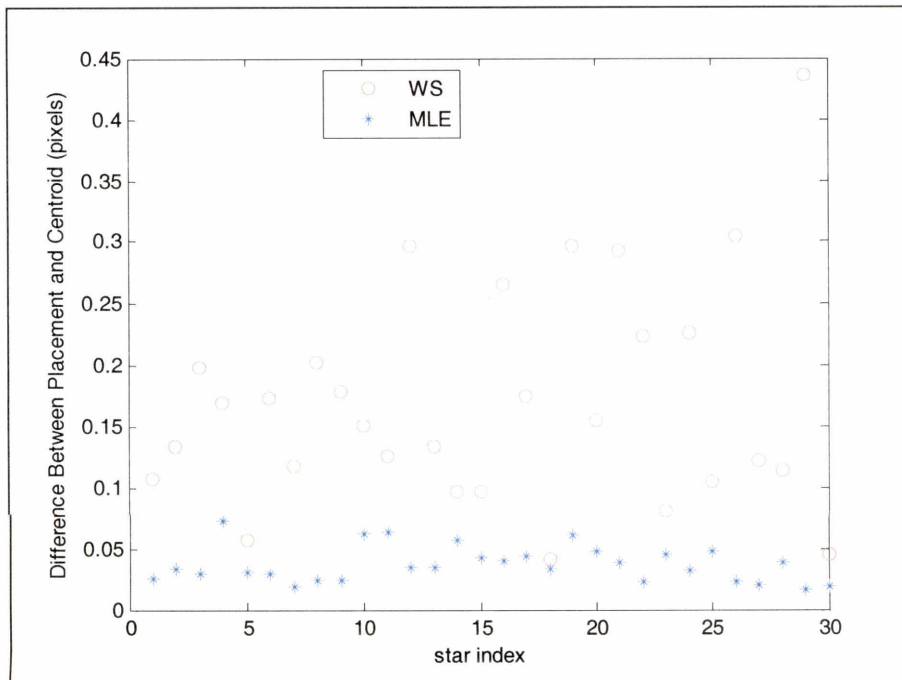


Figure 4.13 Comparison of WS and MLE Centroiding Errors for  $M_V = 4$

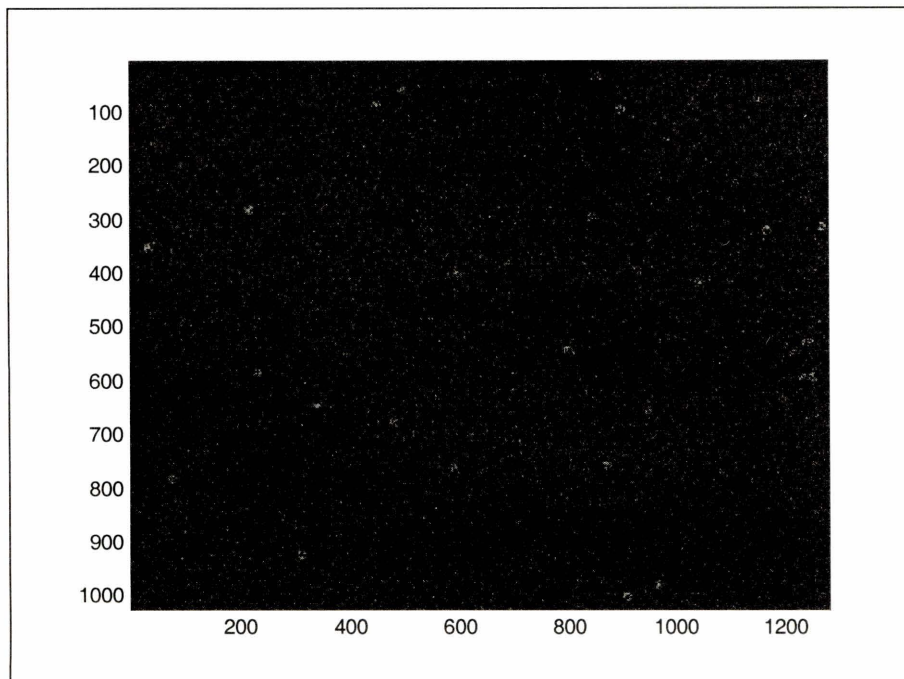
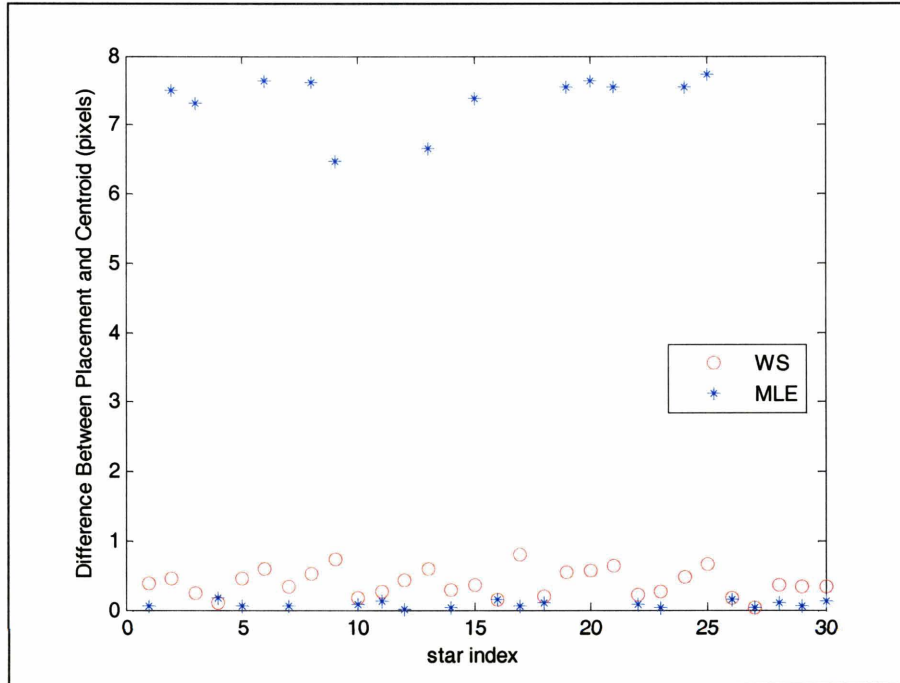


Figure 4.14 Simulated Noisy Image for  $M_V = 5$





**Figure 4.15 Comparison of WS and MLE Centroiding Errors for  $M_V = 5$**

The general trend gained from this trade study is the fact that both the WS and MLE centroiders perform better for brighter stars, as was to be expected. Also, in all cases except for the case when the stars were of  $M_V = 5$ , the MLE centroider performs better. However, the tradeoff with the better accuracy, which is generally on the order of 0.1pixels, is a large computational time requirement, as shown in Table 4.2. The centroiders were evaluated in the Matlab environment; the actual LIST and FAR-MST software will run in the C environment which is significantly faster than the Matlab environment. These times are included to demonstrate a relative comparison between the WS centroider and MLE centroider time requirements.

**Table 4.2 Computational Time Requirements for WS and MLE Centroiders**

$M_V$	WS Time (s)	MLE Time (s)
1	5.74	81.07
2	5.10	77.57
3	2.81	74.56
4	1.61	67.60
5	0.73	46.78

As seen from Table 4.2, the computational time required for the MLE centroider is significantly more than that required for the WS centroider for all five visual magnitudes. In both centroiders, the time decreases for increasing  $M_V$  values because the number of pixels that each centroider must examine decreases for increasing  $M_V$ , since dimmer stars contain fewer pixels that exceed the noise threshold. Because of the enormous increase in computational requirements, it has been decided that the small increase in centroiding accuracy is not worth the time and memory required of the MLE centroider because it would impose requirements unavailable for micro star trackers. Thus the weighted sum centroiders have been selected for use onboard the LIST and FAR-MST star trackers.

### 4.2.3 Predictive Centroiding

Another centroiding technique besides a WS or MLE is a predictive centroider. Samaan, Mortari, Pollock, and Junkin's Predictive Centroiding Technique uses both rate information and image processing to precisely locate stars that fall on an image.<sup>[57]</sup> The centroid locations are first approximated using the current rate information and updated based on local image processing. Current rate information is obtained either via rate gyros or successive attitude estimations from lost-in-space (LIS) algorithms.

When the star tracker is first enabled, no attitude information is available. Therefore, the first image is centroided using a mass moment centroiding method, similar to that used in the various weighted sum centroiders. The mass-moment technique uses a centroid window that encompasses a predefined number of pixels centered on the brightest pixel of the star, and then uses

$$x_{cent} = \frac{\sum_{i,j} x_{ij} I_{ij}}{\sum_{i,j} I_{ij}} \quad (4.23)$$

$$y_{cent} = \frac{\sum_{i,j} y_{ij} I_{ij}}{\sum_{i,j} I_{ij}} \quad (4.24)$$

to determine the  $x$  and  $y$  centroid locations on the imager where  $I_{ij}$  is the intensity of the  $(i,j)$  pixel.<sup>[56]</sup> An attitude matrix that defines the initial attitude in the inertial frame is calculated via a LIS algorithm.

For each consecutive image, predictive centroiding is used to estimate where the stars' centers are located on the imager. The initial angular velocity data is projected forward in time to predict the attitude matrix for subsequent images using the previous attitude matrix and a linear approximation. Along with the stars, vectors associated with the four corners of the FOV are also projected onto the inertial reference frame using this predictive attitude matrix; these four corners are used as the bounds in searching the star catalog, which leads to immediate selection of the stars located in the current image. Stellar locations are predicted and these centers are used as the center of the masks in centroiding the real image. A recursive star identification algorithm uses the star nearest neighbor approach to identify the observed stars in the frame. Finally, the best estimate of angular velocity is used to determine the predicted star location in the next image.

A similar predictive centroiding algorithm is currently under development for the LIST and FAR-MST micro star trackers. The primary advantage of a predictive centroider is the fact that computational time is significantly decreased because the centroider knows ahead of time where on the imager centroids are likely to be located. This centroider uses the current rate information to estimate where the stars will most likely be located in subsequent images. Also, if images are compared to an onboard star catalog, known interstellar distances can be used to better centroid the stars. Finally, since the stars in the catalog are matched a priori with the stars in the image, no additional matching algorithm needs to be employed.



# Chapter 5

## IMAGE PROCESSING: PATTERN MATCHING ALGORITHMS

The primary goal of image processing is to efficiently and accurately determine the attitude orientation and angular rotation rates of the satellite or spacecraft at any time during its mission. Once stars are extracted from noisy images and centroided, numerous pattern matching techniques can be employed to compare images in a frame-to-frame manner or from one frame to an onboard star catalog. Once pattern matching is achieved, attitude quaternions defining the relative or absolute motion of the star tracker can be attained, which is discussed in Chapter 6.

### 5.1 NASA Sky2000 Star Catalog

NASA's Sky2000 Master Star Catalog is a compilation of basic stellar information, including right ascension, declination, position uncertainty, and visual magnitude, for approximately 300,000 stars up to a visual magnitude of 9.<sup>[60]</sup> Based on this star catalog, studies were completed that determined the average and minimum number of stars guaranteed to be within a given FOV up to a specified  $M_V$ . This was accomplished by looping over all fields of view that were appropriate choices for star tracker cameras and all visual magnitudes that the star trackers could resolve, and by counting the number of stars in each FOV while stepping across the entire sky in  $1^\circ$  increments. The number of stars from each FOV increment were averaged and lower-bounded, and these results are displayed in Figure 5.1 and Figure 5.2.

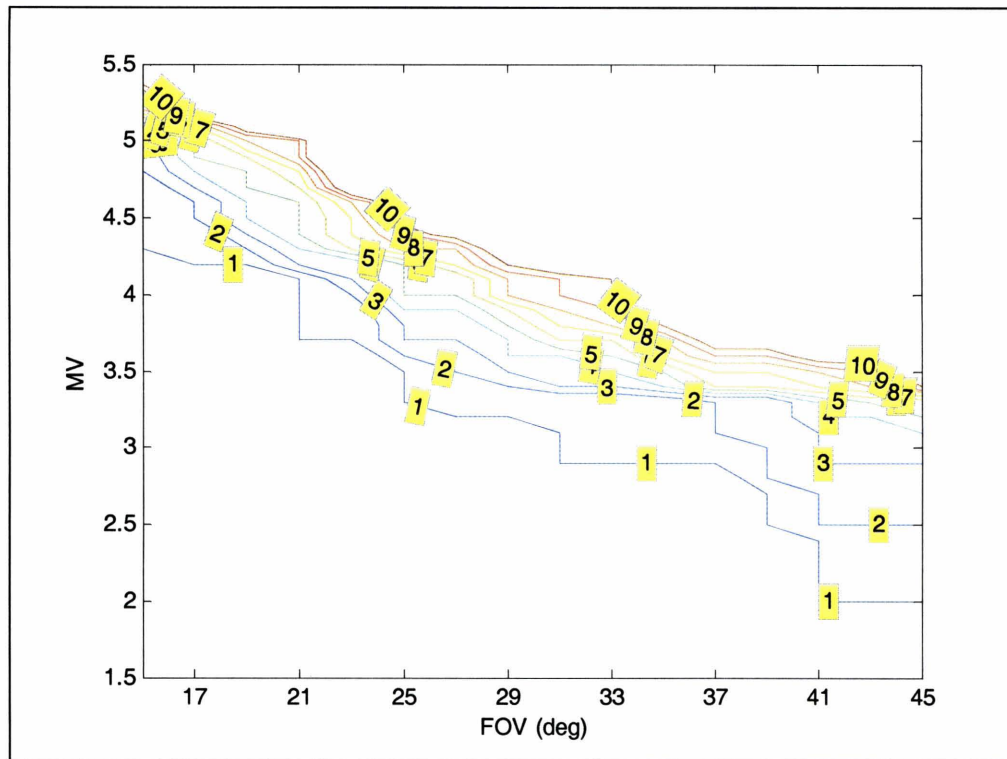


Figure 5.1 Average Number of Stars for Given  $M_V$  and FOV

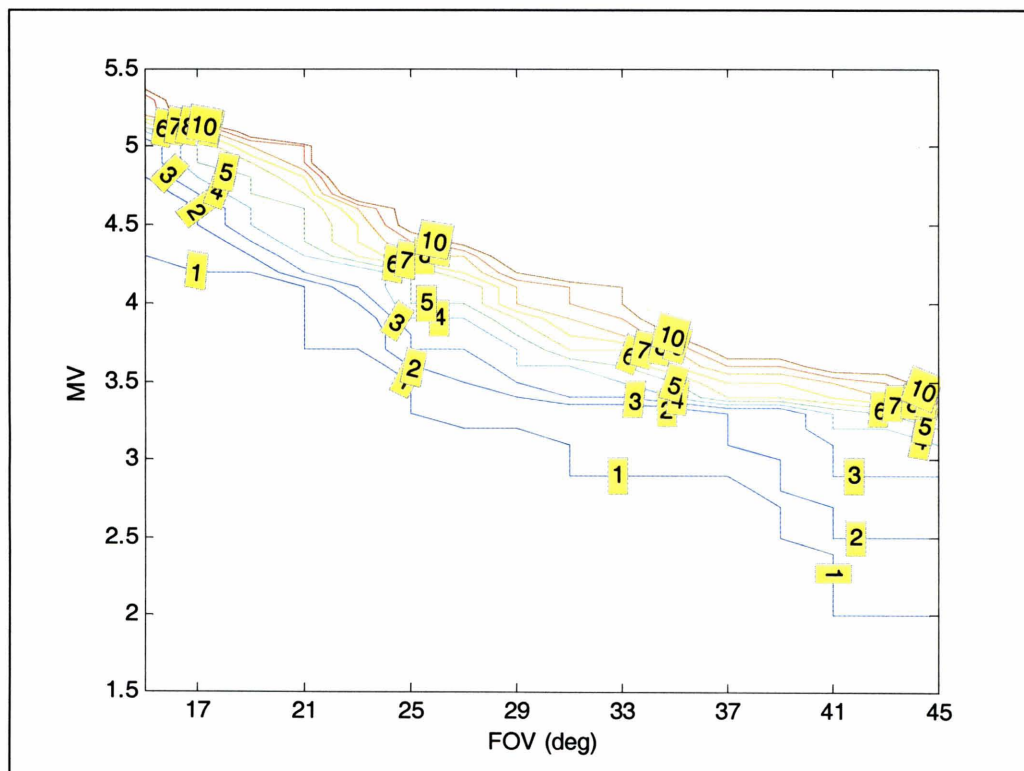


Figure 5.2 Minimum Number of Stars for Given  $M_V$  and FOV

Most micro star trackers will require a visual magnitude of at least 5 and FOV between 25° and 45°. Thus, based on the figure above, any given image taken by the star tracker can expect to have an average and minimum number of stars in the FOV described in Table 5.1.

**Table 5.1 Required  $M_V$  for a Given Minimum Number of Stars and FOV**

FOV (°)	Min # Stars	Required $M_V$
25	4	3.9
	6	4.3
30	4	3.6
	6	3.9
35	4	3.4
	6	3.6
40	4	3.3
	6	3.4
45	4	3.1
	6	3.3

This trade study provides insight concerning the number of stars a given snap-shot can be expected to have available for image processing. It is generally understood that a larger number of stars in a particular image corresponds to a greater likelihood that attitude information can be determined, and a higher probability of accuracy regarding that attitude information, which will be addressed in Chapter 6.

## 5.2 Stellar Matching Techniques

There are numerous ways to process stellar data, and results can provide both relative and absolute attitude information. Relative attitude information can be attained through any of the techniques described below, and is generally easier to acquire than absolute attitude information. In relative image processing, attitude orientations are calculated relative to a previous or future orientation; no information is available to match an image to a definite and specific region of the sky. Absolute attitude information is generally more difficult to attain. It is found by matching stars to an onboard star catalog, which provides an exact match to a known, fixed position in the sky.

The most common image processing algorithms for finding attitude orientation information require matching individual stars or configurations of multiple stars, employing past and current rate information to predict future orientation, and examining stellar streak patterns across highly exposed images. Once stars can be matched between images, relative or absolute attitude quaternions defining the motion of the spacecraft in the sky can be directly computed.

### 5.3 Single Star Matching

The parameter used to match single stars between consecutive images or to an onboard star catalog is visual magnitude. However, the difficulty with matching single stars is that near-perfect noise rejection is required; stars of similar  $M_V$  are likely to be present within the same image and it is likely that noise variations can lead to incorrect measures of stellar intensity. In most cases, unless the stars are extremely bright, brighter than magnitude 0 or 1, the noise characteristics of the star tracker will impede upon the accuracy of the image processing. The chances of very bright stars occurring in the FOV are small, as relatively few stars of this brightness exist in the sky, and integration times will be short to allow for larger tumble rates.

In general, imager measurements of visual magnitude are somewhat uncertain and non-standard because different imagers have different spectral responses. Additionally, imager sensitivity often changes over time. Visual magnitude alone should be used only as a discriminator in the situations where only two stars are observed and no current rate information exists or when stellar centroiding is too erroneous to provide unambiguous interstellar separation angles.<sup>[61]</sup>

Trade studies were completed to describe how noise variations and centroid positioning affect the value of  $M_V$  calculated by the centroider. The purpose of this trade study was to determine the magnitude of difference between an assigned  $M_V$  and the  $M_V$  calculated by the centroider. In this experiment, five  $M_V$  values were studied and compared:  $M_V = 0$ ,  $M_V = 1$ ,  $M_V = 2$ ,  $M_V = 3$ , and  $M_V = 4$ . For each value of  $M_V$  examined, twenty stars were randomly placed across a 1280 x 1024 pixel matrix. Only twenty stars were considered so that there was little chance that stars would overlap due to their random placement and predefined PSF width. This was important because if stars were randomly placed close enough to one another that the

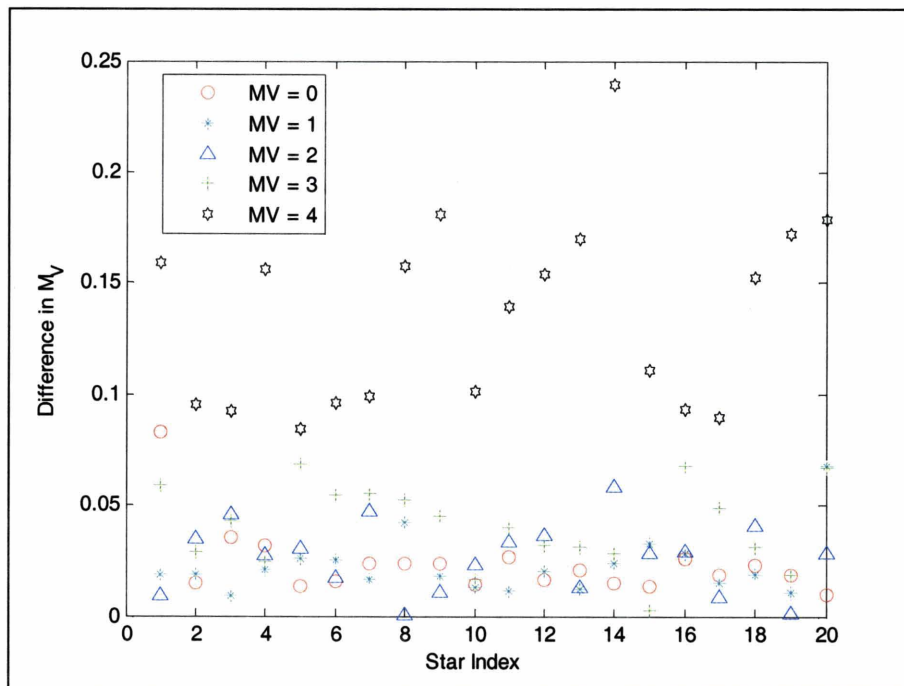


centroider could not distinguish them as separate stars, they would be centroided as a single star with an overall combined  $M_V$  brighter than the  $M_V$  of either individual star. The parameters of the imager and noise properties for this experiment are described in Table 5.2.

**Table 5.2 Parameters for  $M_V$  Trade Study**

<b>a (m<sup>2</sup>)</b>	$\pi \cdot 0.01^2$
<b>FFSR (A/W)</b>	0.12
<b>gain</b>	10
<b>t<sub>shutter</sub> (s)</b>	0.1
<b>psfWidth</b>	10
<b>RO (e-)</b>	$410 \cdot t_{\text{shutter}}$
<b>DK (e-)</b>	40
<b>noise threshold</b>	$5\sigma$

After running the stars through the centroider, each star's  $M_V$  was recalculated and compared to the originally assigned  $M_V$ . Figure 5.3 depicts this difference. In this case, all twenty stars were located and centroided correctly.



**Figure 5.3 Difference Between Assigned  $M_V$  and Calculated  $M_V$**

It is clear from this figure that, in general, brighter stars have smaller differences in calculated  $M_V$  versus assigned  $M_V$ . Dimmer stars, such as those dimmer than  $M_V = 2$ , have differences of over 0.1 magnitude.

This experiment was also run for 500 stars for each  $M_V$  to obtain more statistical information. The code was still run with only twenty stars at a time, and only data sets that centroided all twenty stars correctly were included. The mean difference in assigned and calculated  $M_V$ , as well as its standard deviation, was obtained for each value of visual magnitude and the results are displayed in Table 5.3.

**Table 5.3 Statistical  $M_V$  Data for 500 Stars**

$M_V$	Mean Difference	Standard Deviation of Difference
0	0.0309	0.0427
1	0.0311	0.0324
2	0.0353	0.0419
3	0.0552	0.0414
4	0.1362	0.0685

This data shows a nonlinear increase in both the mean value of the difference and standard deviation of the difference. As visual magnitudes approach values of 4, it becomes difficult to obtain accurate visual magnitude information for randomly placed stars. Data could not be obtained for  $M_V > 4$  without the noise floor being dropped below the  $5\sigma$  value. These studies were not continued for dimmer magnitude stars due to the increased likelihood of noise spikes being centroided as stars.

In all, this trade study supports the fact that visual magnitude should not be used as the sole parameter in matching stars. However, although visual magnitudes should not be used for direct star matching between images, relative visual magnitude information can be used to rank stars in order of brightness. As shown in Chapter 4, stars of bright visual magnitudes have smaller errors in positioning. Thus it will be advantageous to incorporate stellar intensity into matching image processing algorithms to select stars that will provide the best information to obtain attitude information.

## 5.4 Pattern Matching

Theoretically, if no noise was present in the system, relative and absolute attitude information could be obtained from single star matching between images. However, this is a highly unreasonable and impractical requirement for any star tracker or attitude sensor in general. Therefore, it is necessary to determine a different parameter in which stars can be matched between images without relying solely on visual magnitude; the general approach is to use interstellar angles, which is a much more accurate method of determining attitude information, given that the centroiding techniques provide accurate stellar positioning.

### 5.4.1 The Angle Method and Pivoting

Most pattern matching techniques use single or multiple star pairs and match the pairs' interstellar angle between images. This technique is generally deemed the “angle method,”<sup>[62]</sup> which is widely known to provide more accurate matching results than single star matching. Once interstellar angles of star pairs in one image are computed, they are matched to interstellar angles of star pairs in a subsequent image or to cataloged star pairs. This technique is the basis for many pattern matching algorithms because it is relatively simple, computationally inexpensive, and does not rely on other attitude sensors.

If centroiding information were exact, a single star pair matched between several images or from one image to a star catalog should also be exact, and thus an exact attitude quaternion defining the motion of the spacecraft should be found. Unfortunately, the inevitable presence of noise, or even discretization alone, causes the centroids to be somewhat ambiguous, which leads to the possibility that a given image will have multiple star pairs possessing the same interstellar distances when accounting for centroid uncertainties through the requirement of positioning tolerances. Therefore, it is generally beneficial to match several star pairs and intelligently combine the attitude determination results to reduce uncertainty in spacecraft position and orientation.

When considering matching either between images or to an onboard star catalog, an efficient method to use when more than one possibility for an angle match exists is known as

*pivoting*, which is implemented as follows. Given one star pair in the first image with a known interstellar angle, all possible angular matches in the second image or from the onboard catalog are marked. Then, one of the stars in the original star pair from the first image is paired with a third star in the first image. This new pair's interstellar angle is calculated in the first image, and then all possible matches are located in the second image or from the star catalog. Star pairs that constitute both of the angles in the second image or in the catalog are examined, as the two star pairs in the second image or catalog must share a common star. If more than one solution still exists, another pivot star is added and the process is repeated. This pivoting process can continue until a solution is found, or until the number of stars in the FOV is depleted. The unfortunate downside of pivoting is that it quickly becomes computationally expensive when more and more pivots are required for accurate pattern matching.

As can be concluded from the understanding presented above, an image containing a larger number of stars can provide attitude information at a higher confidence level. For example, when considering matching pairs of stars for an image containing  $n$  stars, a total of

$$\binom{n}{2} = \frac{n(n-1)}{2} \quad (5.1)$$

unique star pairs exist. Similarly, when considering matching triads of stars, a total of

$$\binom{n}{3} = \frac{n(n-1)(n-2)}{3!} \quad (5.2)$$

unique triads of stars exist, and when matching quadrilaterals of stars, a total of

$$\binom{n}{4} = \frac{n(n-1)(n-2)(n-3)}{4!} \quad (5.3)$$

unique quadrilaterals of stars exist. Thus, any number of these polygons, or any combination of them, can be used to match between frames.

## 5.4.2 Triangle Matching Algorithms

Numerous triangle matching techniques have been developed for stellar pattern matching.<sup>[63, 64, 65]</sup> They generally combine some form of the angle method, in which the interstellar angles of star pairs are compared either between images or to an onboard catalog. Triangles are advantageous over larger-sized polygons mainly because they are simpler in nature, require fewer stars to be present in the FOV, and enable less computationally expensive algorithms. Examining triangles instead of star pairs adds the additional constraints that three angles must be matched and that each star must be attached to two sides of the triangle, which is beneficial when multiple star pairs with similar interstellar angles create ambiguity.

However, several others in the literature have employed techniques in replacement of or in addition to the angle method when considering triangles. Instead of using the typical side-side-side (SSS) properties of the triangle, which is done when using the angle method alone, side-angle-side (SAS) techniques can also be employed to further reduce ambiguities. Or, by using other triangle properties, such as area and polar moments of inertia, authors have shown an ability to match stars more quickly and more efficiently due to the fact that fewer pivot stars are required for accurate matching. Two of these methods are briefly discussed below.

### ***Spherical Triangle Pattern Matching***

Cole and Crassidis have developed a robust pattern matching technique that employs spherical triangles for fast and accurate pattern matching.<sup>[66]</sup> In their algorithms, spherical triangles from sets of three stars within the FOV are matched to spherical triangles stored within an onboard star catalog. Spherical triangles are advantageous over the angles because more information can be obtained from spherical triangles; the two different properties of spherical triangles, the triangle's area and polar moment of inertia, can be used to quickly eliminate ambiguities when multiple match possibilities arise. The disadvantage of the spherical triangle method is that it requires a large amount of data storage space to balance its speed and accuracy, which may make this method impractical for small-scale star trackers.

Given that there are at least three stars in the FOV, the spherical triangle algorithm can be implemented. The area of a spherical triangle,  $A_{ST}$ , is given by<sup>[67]</sup>

$$A_{ST} = 4 \tan^{-1} \sqrt{\tan\left(\frac{s}{2}\right) \tan\left(\frac{s-a}{2}\right) \tan\left(\frac{s-b}{2}\right) \tan\left(\frac{s-c}{2}\right)} \quad (5.4)$$

with

$$a = \cos^{-1} \left( \frac{\mathbf{b}_1 \cdot \mathbf{b}_2}{|\mathbf{b}_1| |\mathbf{b}_2|} \right), \quad (5.5)$$

$$b = \cos^{-1} \left( \frac{\mathbf{b}_2 \cdot \mathbf{b}_3}{|\mathbf{b}_2| |\mathbf{b}_3|} \right), \quad (5.6)$$

$$c = \cos^{-1} \left( \frac{\mathbf{b}_3 \cdot \mathbf{b}_1}{|\mathbf{b}_3| |\mathbf{b}_1|} \right), \quad (5.7)$$

and

$$s = \frac{1}{2}(a + b + c) \quad (5.8)$$

where  $\mathbf{b}_i$  is the direction of the  $i^{\text{th}}$  star in the star tracker body frame. Additionally, the standard deviation of the area error for a spherical triangle can be found using a linearization approach.<sup>[68]</sup> This information is included as both a measurement of precision and a means of bounding the ranges of possible areas that correspond to the desired spherical triangle.

In general, two spherical triangles that have the same area can have very different polar moments of inertia and vice-versa. Thus both of these properties can be used simultaneously to greatly reduce the number of possible matches. The polar moment of inertia of the spherical triangle,  $I_{PMST}$ , is calculated by dividing the spherical triangle into  $n$  smaller spherical triangles.

$$I_{PMST} = \sum_{i=1}^n \theta^2 dA \quad (5.9)$$

where  $\theta$  is the arc distance from the centroid of the smaller spherical triangle to the centroid of the overall spherical triangle and  $dA$  is the area of each smaller spherical triangle. Larger values of  $n$  produce a greater accuracy in polar moment calculation, but can also greatly increase the

computational expense. The standard deviation in error of the polar moment of inertia is difficult to analytically compute. However, a Monte Carlo simulation of 1000 random spherical triangles from a star catalog with 100 polar moment measurement calculations per triangle demonstrated that the probability that the  $I_{PMST}$  will be within a  $3\sigma$  bound would occur at least 99.7% of the time, as to be expected.

### ***Planar Triangle Pattern Matching***

Two years following the development of the spherical triangle pattern matching technique, Cole and Crassidis designed a planar triangle pattern matching algorithm.<sup>[69]</sup> Just as the spherical triangle algorithm, the planar triangle algorithm uses the area and polar moment of triangles to match stars to an onboard catalog.

The area of a planar triangle,  $A_{PT}$ , is calculated using Heron's equation

$$A_{PT} = \sqrt{s(s-a)(s-b)(s-c)} \quad (5.10)$$

with

$$a = \|\mathbf{b}_1 - \mathbf{b}_2\|, \quad (5.11)$$

$$b = \|\mathbf{b}_2 - \mathbf{b}_3\|, \quad (5.12)$$

$$c = \|\mathbf{b}_3 - \mathbf{b}_1\|, \quad (5.13)$$

and

$$s = \frac{1}{2}(a+b+c) \quad (5.14)$$

where  $\mathbf{b}_i$  is the direction of the  $i^{\text{th}}$  star in the star tracker body frame. The polar moment for a planar triangle,  $I_{PMPT}$ , is given by

$$I_{PMPT} = A_{ST} \frac{(a^2 + b^2 + c^2)}{36}. \quad (5.15)$$

In general, the planar triangle pattern matching and spherical triangle pattern matching techniques have relatively similar performance characteristics. They are both approximately one-and-a-half times more successful than the angle method alone and require approximately one-third as many pivots as the angle method. As a result, they are also computationally more efficient. Due to the complexity of the formulas for the spherical triangle method, the planar triangle method is more computationally efficient.

### 5.4.3 Rate Matching

Rate matching techniques allow for faster pattern matching due to their ability to incorporate current rate information to predict where stars will be located in future images. The obvious disadvantage is that this method only works when accurate rate information is readily available. The effectiveness of rate matching algorithms is defined by their ability to accurately predict future rate information. In general, the reliability of rate matching is dependent upon the speed at which the spacecraft is tumbling; faster tumble rates require images to be taken more frequently and processed more quickly.

Several predictive pattern matching techniques have been proposed in the literature. For example, Samaan, Mortari, and Junkins have proposed two recursive mode star identification algorithms.<sup>[70]</sup> The spherical polygon approach accesses all cataloged stars observed by the imager FOV and recursively adds or removes candidate cataloged stars according to the predicted image motion, which is based on current rate information. In this algorithm, star identification is accomplished by matching interstellar angles. The star neighborhood approach uses star neighborhood information and a cataloged neighborhood pointer matrix to access the star catalog. Previously identified stars are used to access candidate stars to match with the current stars. This technique is also dependent on the stellar locations with respect to the four corners of the image frame.

### 5.4.4 Selecting Stars to Process

All stars appearing in one image would ideally be matched to all stars in the following image. However, this is rarely the case, as some amount of time is required to process the stellar



data for one image before the camera can record a second image, and so consecutive images must be separated by a specific amount of time. As a result of this time passage, the satellite will most likely rotate in such a manner that causes particular stars fall off of or onto the imager plane.

However, in reality, based on integration time, rotation rates, and noise, stars are continually falling on and off of the imager plane. Several issues must be considered when selecting the best stars to match between images. Contributing factors are stellar intensity, absolute location on the imager, and relative distance from other stars. In general, the matching techniques for LIST and FAR-MST begin by using the brightest stars on the image because they have the most accurate centroids. However, the stars are still selected in a manner that does not assume that a selected star will remain in consecutive images.

## **5.5 LIST and FAR-MST Matching Techniques**

Four stellar pattern matching algorithms have been proposed for determining relative attitude information between consecutive images. The distance match algorithm and triangle match algorithm define pairs and triads of stars, respectively, in one image and matches them to the same pairs or triads of stars in a subsequent image. Once stellar matches are found between images, the attitude quaternion that defines the star tracker's angular motion can be determined. The rate match algorithm uses the current attitude rate information to quickly and precisely estimate where stars are likely to be located in the present image based on the stars' locations in previous images. Again, once the star matches are found, attitude quaternions are easily generated. Each of these algorithms can also lead to absolute attitude information. The stars can be directly matched to an onboard star catalog, usually sorted by interstellar separation angles, and once a match is determined, a previous or subsequent attitude quaternion can be translated into an absolute attitude quaternion based on the desired frame of reference. The n-vertex match algorithm matches n-sided polygons from one image to an onboard star catalog and absolute attitude information is automatically generated.

### 5.5.1 Pair Match

The pair match algorithm matches pairs of stars between consecutive frames. The algorithm first centroids all stars in both images and creates a distance matrix for each image that defines the distances between all pairs of stars on that image. The algorithm then selects stars to match between the two images based on their visual magnitude, angular separation, and centroided position on the imager; brighter star pairs with large interstellar distances that are located in places on the imager that will most likely remain in consecutive images are selected first. Based on the stars' locations relative to the center pixel on the image in the first frame, the stars can be uniquely matched to those in the second frame based on the motion of the center pixel into the second frame. Orientation is further discriminated by visual magnitude and other nearby stars.

This method is robust in that it can account for stars that fall off the edges of the imager between the first and second images due to rotation and for new stars that appear in the second image. The disadvantage of this pair match is that there are numerous special cases that must be taken into account. The most likely situation that cannot be solved by simple pair matching is dealing with star pairs that have the same interstellar angles that are located on the same frame. Given that actual images are tainted with noise variations, the resulting uncertainties in centroiding locations will require positioning tolerances to be placed on each star, making it easy to have several star pairs in a given image having the same interstellar distances. Because of this uncertainty, the pair match is not used onboard LIST or FAR-MST unless only two stars are present in the image.

### 5.5.2 Triangle Match

The triangle match was created to deal with the issue of multiple interstellar distances occurring in the same image. Instead of examining two stars simultaneously, this algorithm examines three stars concurrently. All stars are first sorted from brightest to dimmest. An angle matrix is created for each image that defines all of the interstellar angles for each star pair. Then, beginning with the brightest stars, all triads of stars are methodically examined until a match

between images is established. One general looping structure to select the three stars, which covers all possibilities of star triads for  $N$  stars, is

$$\text{Loop1} = 1 : N - 2$$

$$\text{Loop2} = \text{Loop1} + 1 : N - 1$$

$$\text{Loop3} = \text{Loop2} + 1 : N$$

The disadvantage of this looping is that it does not effectively consider the fact that a star in the first image may not appear in the second image; a star in the first image could easily fall out of the camera FOV or drop below the noise floor. If, for example, star #1 does not occur in the second image, considerable time is wasted in attempts to match a triangle containing that star that does not even occur in the second image. Therefore, a smart looping structure has been implemented to prevent any given star from being used as one star in the triangle for more than three consecutive iterations. This looping structure still considers the brightest stars first, but removes them systematically in case they are not present in the second frame:

$$\text{Loop1} = N : -1 : 3$$

$$\text{Loop2} = \text{Loop1} - 1 : -1 : 2$$

$$\text{Loop3} = \text{Loop2} - 1 : -1 : 1$$

This looping seeks to maximize the changing of stellar indices from one triad selection to the next; this maximization defines the optimal looping structure.<sup>[71]</sup> However, this particular looping structure loops over the stars to exclude, whereas most algorithms loop over stars to keep.

Once a triad of stars is selected in the first image, the interstellar angles of each star pair in the triad is searched for in the angle matrix of the second image. The star indices in the second image that match the interstellar angles found in the first image are listed in a candidate star matrix. It is important to note that multiple star pairs in the second image may have the same interstellar angles, but all star pairs that contain one of the angles defined by the legs of the

triangle are recorded in the candidate star matrix. Examination of the candidate star matrix allows for its reduction to uniquely determine which three stars in the second image are the match to those in the first image.

For example, suppose that the three stars labeled **1**, **2**, and **3** in Figure 5.4 are the three stars in the first image that have been selected to be matched to stars in the second image. The interstellar distances for this triangle are **a**, **b**, and **c**, as displayed in the figure.

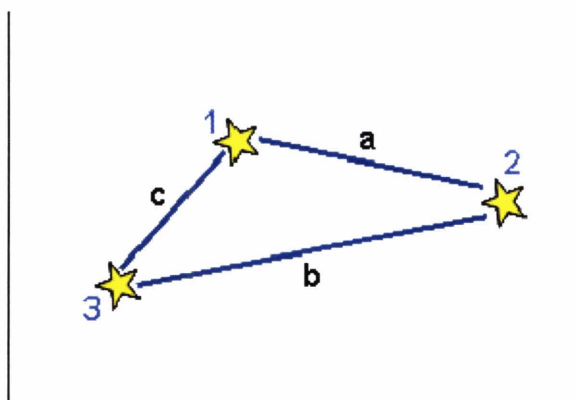


Figure 5.4 Stars to Match from Image 1

The second image is scanned for all star pairs that have interstellar distances of either **a**, **b**, or **c**. Suppose the results of this scan are as displayed as in Figure 5.5; the candidate star matches in the second image are star numbers **3**, **4**, **5**, and **9**.

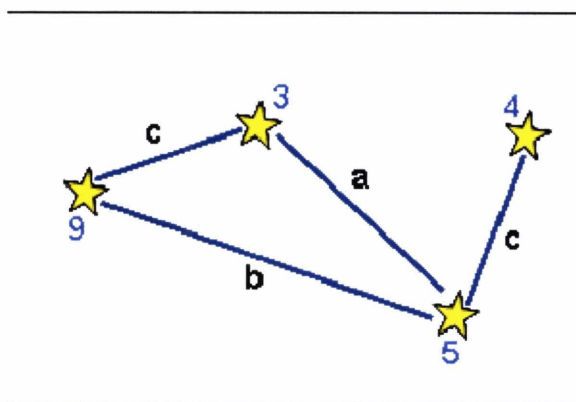


Figure 5.5 Candidate Matching Stars from Image 2

The candidate star matrix is formed using the new star indices from the second image, as shown in Figure 5.6. Note that since stars have not yet been uniquely determined they must be listed a second time in the candidate star matrix, in reverse order.

star # in image 1 {	1	2	3	
	3	5		← a
	5	3		← a
		9	5	← b
candidate star # in image 2 {		5	9	← b
	3		9	← c
	9		3	← c
	5		4	← c
	4		5	← c

Figure 5.6 Candidate Star Matrix

The rows of the candidate star matrix can be eliminated as follows. First, consider each column of the candidate star matrix. If any particular star index in that column does not appear at least twice, that entry and its entire row can be removed because that star index cannot be a star candidate if it does not join at least two sides of the triangle. Thus, from this example, candidate stars 4 and 9 each appear only once in the first column, so these rows can be eliminated from the candidate star matrix. Similarly, candidate stars 3 and 9 appear only once in the second column, and so their respective rows can be removed from the matrix. In this case only two columns must be reviewed for elimination, as all remaining stars occur twice in each column. This reduction is illustrated in Figure 5.7.

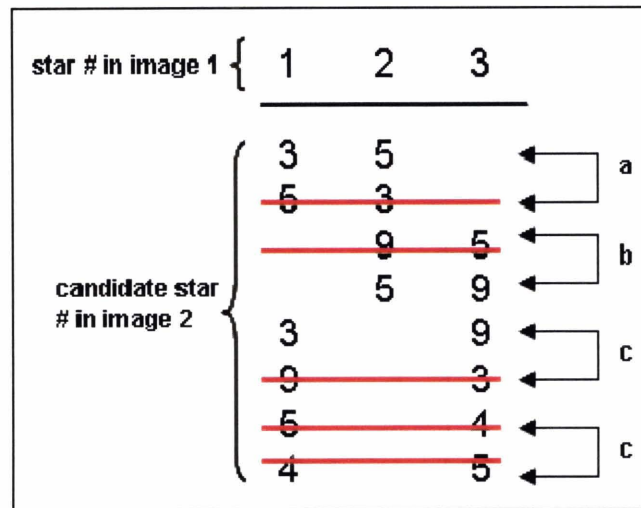


Figure 5.7 Row Reduction of Candidate Star Matrix

Once all appropriate rows are removed, the resulting matrix will always contain either zero rows or three rows. If three rows remain, a triangle has been matched, and the correct star match to the star number in the first image is the value left in that column of the candidate star matrix. If zero rows remain, the algorithm could not uniquely match the three stars in the first frame to three stars in the second frame. In these instances one of the stars selected in the first image did not appear anywhere in the second image; it either moved outside of the appropriate FOV or fell below the noise threshold. The only way for this algorithm to fail is if duplicate triangles exist in the same image and an erroneous match is made.

### 5.5.3 Rate Match

The main idea behind rate matching algorithms is to predict stellar locations based on previously determined rate information. This is accomplished by knowing centroided star positions for both images, quaternion rate information, and time differences between the images to compare. Stellar locations are projected from the first image into predicted locations in the second image based on the current attitude information; stars that are predicted to fall outside of the FOV are ignored. Stars are then matched in the second frame to the stars located nearest to the predicted locations. The rate match algorithm outputs a star list that maps the stars in the first image to stars in the second image.

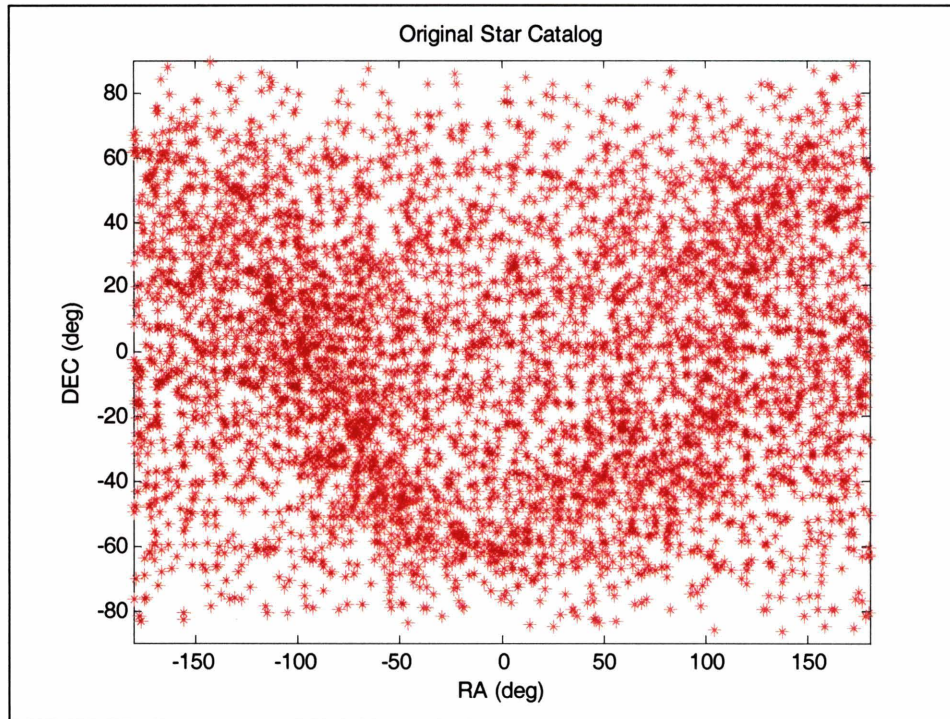
The primary advantage of this technique is that since stellar locations are predicted, the area of the pixel matrix that must be searched is greatly reduced. Additionally, it is robust enough to handle faster rotation rates. The clear disadvantage is that the algorithm requires rate information from previous images and assumes this rate is constant and can be directly applied to subsequent images. This may only be a valid assumption for very short periods of time, which could be problematic if the star tracker is facing a direction in the sky that is devoid of many stars. Hence, the primary challenge of rate matching algorithms is in predicting how far into the future rate information can be projected.

#### **5.5.4 N-vector Match**

A fourth algorithm, the n-vector match algorithm, has been implemented to solve lost-in-space problems, where no attitude information is known a priori. This algorithm utilizes the k-vector<sup>[61]</sup> approach to access an onboard star catalog in a searchless manner, thus providing immediate matching information to the star tracker. The onboard star catalog is based on a specified  $M_V$  and camera FOV, which is based on the resolving power of the imager and physical size of the optics subsystem. The directory contains all cataloged star pairs whose interstellar angles are less than the camera FOV and within the limits of resolvable  $M_V$ . The star pairs are ordered by increasing interstellar angles, and the value stored in the catalog is the sine-squared of the interstellar angle.

#### ***Creating a Reduced Star Catalog***

A reduced version of the NASA SKY2000 Master Star Catalog was created to be stored onboard the star tracker. This catalog was created by searching the entire sky in one degree increments and selecting only the brightest stars in the current FOV to be included in the catalog. The original star catalog of all stars in the NASA SKY2000 Master Star Catalog brighter than  $M_V = 6$  is shown in Figure 5.8. There are 5060 stars in this catalog.



**Figure 5.8 All Stars in NASA SKY2000 Master Star Catalog Brighter than MV6**

After the reduction, which included keeping the eight brightest stars in each FOV, the catalog was reduced to 540 stars, as shown in Figure 5.9.



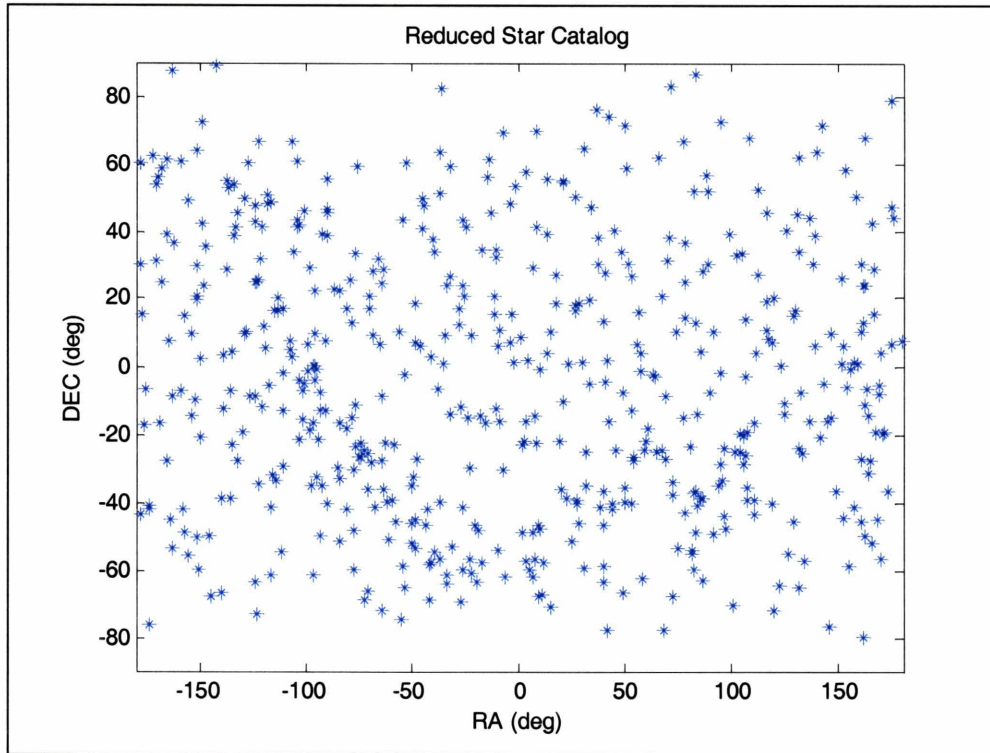


Figure 5.9 Reduced Version of NASA SKY2000 Master Star Catalog

### Double Star Reduction

Stars whose interstellar separation angles are indistinguishable due to the low resolving power of the lens can be combined into a single star, and are listed in the onboard star catalog as a single star. This is accomplished by locating an overall centroid position on the imager and calculating an overall visual magnitude. The combined visual magnitude,  $M_V$ , is found by

$$M_V = M_{V,2} - 2.5 \log \left( 1 + e^{\frac{(M_{V,2} - M_{V,1})}{2.5}} \right) \quad (6.1)$$

where  $M_{V,1}$  and  $M_{V,2}$  represent the individual visual magnitudes of the two stars. The combined centroid unit vector representing the combined star weighs the two unit vectors by their stellar intensity

$$\mathbf{v} = \sum_{i=1}^2 \left( \mathbf{v}_i e^{\frac{(M_V - M_{V,i})}{2.5}} \right) \quad (6.2)$$

where  $v_i$  is the unit vector for the current star.<sup>[3]</sup>

### **Projection of Stars onto the Celestial Sphere**

Prior to accessing the star catalog in the n-vertex match algorithm, centroided stars are projected from locations on the imager in the x-y plane to positions on the celestial sphere. The position on the celestial sphere is defined by a unit vector pointing towards the position on the celestial sphere. The conversion from a centroid location into a unit vector is

$$\mathit{unitVect} = \begin{bmatrix} \mathit{unitVect}_x \\ \mathit{unitVect}_y \\ \mathit{unitVect}_z \end{bmatrix} = \begin{bmatrix} x \frac{pp_x}{f} \left( 1 + \left( x \frac{pp_x}{f} \right)^2 + \left( y \frac{pp_y}{f} \right)^2 \right)^{-\frac{1}{2}} \\ y \frac{pp_y}{f} \left( 1 + \left( x \frac{pp_x}{f} \right)^2 + \left( y \frac{pp_y}{f} \right)^2 \right)^{-\frac{1}{2}} \\ \left( 1 + \left( x \frac{pp_x}{f} \right)^2 + \left( y \frac{pp_y}{f} \right)^2 \right)^{-\frac{1}{2}} \end{bmatrix} \quad (6.3)$$

where  $x$  and  $y$  are the centroid  $x$  and  $y$  coordinates on the imager plane, respectively,  $pp_x$  and  $pp_y$  are the pixel pitches of the imager in the  $x$  and  $y$  directions, respectively, and  $f$  is the focal length of the lens. If the imager contains square pixels, (6.3) can be reduced to

$$\mathit{unitVect} = \begin{bmatrix} \mathit{unitVect}_x \\ \mathit{unitVect}_y \\ \mathit{unitVect}_z \end{bmatrix} = \begin{bmatrix} x \frac{pp}{f} \left( 1 + (x^2 + y^2) \left( \frac{pp}{f} \right)^2 \right)^{-\frac{1}{2}} \\ y \frac{pp}{f} \left( 1 + (x^2 + y^2) \left( \frac{pp}{f} \right)^2 \right)^{-\frac{1}{2}} \\ \left( 1 + (x^2 + y^2) \left( \frac{pp}{f} \right)^2 \right)^{-\frac{1}{2}} \end{bmatrix}. \quad (6.4)$$

### The k-vector

The n-vertex match is efficient due to its use of the k-vector technique. The k-vector works in a manner similar to a hashing function, whose purpose is to arrange data in a manner conducive to rapid access. The k-vector allows for the avoidance of searching an entire star catalog, as it points directly to the indices of the admissible range of stellar pairs.

Let  $A$  be the column vector of  $n$  elements representing the cosine of all interstellar angles and sorted between 0 and 1. The plot of  $A$  is shown below in Figure 5.10.

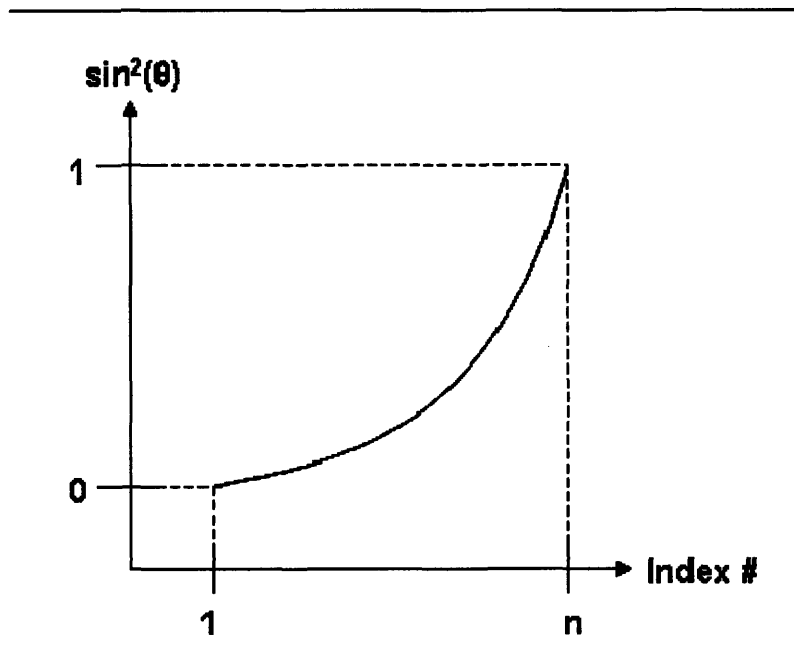


Figure 5.10 Plot of  $A$

Because there are  $n$  elements, an  $n^{\text{th}}$  order polynomial is required to uniquely define this curve. However, this curve can be approximately by a linear curve as shown below in Figure 5.11.

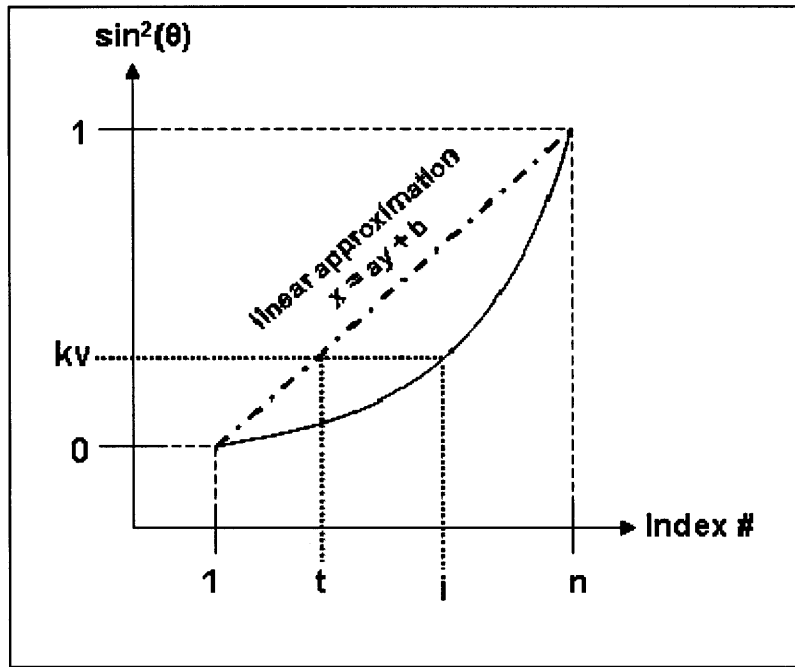


Figure 5.11 Linear Approximation of Plot of A

Let  $B$  be the column vector of  $n$  elements representing the linear approximation of the sine-squared of all interstellar angles and sorted between 0 and 1. Both the cosine of the interstellar angle and the sine-squared of the interstellar angle are appropriate candidates since they present nearly linear distributions over the list of star pairs. The sine-squared values are chosen because sine-squared of theta is two times more sensitive to changes in theta than the cosine of theta is, as

$$\frac{d}{d\theta}(\cos(\theta)) = -\sin(\theta) \approx -\theta \quad (6.5)$$

and

$$\frac{d}{d\theta}(\sin^2(\theta)) = 2\sin(\theta)\cos(\theta) \approx -2\theta. \quad (6.6)$$

The goal of the  $k$ -vector is to find the index number  $i$  by first determining the value of  $t$ . This is accomplished by plugging the known value  $kv$  into the linear approximation  $kv = at + b$  and solving for  $t$ . This value  $t$  is the index into  $B$ , and the value in  $B$  at index  $t$  is the index into  $A$ , which is  $i$ .<sup>[61]</sup>

### ***The n-vertex Match Algorithm***

The n-vertex match is a modified version of Mortari et. al's Pyramid match.<sup>[71]</sup> The Pyramid match selects a triad of centroided stars in a given image and matches them to the onboard star catalog via the k-vector approach. Once a successful match is found, a fourth star is added to the triad and checked against the catalog to increase the confidence of stellar identification. If a fourth star cannot be successfully matched, then the algorithm searches for a new triad of stars and repeats the process until four stars, a “pyramid,” are successfully matched to the catalog. One advantage of this original algorithm is that it can be extended to search for an n-star polygon to increase the confidence level to one that adequately meets the requirements of the star tracker.

Three versions of the n-vertex match algorithm have been created, one each for values of n equal to 3, 4, and 5. These modified versions of the pyramid algorithm hold slightly higher image requirements, but accelerate the search process because instead of matching three stars first and then matching a fourth, a match on all four stars is attempted at once. The same looping structure as found in the triangle match is implemented to maximize the changes in the index of stars from one pyramid of stars to the next. The algorithm does not terminate until an n-sided polygon of stars is successfully matched to the catalog, or all star pair candidates have been exhausted.

As an example of how the n-vertex match algorithm works, consider the n-vertex match when n is equal to 4. Let  $Y$  be the matrix containing all the star pair indices in the onboard star catalog within the camera field of view. Suppose four stars, **A**, **B**, **C**, and **D** are selected to match on a given image. These stars are connected to one another by six baselines numbered **1** through **6**, as shown in Figure 5.12.

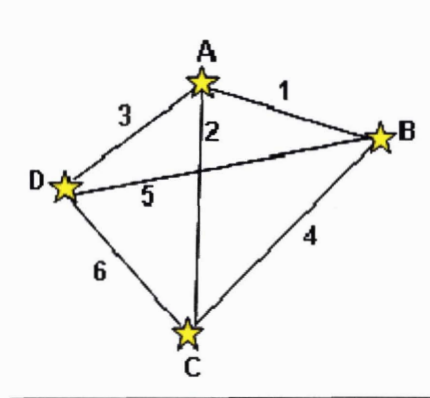


Figure 5.12 N-Vertex Match Example with 4 Stars

To determine the four stars in the catalog that match the stars on the image, a candidate star matrix  $C$  is created that contains six columns, or

$$\frac{n(n-1)}{2}, \quad (6.7)$$

one for each baseline of the 4-vertex pattern. The elements of  $C$  are the indices of all of the candidate stars pairs determined by the  $k$ -vector. Since star **A** must be connected to stars **B**, **C**, and **D**, star **A** must be listed as an index in columns 1, 2, and 3 of  $C$ . This knowledge would theoretically be enough to uniquely match the 4-vertex star pattern, but the imprecision in the centroider prevents this from occurring in all circumstances.

To accommodate for the inaccuracy of the centroider, the second matrix,  $S$ , is built with one row for each star in the catalog that is selected by the  $k$ -vector. The entries of the  $S$  matrix are determined as follows. Suppose, for example, the  $k$ -vector selected stars with indices 45 and 46 as candidate stars corresponding to baseline 1 (these entries are found in column 1 of the  $C$  matrix). Thus, in column 1 of  $S$ , a 46 is entered in row 45 and a 45 is entered in row 46. This says for baseline 1, star 46 could be either star **A** or star **B**, and if it is determined to be either star **A** or star **B**, star 45 is the other star. In the same manner, all of the remaining entries in column 1 of  $C$ , are appropriately placed in the  $S$  matrix. This is also done for the remaining five baselines.

It has already been determined that any candidate for star **A** must appear in columns 1, 2, and 3 of  $S$ . It follows that any candidate for star **B** must appear in columns 1, 4, and 5, any

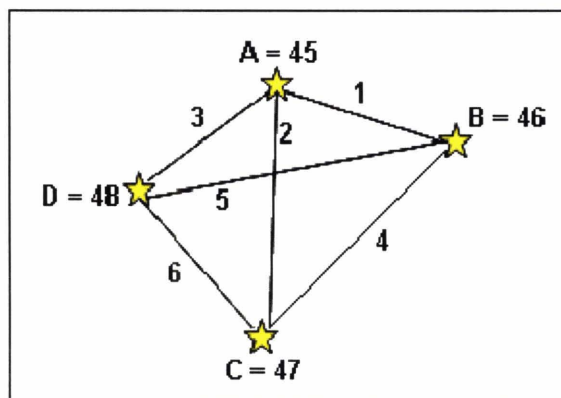
candidate for star **C** must appear in columns 4, 5, and 6, and any candidate for star **D** must appear in columns 3, 5, and 6. With this knowledge, a “self-consistency” check is performed to assign star numbers to **A**, **B**, **C**, and **D**.

Suppose the *S* matrix is created to be as shown in Table 5.4.

**Table 5.4 Results of S Matrix Construction**

row #	1	2	3	4	5	6
...	...	...	...	...	...	...
45	46	47	48			
46	45			47	48	
47		45		46		48
48			45		46	47
...	...	...	...	...	...	...

It can easily be determined that the stars are matched as shown in Figure 5.13.



**Figure 5.13 Results of N-Vertex Match Example**

In most cases, a unique solution can be found by this method alone. If, however, a set of non-unique solutions are found, a fifth star can be considered. Either that fifth star can replace one of the four original stars so that four are matched again, or all five can be matched together. Further trade studies need to be investigated to determine whether or not matching multiple patterns of four stars or adding a fifth star will provide the highest level of performance at the lowest computational cost.





# Chapter 6

## IMAGE PROCESSING: DETERMINING THE ATTITUDE QUATERNION

### 6.1.1 Wahba's Loss Function

If measured and reference vectors are error free, the rotation matrix between body frame to reference frame is always the same. However, if measurement errors exist (e.g. noise) a least-squares method is used to minimize the weighted sum of the squares of the observed residuals.<sup>[72]</sup>

The most common method for determining the attitude of the spacecraft is through implementation of a loss function, which was first posed by Wahba in 1965.<sup>[73]</sup> In terms of attitude determination, the least squares problem to be solved is defined as

$$L(A) \equiv \frac{1}{2} \sum_{i=1}^k |\mathbf{b}_i - A\mathbf{r}_i|^2 \quad (6.8)$$

where  $\mathbf{b}_i$  and  $\mathbf{r}_i$  are the unit vectors describing the stars on the image and the corresponding stars from the catalog; the goal is to find the attitude matrix  $A$  that minimizes the loss function  $L$ .<sup>[73]</sup>

Originally, the  $A$  matrix was calculated directly, but for practicality in application, algorithms were created to solve for a quaternion that represented the attitude matrix.<sup>[68]</sup> The quaternion is advantageous because it provides the same amount of information as the attitude matrix, but requires only four elements instead of nine. The quality of the attitude matrix or attitude quaternion is expressed statistically by a covariance uncertainty matrix. The methods considered for LIST and FAR-MST star trackers include Crassidis and Markley's Predictive Attitude Determination algorithm, Davenport's q-method and Shuster's QUEST algorithm, and Markley's Singular Value Decomposition algorithm and his updated version, the Fast Optimal Matrix algorithm, all of which are discussed below.

### 6.1.2 The Predictive Attitude Determination Algorithm

The Predictive Attitude Determination (PAD) algorithm is derived from a general nonlinear predictive filter approach and holds the primary advantage over other algorithms in that it is able to be applied when anisotropic measurements errors exist. The attitude model is assumed to be given by the quaternion kinematics model, which does not require a dynamics model that generally requires a complicated Kalman filter approach.

The attitude rate can be modeled by a constant model error  $d$  between measurements

$$\dot{q} = \frac{1}{2} \Xi(q) d. \quad (6.9)$$

The attitude matrix that minimizes Wahba's loss function is related to the quaternion  $q \equiv [q_1 \ q_2 \ q_3 \ q_4]^T$  via

$$A(q) = -\Xi^T(q) \Psi(q) \quad (6.10)$$

where

$$\Xi(q) = \begin{bmatrix} q_4 I + \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \\ -q_{13}^T \end{bmatrix} \quad (6.11)$$

and

$$\Psi(q) = \begin{bmatrix} -q_4 I + \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{bmatrix} \\ q_{13}^T \end{bmatrix} \quad (6.12)$$

with  $I$  being the 3 x 3 identity matrix.

A covariance matrix statistically describes the amount of errors in the attitude matrix  $A$ . Assuming that the errors between the vector measurement sets are uncorrelated, the model error is shown to be

$$\mathbf{d}(t_k) = \frac{1}{\Delta t} \left\{ \sum_{i=1}^n [A(\mathbf{q}_k) \mathbf{r}_i \times] R_i^{-1} [A(\mathbf{q}_k) \mathbf{r}_i \times] \right\}^{-1} \sum_{i=1}^n [A(\mathbf{q}_k) \mathbf{r}_i \times] R_i^{-1} (\mathbf{b}_i(t_{k+1}) - A(\mathbf{q}_k) \mathbf{r}_i) \quad (6.13)$$

using the shorthand notation for the cross product operator. The cross product  $\mathbf{c} = \mathbf{a} \times \mathbf{b}$  can be expressed as  $\mathbf{c} = [\mathbf{a} \times] \mathbf{b}$  where the matrix  $[\mathbf{a} \times]$  is defined as

$$[\mathbf{a} \times] \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \quad (6.14)$$

From this, the attitude error covariance is found to be

$$P_{pad} \approx \left[ \sum_{i=1}^n [A(\mathbf{q}_k) \mathbf{r}_i \times] R_i^{-1} [A(\mathbf{q}_k) \mathbf{r}_i \times]^T \right]^{-1} \quad (6.15)$$

If the measurement errors are isotropic, (6.13) and (6.15) can be reduced to

$$\mathbf{d}(t_k) = \frac{1}{\Delta t} \left\{ \sum_{i=1}^n \sigma_i^{-2} [A(\mathbf{q}_k) \mathbf{r}_i \times]^2 \right\}^{-1} \sum_{i=1}^n \sigma_i^{-2} [A(\mathbf{q}_k) \mathbf{r}_i \times] \mathbf{b}_i(k+1) \quad (6.16)$$

and

$$P_{pad} \approx \left[ \sum_{i=1}^n \sigma_i^{-2} [A(\mathbf{q}_k) \mathbf{r}_i \times]^2 \right]^{-1} \quad (6.17)$$

respectively, where (6.17) is equivalent to the QUEST covariance, discussed next.

### 6.1.3 The q-method and the QUEST Algorithm

The q-method provides an optimal least-squares estimate of the attitude of a spacecraft, given two sets of vector measurements: one in the body frame and the same set in a reference

frame.<sup>[74]</sup> The quaternion estimation, QUEST, algorithm approximates the q-method, making it more computationally efficient.<sup>[75]</sup> Because of its ease of implementation, efficiency, and relatively high accuracy the QUEST algorithm is the method of choice for determining the attitude quaternion of small-scale satellites and spacecraft. It has been implemented for all of the pattern matching algorithms discussed in Chapter 5, and has proven to provide sufficiently accurate attitude information to meet LIST and FAR-MST goals.

The key to both the q-method and the QUEST algorithm is to define, recognize, and solve an eigenvector problem, which can be derived from Wahba's loss function described in (6.8). This loss function can be redefined in terms of quaternions as

$$J(\mathbf{q}) = \frac{1}{2} \sum_{i=1}^k a_i |\mathbf{b}_i - A(\mathbf{q})\mathbf{r}_i|^2, \quad (6.18)$$

where the goal is to find the quaternion  $\mathbf{q}$  that minimizes  $J$ . This is identical to finding the quaternion that maximizes  $g(\mathbf{q})$  in

$$g(\mathbf{q}) = 1 - \frac{J(\mathbf{q})}{\frac{1}{2} \sum_{i=1}^k a_i}. \quad (6.19)$$

Keat showed that  $g(\mathbf{q})$  can also be written as<sup>[76]</sup>

$$g(\mathbf{q}) = \mathbf{q}^T K \mathbf{q} \quad (6.20)$$

The goal is to find an attitude matrix,  $K$ , that transforms each unit vector from one image to the next,<sup>[75]</sup> or from one image to that stored by a star catalog.

Equation (6.20) is solved by first weighing each stellar unit vector by its position accuracy, which was determined by the centroider; the smaller the position uncertainty, the higher the weighting. This weighting was found by knowing the SNR of each individual star, and was computed via (4.2). This matrix has been defined as

$$K = \begin{bmatrix} S - \sigma I & z \\ z^T & \sigma \end{bmatrix} \quad (6.21)$$

with

$$S = \left[ \frac{1}{m_k} \sum_{i=1}^k a_i b_i r_i^T \right] + \left[ \frac{1}{m_k} \sum_{i=1}^k a_i b_i r_i^T \right]^T, \quad (6.22)$$

$$z = \frac{1}{m_k} \sum_{i=1}^k a_i (b_i \times r_i), \quad (6.23)$$

and

$$\sigma = \frac{1}{m_k} \sum_{i=1}^k a_i b_i^T r_i \quad (6.24)$$

where  $a_i$  is the weighting based on centroid uncertainty and  $m_k$  is the sum of the uncertainties. To uniquely determine spacecraft attitude, a minimum of three parameters are required. With a quaternion, which contains four parameters, the constraint

$$q^T q = 1 \quad (6.25)$$

must also be satisfied. This constraint can be accounted for in the minimization problem by incorporating it as a Lagrange multiplier. Thus the new maximization function that incorporates the constraint is

$$g'(q) = q^T K q - \lambda q^T q. \quad [16] \quad (6.26)$$

Following differentiation,

$$K q = \lambda q \quad (6.27)$$

results, which is an eigenvalue problem with

$$g(q) = q^T K q = q^T \lambda q = \lambda q^T q = \lambda. \quad (6.28)$$

The eigenvalue that maximizes the gain function described in (6.19) is the largest eigenvalue of  $\mathbf{K}$ ; the eigenvector corresponding to this eigenvalue is the least-squares optimal estimate of the attitude. In other words,

$$g = \lambda_{opt}. \quad (6.29)$$

The q-method requires directly solving the eigenvalue/eigenvector problem, which is generally computationally demanding. QUEST was developed to deal with this issue, and uses an approximation to evaluate the largest eigenvalue. QUEST then calculates the eigenvector corresponding to this approximate eigenvalue.

Rearranging (6.19) and (6.29),

$$\lambda_{opt} = m_k - J \quad (6.30)$$

can be derived. Since the optimal eigenvalue should correspond to a small loss function, the optimal eigenvalue can be approximated by

$$\lambda_{opt} \approx m_k, \quad (6.31)$$

which is the heart of the QUEST algorithm.

Once the optimal eigenvalue is found, the corresponding optimal quaternion can be evaluated directly, without computation of the corresponding eigenvector of  $\mathbf{K}$ , using Rodriguez parameters,<sup>[77]</sup> which gives

$$\mathbf{q}_{opt} = \frac{1}{\sqrt{1 + |\mathbf{y}_{opt}|^2}} \begin{bmatrix} \mathbf{y}_{opt} \\ 1 \end{bmatrix} \quad (6.32)$$

with

$$\mathbf{y}_{opt} = [(\lambda_{opt} + \sigma)\mathbf{I} - \mathbf{S}]^{-1} \mathbf{z}. \quad (6.33)$$

A covariance matrix can also be calculated for the QUEST algorithm to quantitatively define estimation errors. The covariance matrix,  $P_{QQ}$ , was calculated to determine the maximum uncertainty associated with the computed attitude quaternions. The covariance matrix was found via

$$P_{QQ} = \frac{1}{4} \sigma_{TOT}^2 \left[ I - \sum_{i=1}^n a_i \hat{w}_i \hat{w}_i^T \right]^{-1} \quad (6.34)$$

with

$$a_i = \sigma_{TOT}^2 / \sigma_i^2 \quad (6.35)$$

and

$$\left( \sigma_{TOT}^2 \right)^{-1} = \sum_{i=1}^n \left( \sigma_i^2 \right)^{-1} \quad (6.36)$$

where  $\sigma_{TOT}$  is the sum of the individual centroid uncertainties,  $\sigma_i$  is the centroid uncertainty of the  $i^{\text{th}}$  star, and  $w_i$  is the unit vector defining the  $i^{\text{th}}$  centroid.<sup>[75]</sup>

The traditional QUEST algorithm transforms the unit vectors from a reference frame (e.g. that of the star catalog) into a body frame (e.g. that of the current image). However, the same method can be applied to transform unit vectors from one body frame to another (e.g. from one image to the next). Therefore, QUEST can be used to find attitude quaternions that describe relative spacecraft motion or absolute spacecraft motion under LIS conditions.

The QUEST algorithm has been selected as the method for determining the attitude quaternion for the LIST and FAR-MST star trackers because of its computational efficiency and relatively high accuracy. Performance characteristics using this method are discussed in Chapter 7.

### 6.1.4 The Singular Value Decomposition Algorithm

Markley's Singular Value Decomposition (SVD) algorithm is one of the most robust numerical algorithms for minimizing Wahba's loss function because it requires no approximates, remains numerically stable for rank two and three matrices, and has the advantage over the q-method and QUEST in that it provides the eigenvalues and eigenvectors of the covariance matrix.<sup>[78]</sup> The covariance of the attitude estimate is useful in understanding attitude uncertainty because it is a statistical measure of the estimation errors in the reference and body vectors. Unlike the q-method and QUEST, it solves directly for the attitude matrix  $A$ , rather than a quaternion representing  $A$ .

Equation (6.18) can be transformed via matrix manipulations to be equivalent to

$$1 - \sum_{i=1}^k a_i \mathbf{b}_i^T A(\mathbf{q}) \mathbf{r}_i = 1 - \text{tr}(A(\mathbf{q}) B^T) \quad (6.37)$$

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

$$B \equiv \sum_{i=1}^k a_i \mathbf{b}_i \mathbf{r}_i^T . \quad (6.38)$$

In the SVD algorithm, the solution to Wahba's problem is determined via singular value decomposition of the matrix  $B$ , which is given by

$$B = USV^T \quad (6.39)$$

where  $U$  and  $V$  are orthogonal matrices and

$$S = \text{diag}(s_1, s_2, s_3) \quad (6.40)$$

with

$$s_1 \geq s_2 \geq s_3 \geq 0 . \quad (6.41)$$

The matrix  $B$  is strategically redefined as



$$B = U_+ S' V_+^T \quad (6.42)$$

where  $V_+^T$  and  $U_+$  are transformation matrices that transform the attitude matrix from the reference frame to an intermediate frame and from the intermediate  $S$  frame to the body frame, respectively and are defined as

$$V_+ \equiv V \left[ \text{diag}(1, 1, \det(V)) \right], \quad (6.43)$$

and

$$U_+ \equiv U \left[ \text{diag}(1, 1, \det(U)) \right], \quad (6.44)$$

and

$$S' \equiv \text{diag}(s_1, s_2, ds_3) \quad (6.45)$$

where

$$d \equiv (\det(U))(\det(V)) = \pm 1. \quad (6.46)$$

Additionally, define

$$W \equiv U_+^T S' V_+^T = \cos(\Phi)I + 1 - \cos(\Phi)\mathbf{e}\mathbf{e}^T - \sin(\Phi)[\mathbf{e}\times] \quad (6.47)$$

where  $\mathbf{e}$  is the unit vector Euler axis representation of  $W$  and  $\Phi$  is the Euler rotation angle representation of  $W$ .

Substituting (6.42) into (6.37) and using cyclic invariance of the trace and (6.47),

$$L(A) = 1 - \text{tr}(S'W) = 1 - \text{tr}(S') + (1 - \cos(\Phi)) \left[ s_2 + ds_3 + (s_1 - s_2)e_2^2 + (s_1 - ds_3)e_3^2 \right]. \quad (6.48)$$

Because of (6.41),  $L(A)$  is minimized when  $\Phi = 0$ , so the optimal  $A$  is given by

$$L(A_{opt}) = 1 - \text{tr}(S') = 1 - s_1 - s_2 - ds_3 \quad (6.49)$$

which means that

$$A_{opt} = U_+ V_+^T = U [diag(1, 1, d)] V^T. \quad (6.50)$$

Once the optimal  $A$  matrix that minimizes the least-squares loss function is found, the covariance matrix  $P$  can be calculated. Let  $D$  be defined as

$$D \equiv diag(s_2 + ds_3, ds_3 + s_1, s_1 + s_2) = [1 - L(A_{opt})] I - S'. \quad (6.51)$$

Then the covariance matrix can be derived to be

$$P = \sigma_{tot}^2 (I - S') D^{-2} \quad (6.52)$$

where

$$\sigma_{tot}^2 = \left[ \sum_{i=1}^k (\sigma_{bi}^2 + \sigma_{ri}^2) \right]^{-1} \quad (6.53)$$

and  $\sigma_{ri}^2$  and  $\sigma_{bi}^2$  are the variances of the reference and observation errors, respectively.

### 6.1.5 The Fast Optimal Attitude Matrix Algorithm

Like SVD, the Fast Optimal Attitude Matrix (FOAM) algorithm solves for the optimal attitude matrix directly. The purpose of FOAM is to present a more efficient method to estimate the attitude matrix than that proposed by the SVD algorithm.<sup>[68]</sup> This is accomplished by rewriting  $A_{opt}$  in terms of properties of the  $B$  matrix and three scalar components

$$A_{opt} = \frac{\left[ (\kappa + \|B\|^2) B + \lambda adj(B^T) - BB^T B \right]}{\zeta} \quad (6.54)$$

where

$$\kappa \equiv S_2 S_3 + S_3 S_1 + S_1 S_2, \quad (6.55)$$

$$\lambda \equiv S_1 + S_2 + S_3, \quad (6.56)$$

and

$$\zeta \equiv (s_2 + s_3)(s_3 + s_1)(s_1 + s_2). \quad (6.57)$$

The advantage of (6.54) over (6.50) is that the scalar coefficients  $\kappa$ ,  $\lambda$ , and  $\zeta$  can be computed either iteratively or analytically, without the requirement of using SVD.

Similarly, the covariance matrix can be rewritten in terms of these same coefficients. Markley showed that (6.52) could be more easily written as

$$P = \frac{\lambda \sigma_{tot}^2 (\kappa I + BB^T)}{\zeta} \quad (6.58)$$

where  $\lambda > 0$ . Again, through the use of the scalar coefficients, this equation eliminates the need to use SVD.



# Chapter 7

## ALGORITHM PERFORMANCE

There are numerous approaches to test the algorithms discussed in Chapters 4 through 6. Simulated images, nighttime sky photographs, and images generated by the FAR-MST prototypes have all been used to obtain performance characteristics for the centroiders, pattern and rate matching algorithms, and attitude determining functions. Images simulated in Matlab have been created by using the parameters of the optics and imager selected for LIST and FAR-MST, whose quantities are listed in Table 7.1. These images hold the PSF width constant across the entire imager pixel matrix, which implicitly assumes the implementation of a double lens.

**Table 7.1 Parameter Settings for Centroider Performance Tests**

Parameter	Value
hPixels	1280
vPixels	1024
hFOV (°)	30
vFOV (°)	24.25
pp <sub>x</sub> (mm)	6.70E-03
pp <sub>y</sub> (mm)	6.70E-03
f (mm)	15.97
gain (electron count/ADU)	10
FFSR (A/W)	0.12
t <sub>shutter</sub> (s)	0.1
max M <sub>v</sub>	6
DK (# of photoelectrons)	410
RO (# of photoelectrons)	39
lens area (cm <sup>2</sup> )	1.77
PSFwidth (pixels)	10
noise threshold range (# of $\sigma$ )	1 to 4

Two different types of simulated images have been created: one that generates its own stars and one that pulls stars from various regions in the sky, as defined in the NASA Sky2000 star catalog. Images that generate their own stars, randomly place the stars of varying M<sub>v</sub> on a

---

simulated pixel matrix to sub-pixel resolution and individually render them to project Gaussian images on the pixel matrix; the appropriate amounts of dark current, readout, and stellar shot noise are then added to the pixel matrix. These images are generally used to test algorithm robustness, as they can be easily modified to measure the effects of changing certain parameters, such as stellar intensity, noise threshold, lens area, or PSF width.

Simulated images that use actual stars from the NASA SKY2000 star catalog are used when a sequence of images is desired. These images are generated by defining a maneuver profile that is based on a desired location in the sky, a particular FOV, and desired direction of movement. Stars are then collected from the star catalog and placed onto a layered pixel matrix, where each layer represents an image snap-shot of the sky. The appropriate dark current, readout, and stellar shot noise are also added to these images.

A sequence of fourteen images based on the NASA SKY2000 star catalog have been generated to examine various performance characteristics of the two weighted sum centroiding algorithms, as well as the triangle match, rate match, n-vertex match, and QUEST algorithms. The images were created by selecting a known region in the sky and mapping the stars in that region onto a pixel matrix. Since a specific maneuver file rotated the images through a series of pre-defined motions, each image had an exact set of centroid locations for all of the stars on that image and a specified attitude quaternion defining the pointing direction of the image in the sky. Stellar shot, readout, and dark current noise were then added to the images. The fourteen images were run through two different weighted sum centroiders, as well as the triangle match, rate match, n-vertex match, and QUEST algorithms to determine approximated centroid positions of all the stars in each image, the attitude quaternion that location of each image in the sky, and the relative attitude quaternion that defines the motion between consecutive images. Studies of algorithm performance included comparisons of execution times between the centroiding algorithms, differences between the actual and computed attitude quaternions defining the direction of pointing in the sky, differences between the actual and computed quaternions defining the movement of the spacecraft between images, and the amount of error propagation over time resulting from these motions.

## 7.1 Centroider Performance

Two centroiding algorithms, each implementing a weighted sum technique, have been created to centroid the images. Both centroiders determine the centroid x-y position on the image plane, stellar brightness, and aspect ratio for each star in the image. The difference between starCentroid2 and starCentroid3 is the type of filtering used to eliminate likely noise spikes, as discussed in Chapter 4. This trade study has been completed to determine which centroider performs best under various noise conditions, where the “best” centroider is the one that centroids the stars the most accurately and in the shortest time.

### 7.1.1 Centroider Speed

Figure 7.1 through Figure 7.5 compare the differences in execution times for starCentroid2 and starCentroid3 at five different noise threshold values. Again, these speeds were calculated in the Matlab environment, which is slower than the C environment that will be used onboard LIST and FAR-MST; the relative centroiding times are what is important rather than the magnitude of the times.

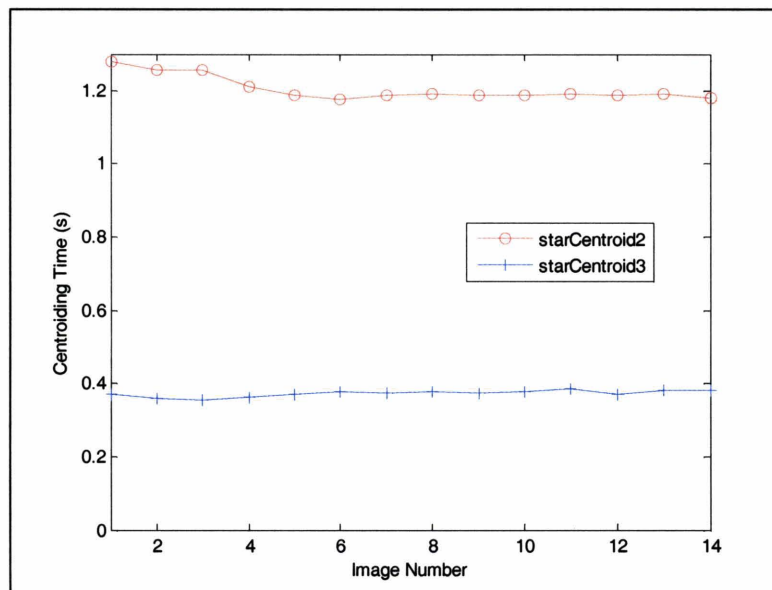


Figure 7.1 Centroiding Execution Times for Noise Threshold of  $1\sigma_{\text{noise}}$

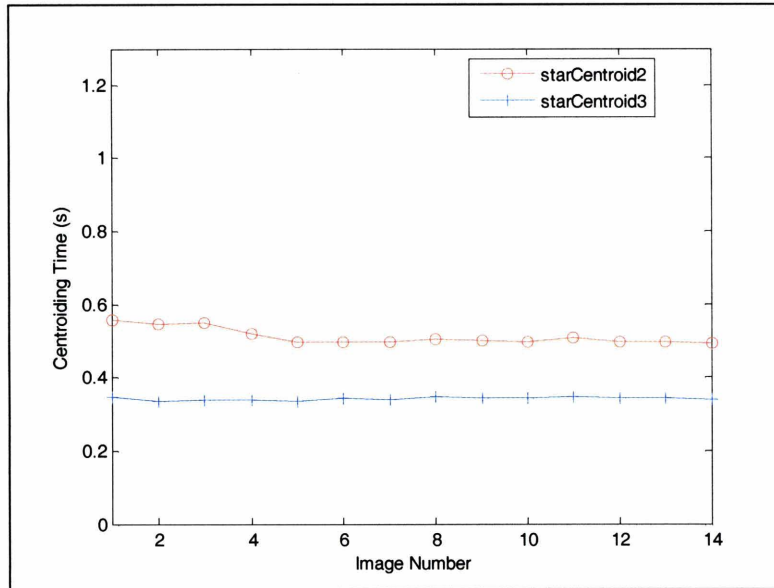


Figure 7.2 Centroiding Execution Times for Noise Threshold of  $1.5\sigma_{\text{noise}}$

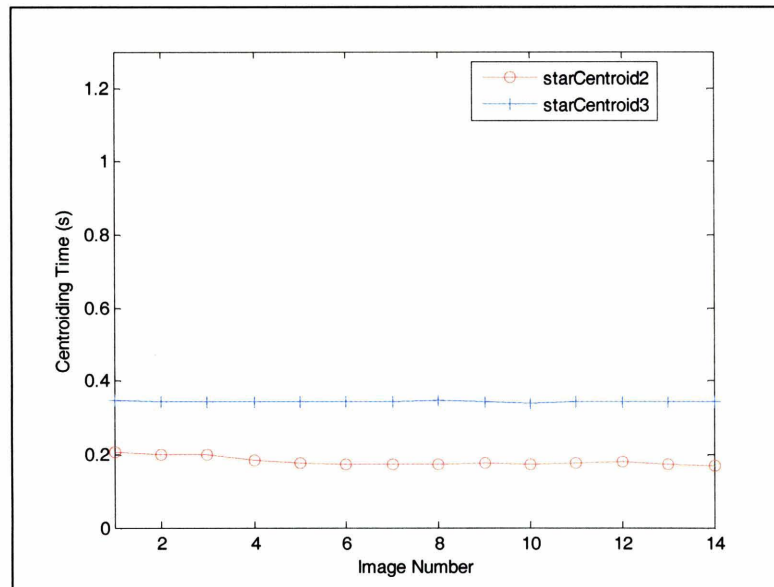
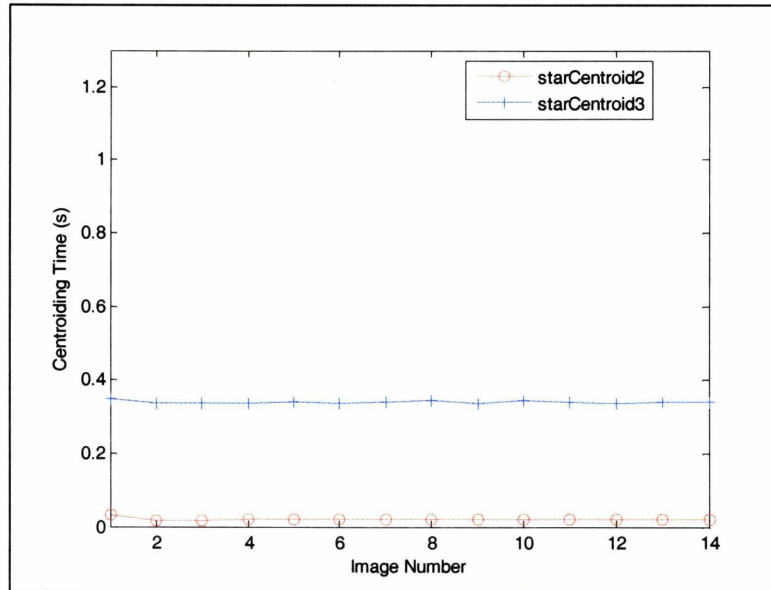
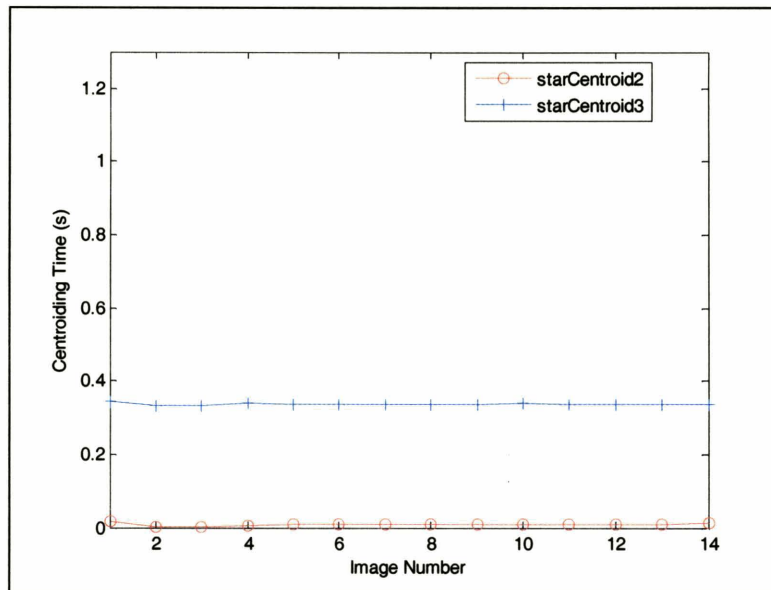


Figure 7.3 Centroiding Execution Times for Noise Threshold of  $2\sigma_{\text{noise}}$





**Figure 7.4 Centroiding Execution Times for Noise Threshold of  $3\sigma_{\text{noise}}$**



**Figure 7.5 Centroiding Execution Times for Noise Threshold of  $4\sigma_{\text{noise}}$**

Table 7.2 displays some of the significant statistics for this trade study that help to explain the results shown in the figures above. The highlighted entries show the centroider that performed the fastest. The “stars centroided” column shows the number of stars accurately centroided by the respective centroiders; the range is due to the fact that each entry in this column represents all fourteen images, where each of the fourteen images contains different numbers of stars. Overall centroider times were split into two components: time to pre-filter and

time to centroid. For each noise threshold level, the average (mean) and standard deviation (std) of the pre-filter and centroid times calculated over the fourteen images is given.

**Table 7.2 Weighted Sum Centroider Comparison Statistics**

noise threshold (# $\sigma$ )	centroider	stars centroided (#)	time to pre-filter (s)		time to centroid (s)	
			mean	std	mean	std
1	starCentroid2	10 to 17	1.1860	0.0363	0.0188	0.0053
	starCentroid3	8 to 15	0.3339	0.0015	0.0393	0.0084
1.5	starCentroid2	8 to 15	0.4976	0.0254	0.0135	0.0041
	starCentroid3	4 to 11	0.3215	0.0016	0.0214	0.0045
2	starCentroid2	5 to 10	0.1698	0.0132	0.0109	0.0036
	starCentroid3	3 to 8	0.3216	0.0027	0.0227	0.0028
3	starCentroid2	3 to 8	0.0132	0.0013	0.0085	0.0032
	starCentroid3	2 to 5	0.3192	0.0024	0.0204	0.0029
4	starCentroid2	2 to 6	0.0023	0.0008	0.0071	0.0030
	starCentroid3	1 to 3	0.3180	0.0013	0.0198	0.0036

In both centroiders, the actual centroiding methods are identical; the difference in these algorithms is in the pre-filter method. The starCentroid2 and starCentroid3 centroiding times vary because of the results from the pre-filtering. It can be seen in this table that, in every case, the time to pre-filter significantly dominates the time to centroid. It is also seen that the time to pre-filter varies considerably with the noise threshold level for starCentroid2, while the time to pre-filter is somewhat less dependent on the noise threshold value for starCentroid3. This, again, is due to the nature of the pre-filter used in each centroider. In starCentroid2, the filtering looping structure is primarily based on the number of pixels that break the noise threshold, which is dependent on the value of the noise threshold. In starCentroid3, the filtering outer-most looping structure is based on the total number of pixels in the entire image, which is a constant regardless of the level of the noise threshold, and an inner loop is based on the pixels that break the noise threshold. As noise threshold levels increase, the number of pixels that exceed the threshold is reduced. This explains the major decrease in mean pre-filtering time for starCentroid2 for increasing values of the noise threshold. The slight decrease in mean pre-filtering time for starCentroid3 is due to the fact that this filter contains an inner loop that is based on the number of pixels that exceed the threshold; higher thresholds contain fewer pixels to be considered in this loop.

### 7.1.2 Centroider Accuracy

Centroid accuracy was tested by comparing the actual centroids defined by the maneuver profile to the centroids calculated by starCentroid2 and starCentroid3. For noise threshold levels of  $1\sigma_{\text{noise}}$ ,  $1.5\sigma_{\text{noise}}$ ,  $2\sigma_{\text{noise}}$ ,  $3\sigma_{\text{noise}}$ , and  $4\sigma_{\text{noise}}$ , all fourteen images were examined and evaluated. The first five images at the  $1\sigma_{\text{noise}}$  and  $3\sigma_{\text{noise}}$  threshold levels, shown in Figure 7.6 through Figure 7.25, sufficiently represent the general trends seen across all fourteen images at the five different noise threshold levels.

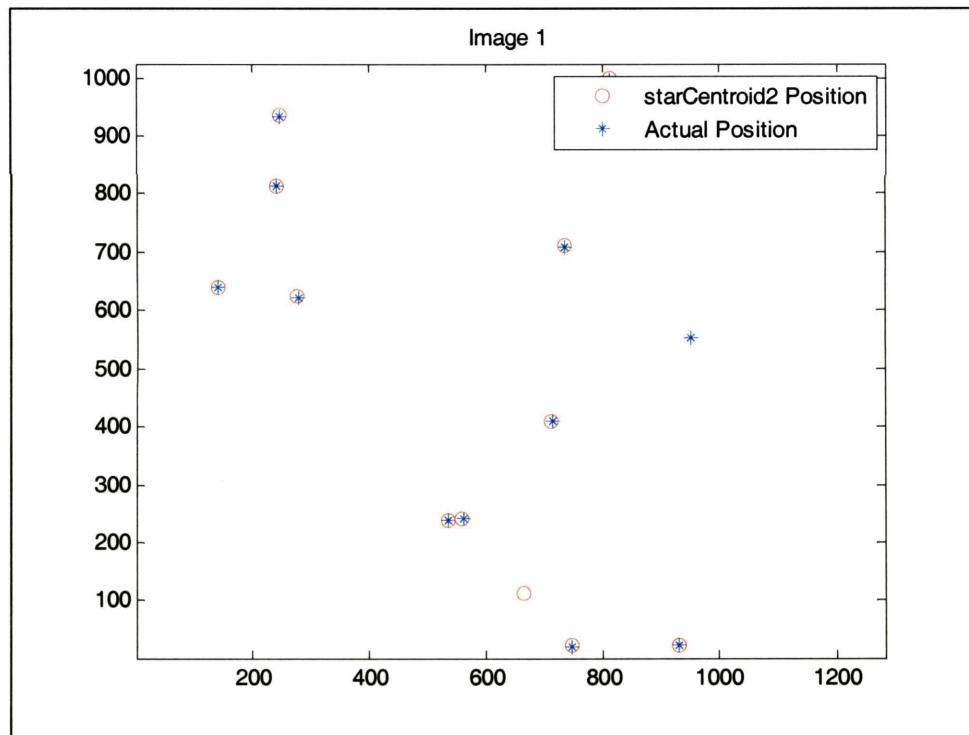


Figure 7.6 Image 1 - starCentroid2 Positioning Accuracy for Noise Threshold of  $1\sigma_{\text{noise}}$

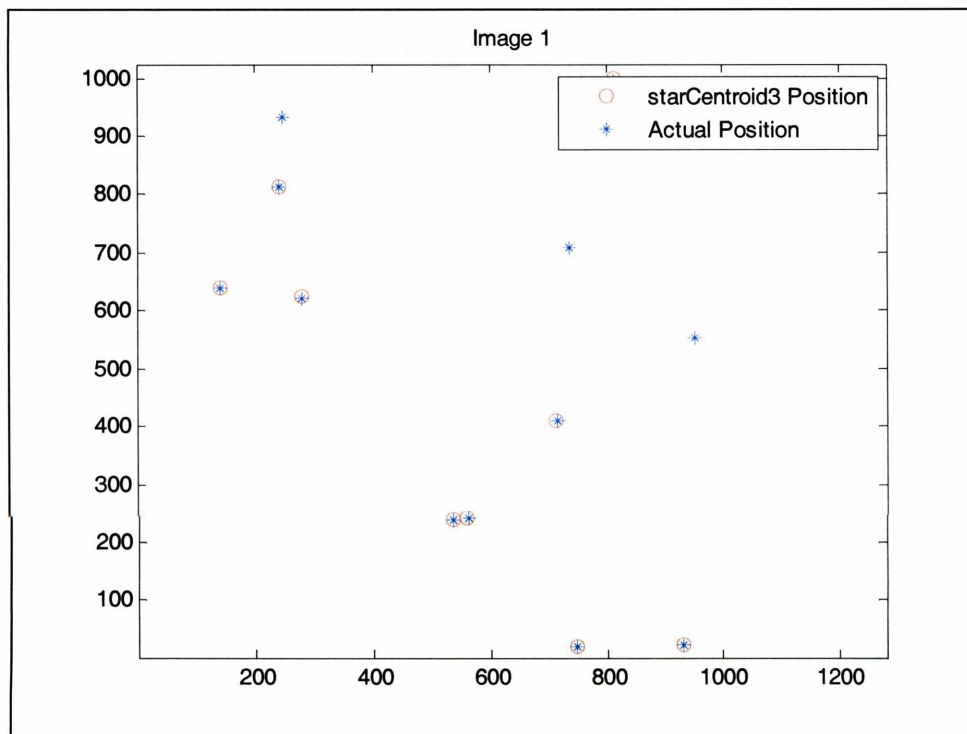


Figure 7.7 Image 1 – starCentroid3 Positioning Accuracy for Noise Threshold of  $1\sigma_{\text{noise}}$

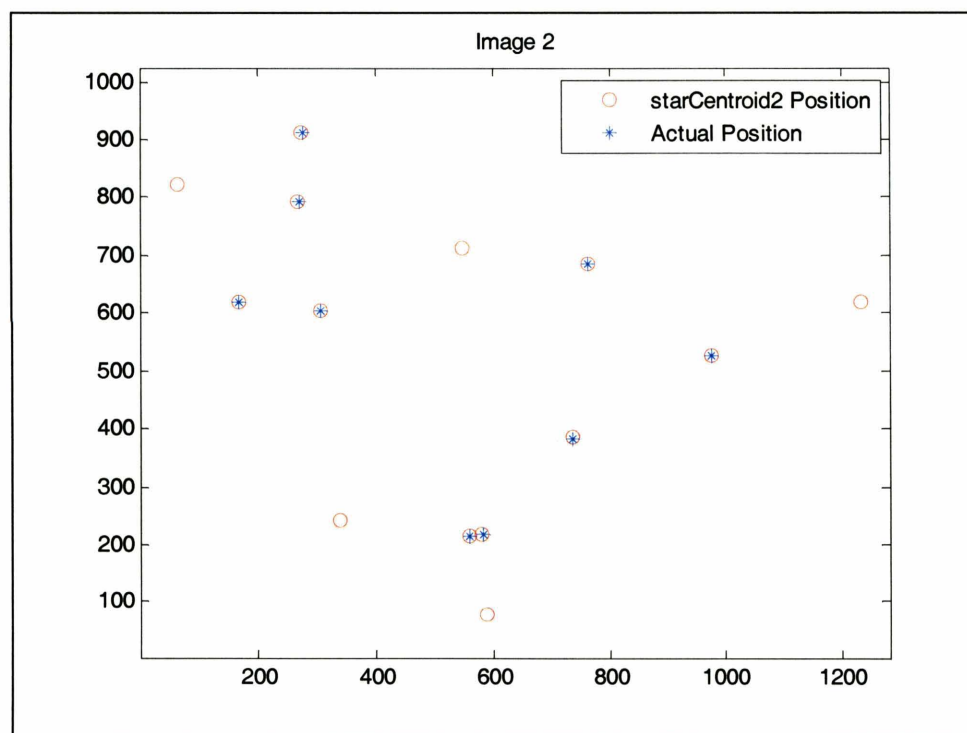


Figure 7.8 Image 2 - starCentroid2 Positioning Accuracy for Noise Threshold of  $1\sigma_{\text{noise}}$

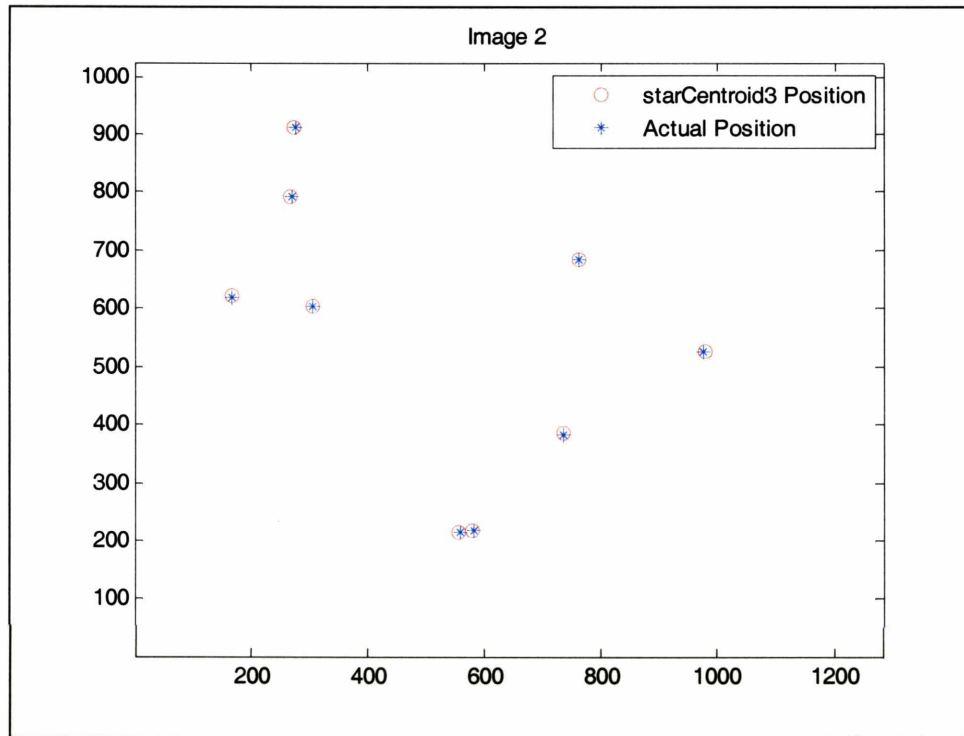


Figure 7.9 Image 2 – starCentroid3 Positioning Accuracy for Noise Threshold of  $1\sigma_{\text{noise}}$

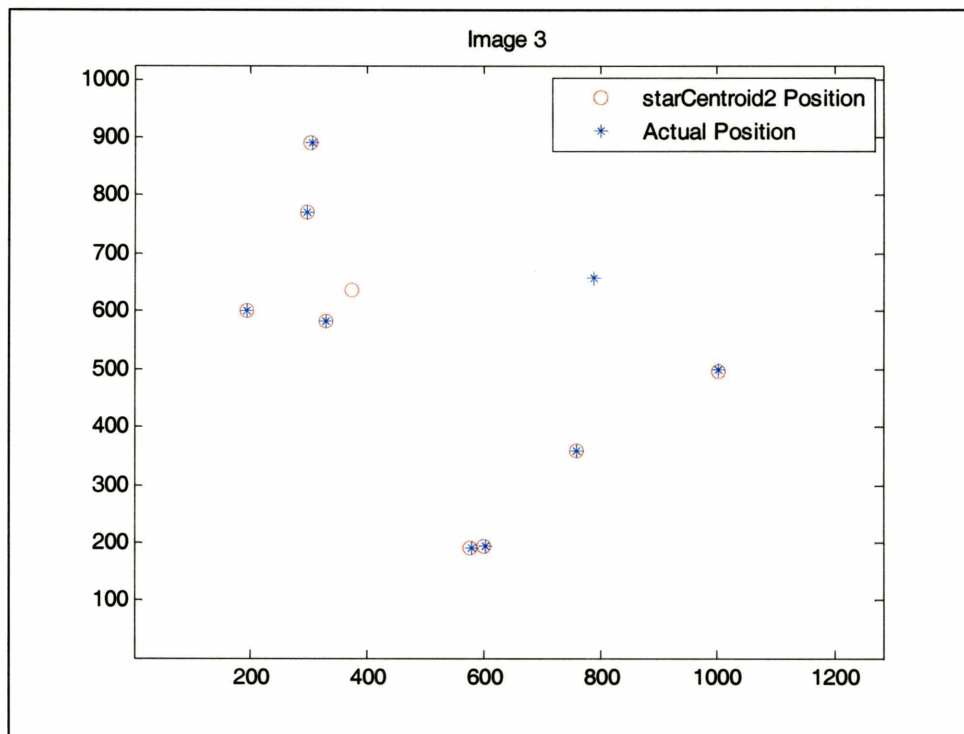


Figure 7.10 Image 3 - starCentroid2 Positioning Accuracy for Noise Threshold of  $1\sigma_{\text{noise}}$

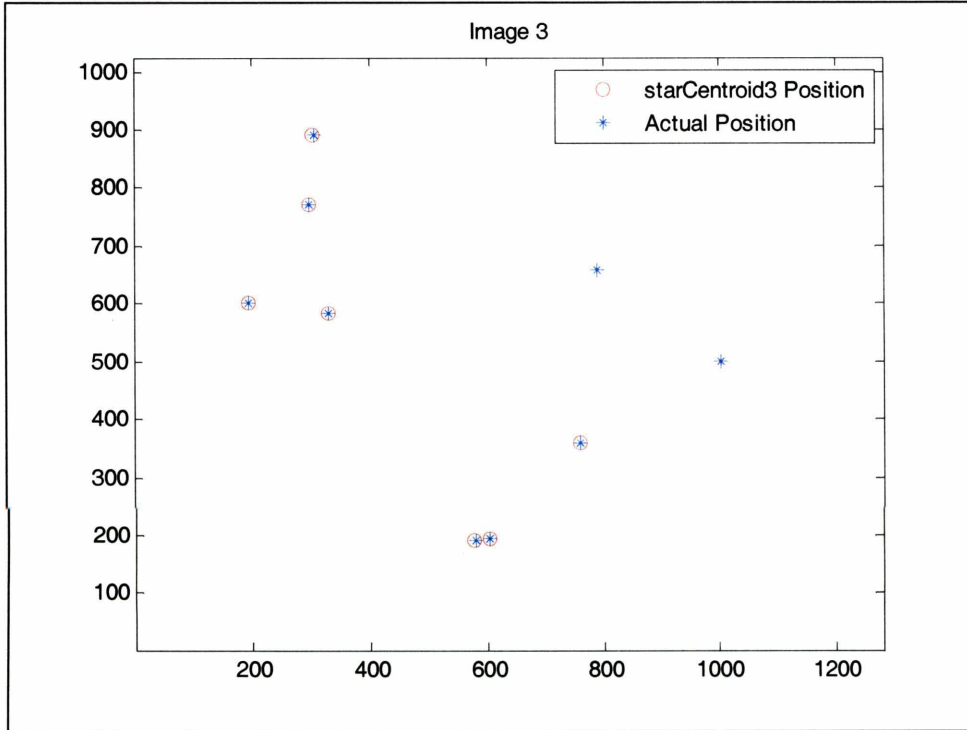


Figure 7.11 Image 3 – starCentroid3 Positioning Accuracy for Noise Threshold of  $1\sigma_{noise}$

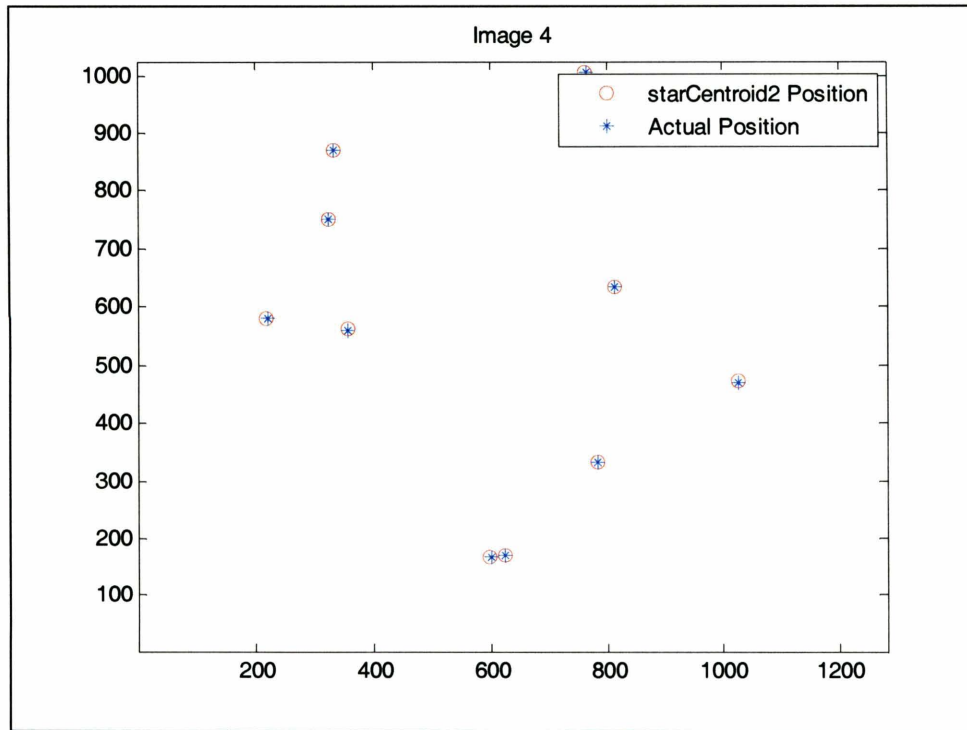


Figure 7.12 Image 4 - starCentroid2 Positioning Accuracy for Noise Threshold of  $1\sigma_{noise}$



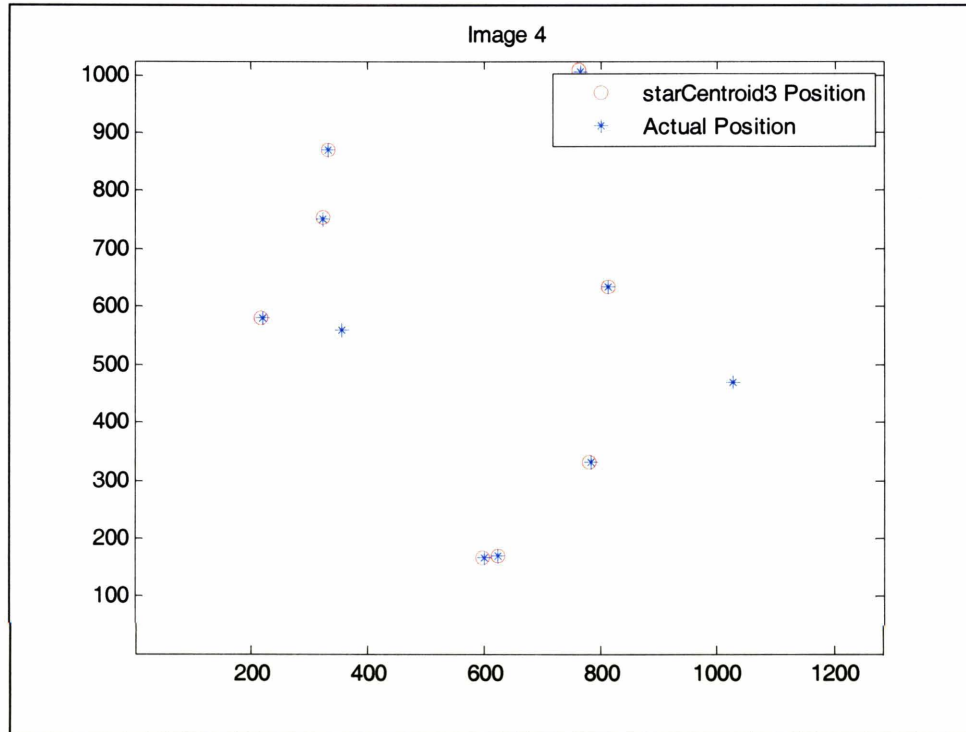


Figure 7.13 Image 4 – starCentroid3 Positioning Accuracy for Noise Threshold of  $1\sigma_{\text{noise}}$

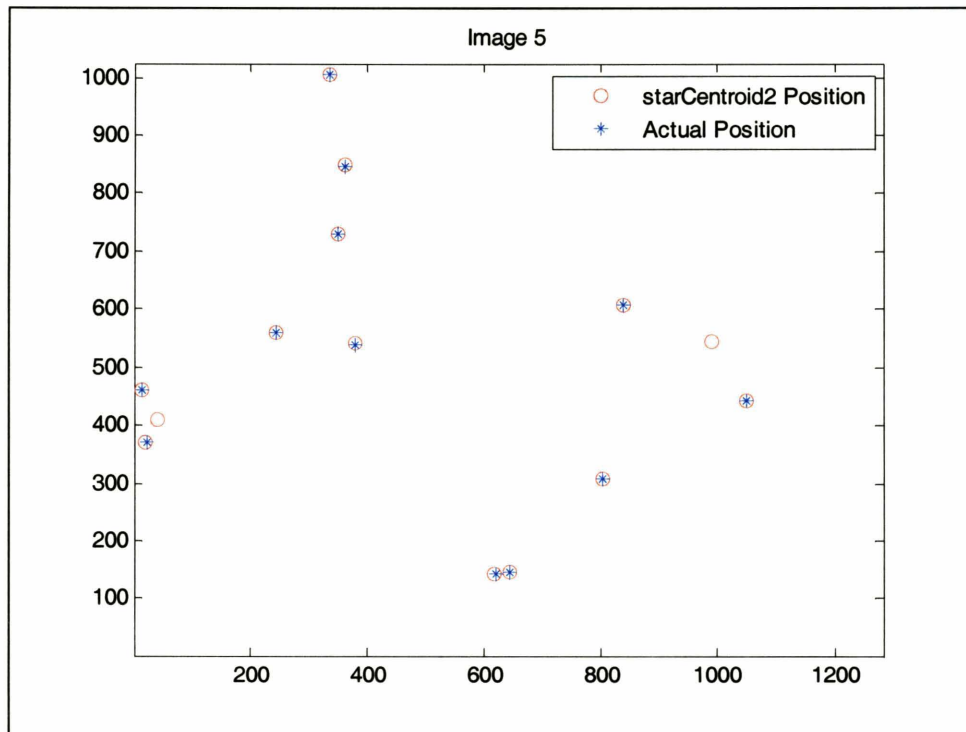
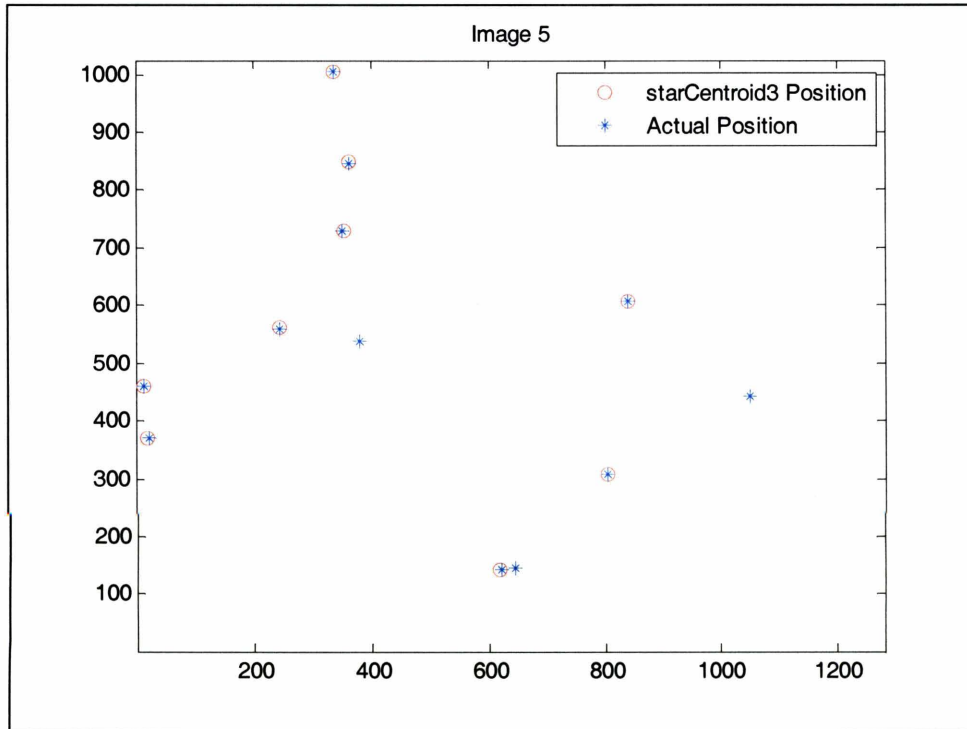


Figure 7.14 Image 5 - starCentroid2 Positioning Accuracy for Noise Threshold of  $1\sigma_{\text{noise}}$



**Figure 7.15 Image 5 – starCentroid3 Positioning Accuracy for Noise Threshold of  $1\sigma_{\text{noise}}$**

Figure 7.6 through Figure 7.15 show that at the  $1\sigma_{\text{noise}}$  threshold level, starCentroid2 centroided more of the actual stars in the image, but had a tendency to also centroid noise spikes. On the other hand, starCentroid3 centroided fewer actual stars but did not centroid noise spikes. The tendency of starCentroid2 to centroid noise spikes is important because in real operating conditions, when no “actual” centroid data is available, the figures above suggest that starCentroid2 will have difficulty distinguishing stars from noise spikes. However, because starCentroid2 can centroid more of the actual stars, starCentroid2 will be helpful in scenarios when a minimal number of stars are available and a centroider is required that can extract as many of these stars as possible to successfully obtain attitude information.



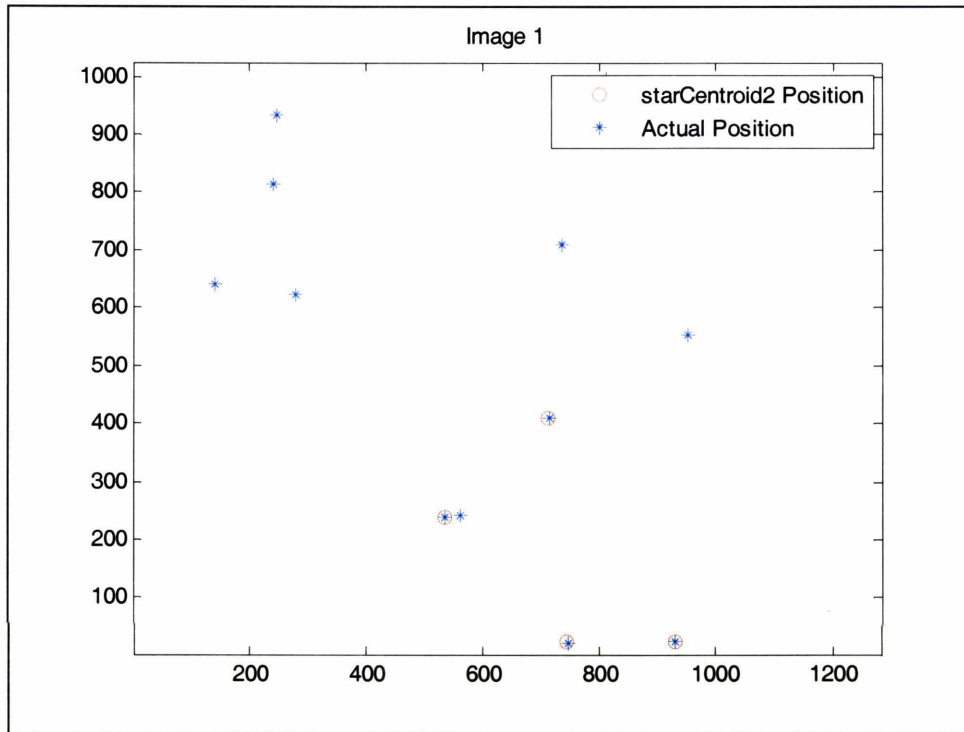


Figure 7.16 Image 1 - starCentroid2 Positioning Accuracy for Noise Threshold of  $3\sigma_{\text{noise}}$

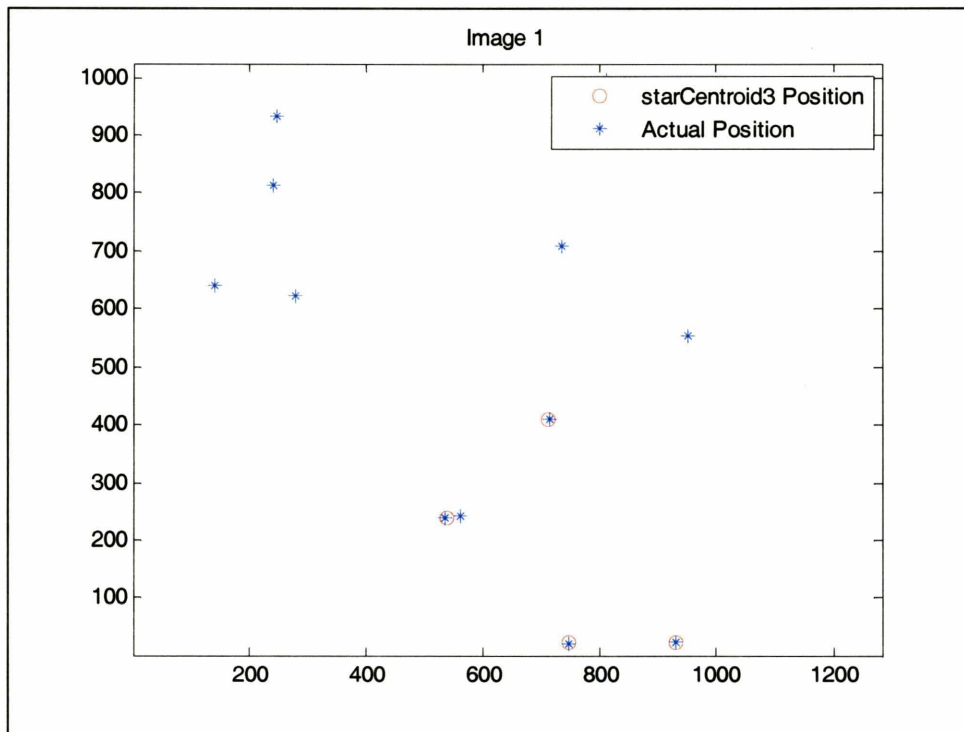


Figure 7.17 Image 1 – starCentroid3 Positioning Accuracy for Noise Threshold of  $3\sigma_{\text{noise}}$

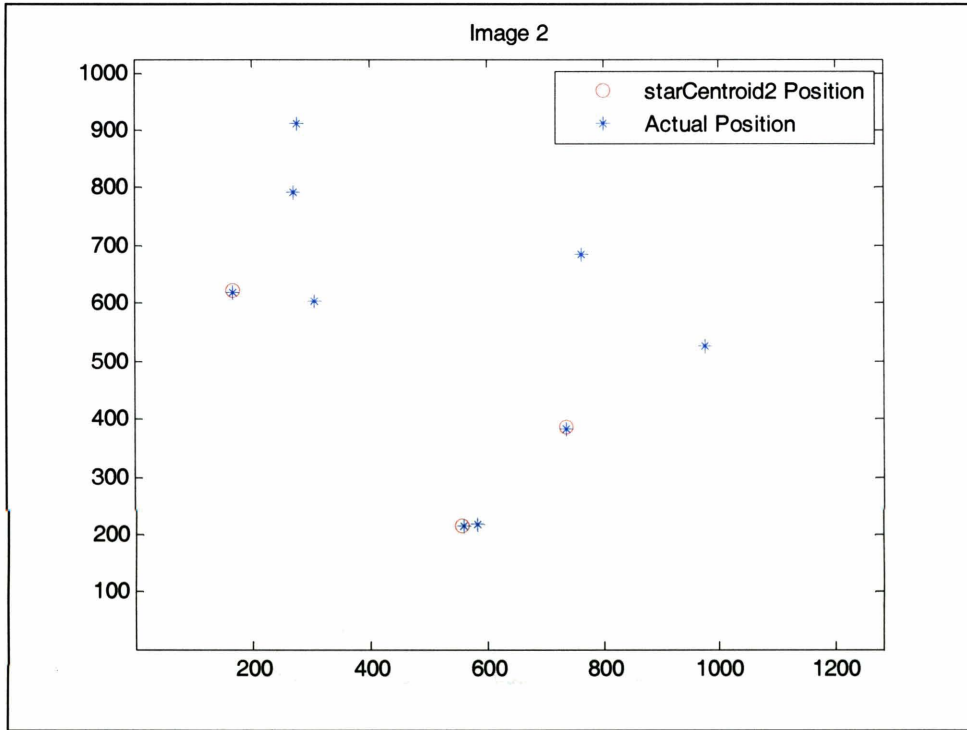


Figure 7.18 Image 2 - starCentroid2 Positioning Accuracy for Noise Threshold of  $3\sigma_{noise}$

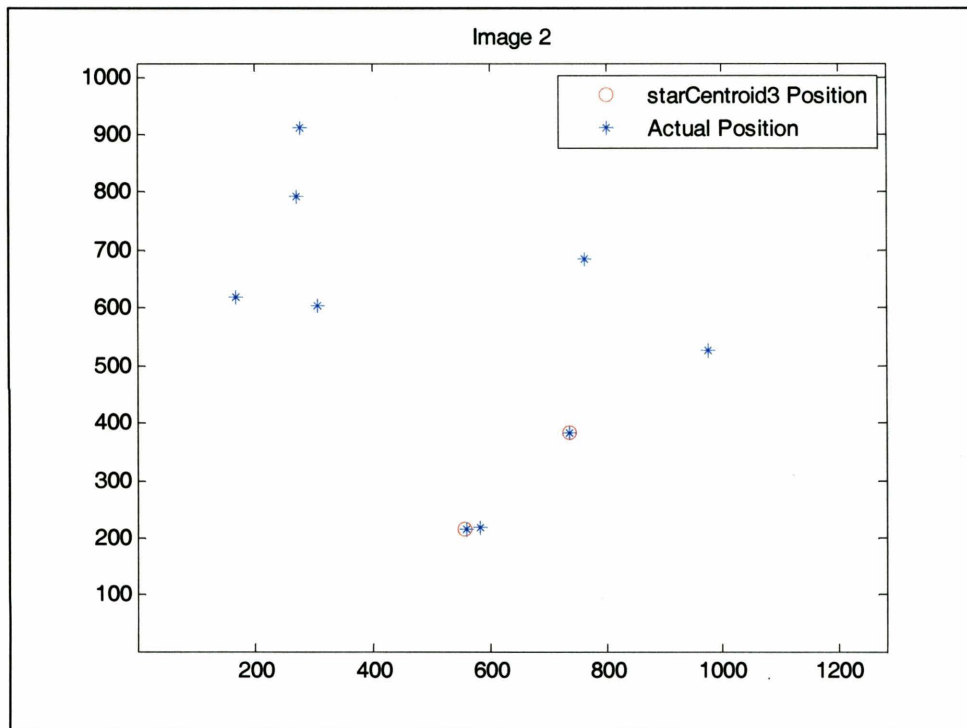


Figure 7.19 Image 2 – starCentroid3 Positioning Accuracy for Noise Threshold of  $3\sigma_{noise}$

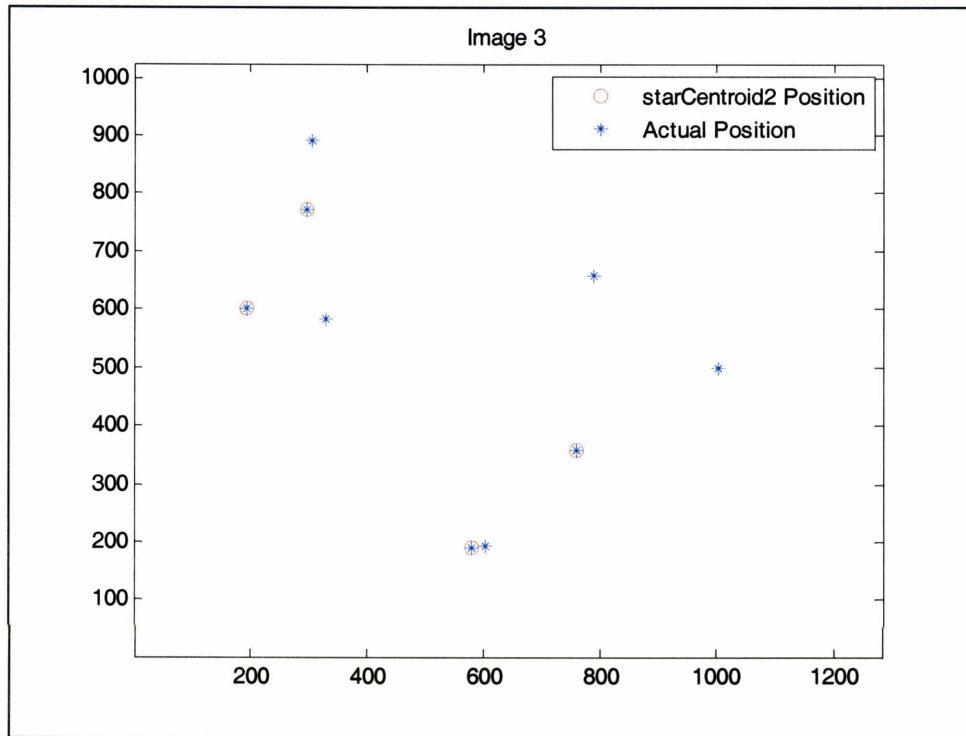


Figure 7.20 Image 3 - starCentroid2 Positioning Accuracy for Noise Threshold of  $3\sigma_{\text{noise}}$

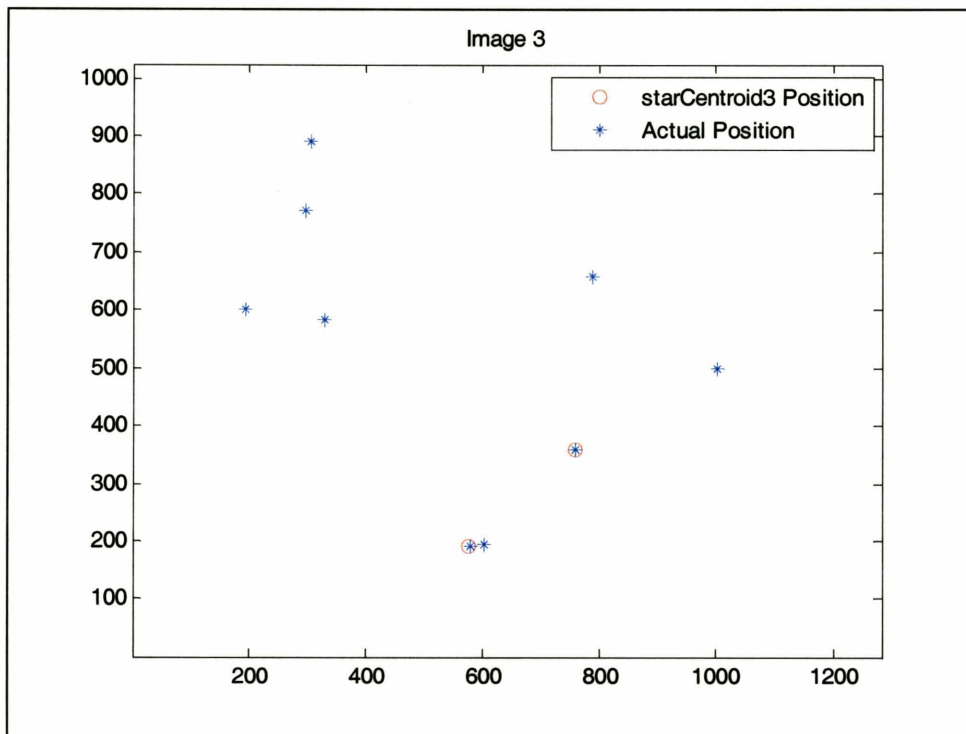


Figure 7.21 Image 3 – starCentroid3 Positioning Accuracy for Noise Threshold of  $3\sigma_{\text{noise}}$

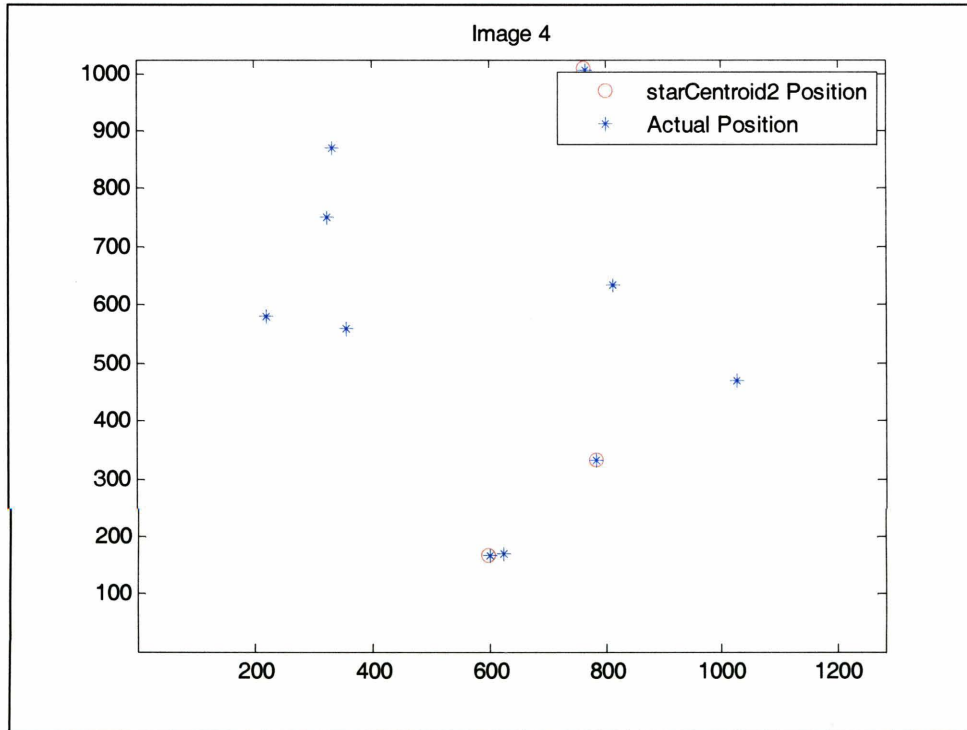


Figure 7.22 Image 4 - starCentroid2 Positioning Accuracy for Noise Threshold of  $3\sigma_{\text{noise}}$

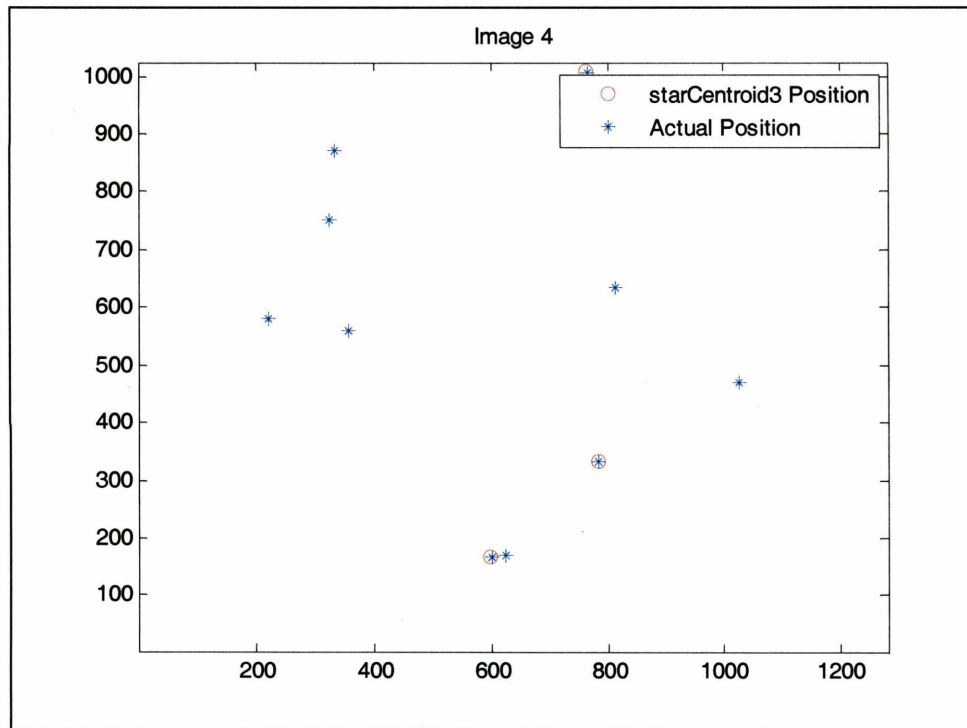


Figure 7.23 Image 4 – starCentroid3 Positioning Accuracy for Noise Threshold of  $3\sigma_{\text{noise}}$

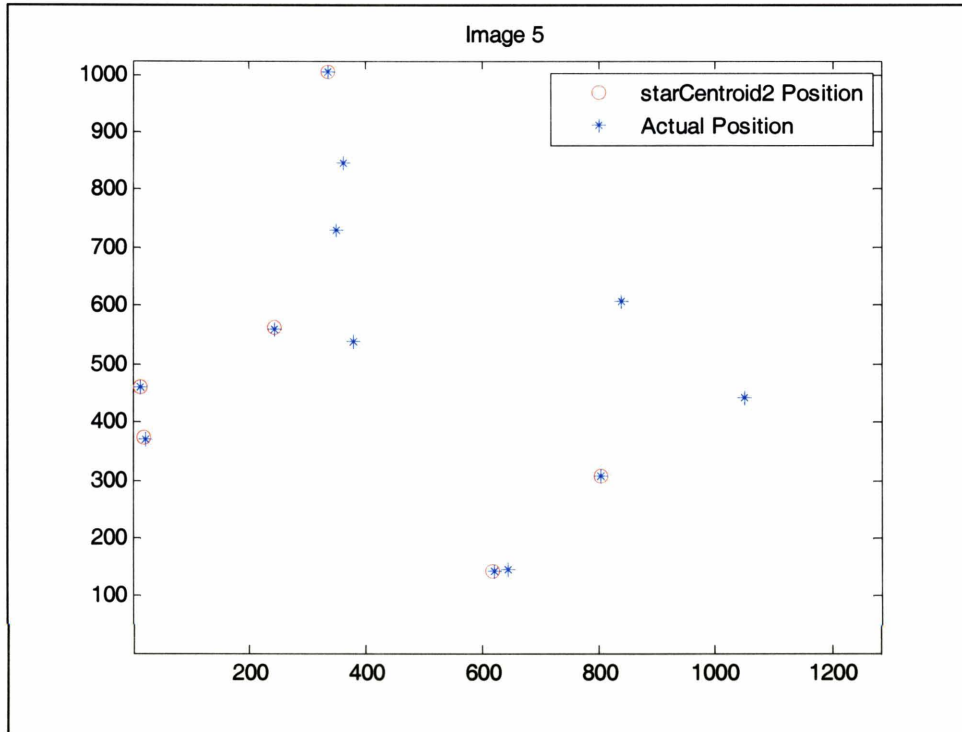


Figure 7.24 Image 5 - starCentroid2 Positioning Accuracy for Noise Threshold of  $3\sigma_{\text{noise}}$

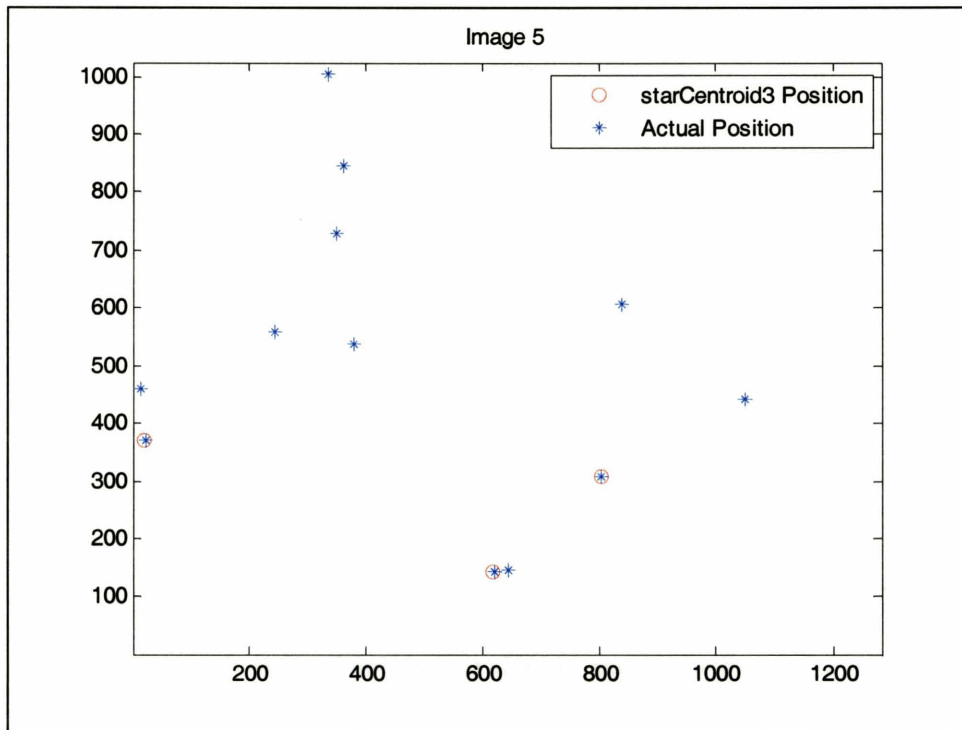


Figure 7.25 Image 5 – starCentroid3 Positioning Accuracy for Noise Threshold of  $3\sigma_{\text{noise}}$

Similar to the cases with the  $1\sigma_{\text{noise}}$  noise threshold, starCentroid2 centroids more stars than starCentroid3 when the noise threshold is set to  $3\sigma_{\text{noise}}$ . The advantage of the higher noise threshold is displayed in these figures because no noise spikes were centroided as stars by either centroider.

In all fourteen images at all five noise threshold levels for both centroiders, all of the stars that were centroided correctly were correct within four pixels. The majority of the stars centroided were correct to within one pixel, but dimmer stars, which were more easily affected by noise and especially present in the second and third images, were centroided less accurately.

### 7.1.3 Attitude Quaternion Uncertainty

The uncertainty measurements for the attitude quaternions representing these fourteen images can be computed based on the centroiding accuracy alone, as described in equations (6.34), (6.35), and (6.36). These uncertainties are displayed in Figure 7.26 through Figure 7.30 for five noise threshold levels between  $1\sigma_{\text{noise}}$  and  $3\sigma_{\text{noise}}$ .

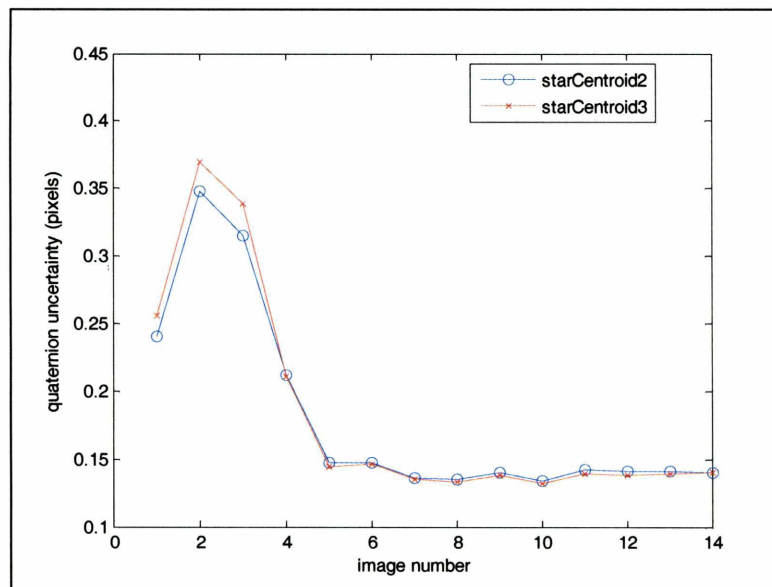


Figure 7.26 Uncertainty in Image Attitude Quaternion for Noise Threshold of  $1\sigma_{\text{noise}}$

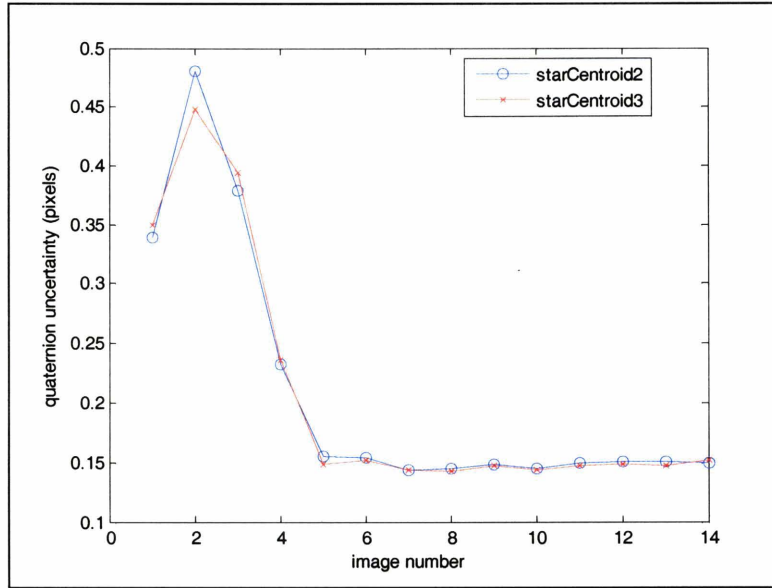


Figure 7.27 Uncertainty in Image Attitude Quaternion for Noise Threshold of  $1.5\sigma_{\text{noise}}$

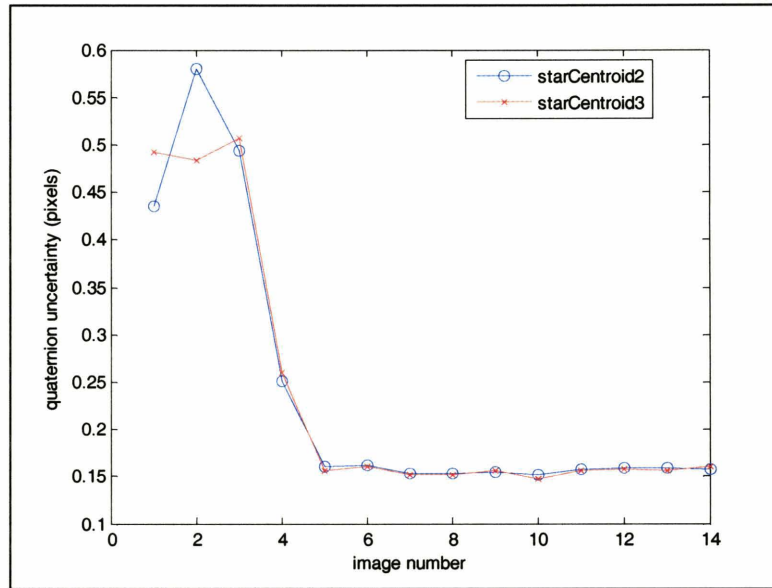
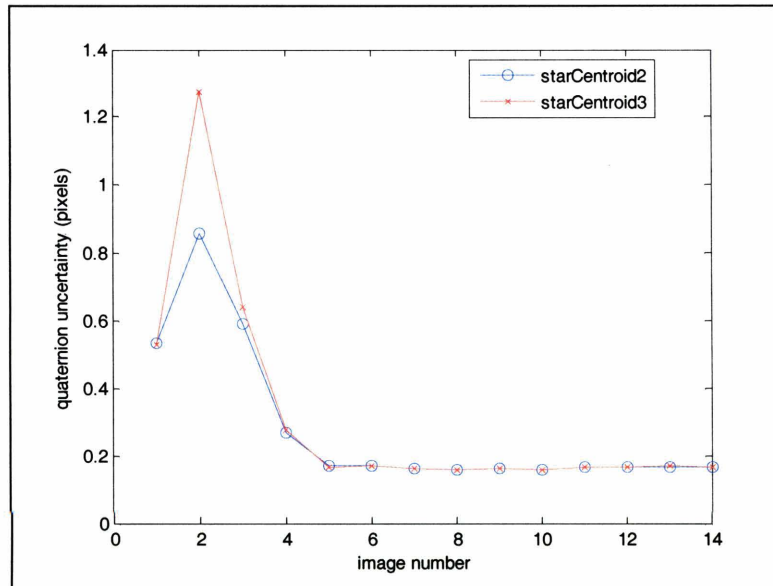
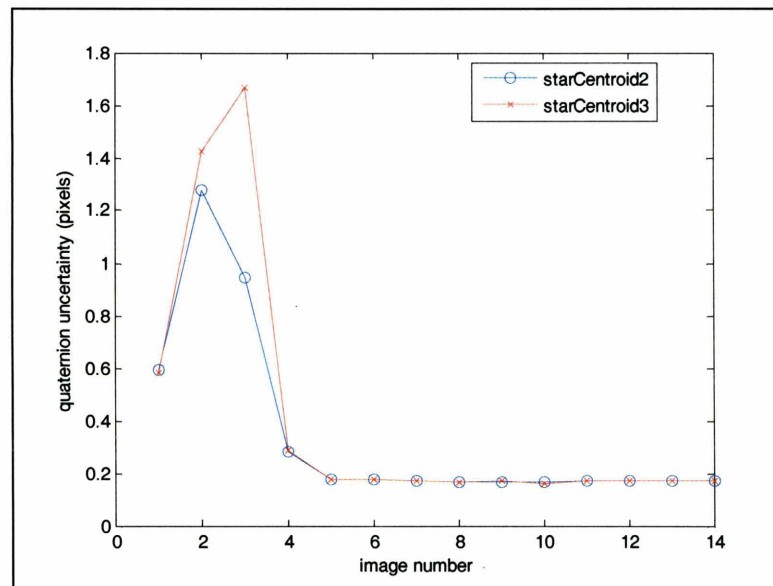


Figure 7.28 Uncertainty in Image Attitude Quaternion for Noise Threshold of  $2\sigma_{\text{noise}}$



**Figure 7.29** Uncertainty in Image Attitude Quaternion for Noise Threshold of  $2.5\sigma_{\text{noise}}$



**Figure 7.30** Uncertainty in Image Attitude Quaternion for Noise Threshold of  $3\sigma_{\text{noise}}$

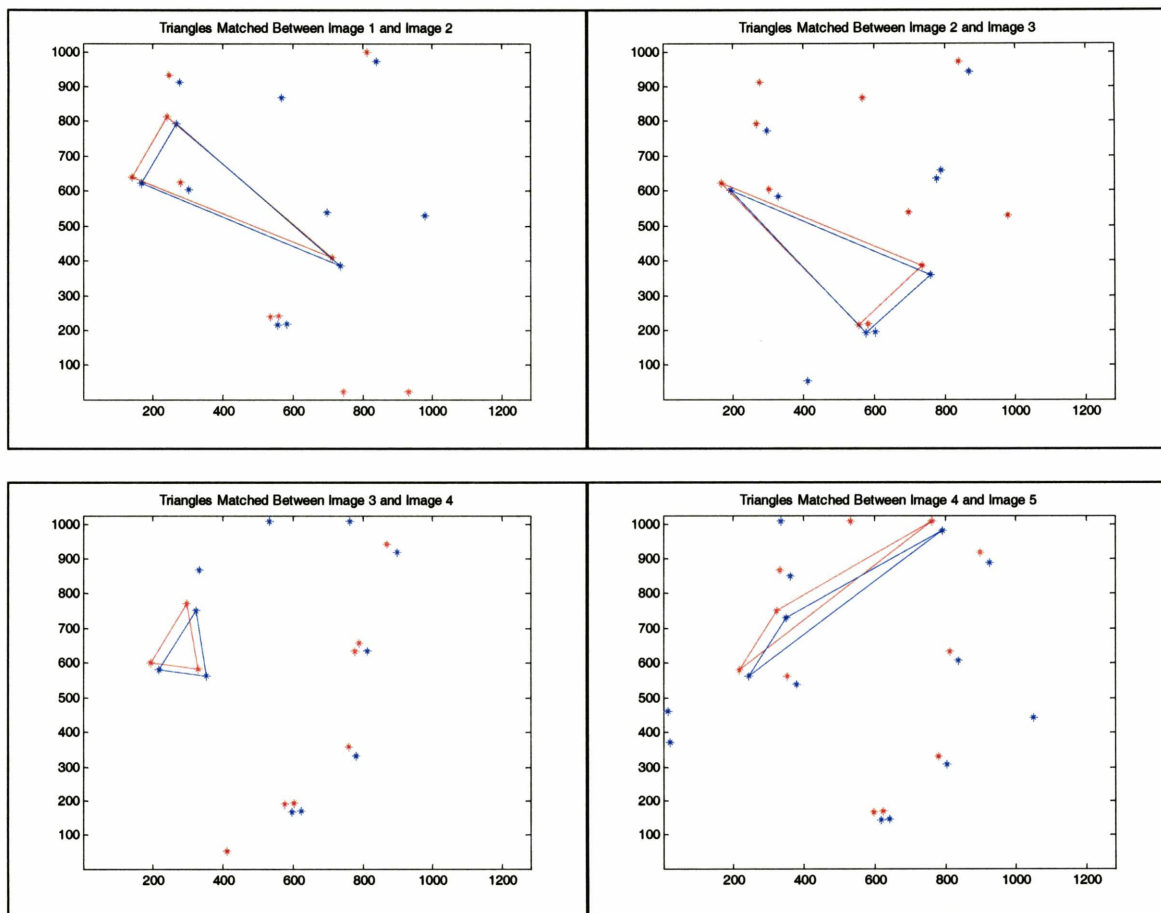
The general pattern of attitude uncertainty remains the same throughout these five figures: attitude uncertainty begins at a moderate level, increases for images 2 and 3, drops rapidly at image 4, and remains fairly constant thereafter. Images 5 through 14 contain two very bright stars, both brighter than a  $M_V$  of 1. Image 4 contains one of these bright stars. Their presence enables high centroid accuracies, which drives down the overall attitude uncertainty for their images. The larger errors associated with images 2 and 3 is due to the fact that these images

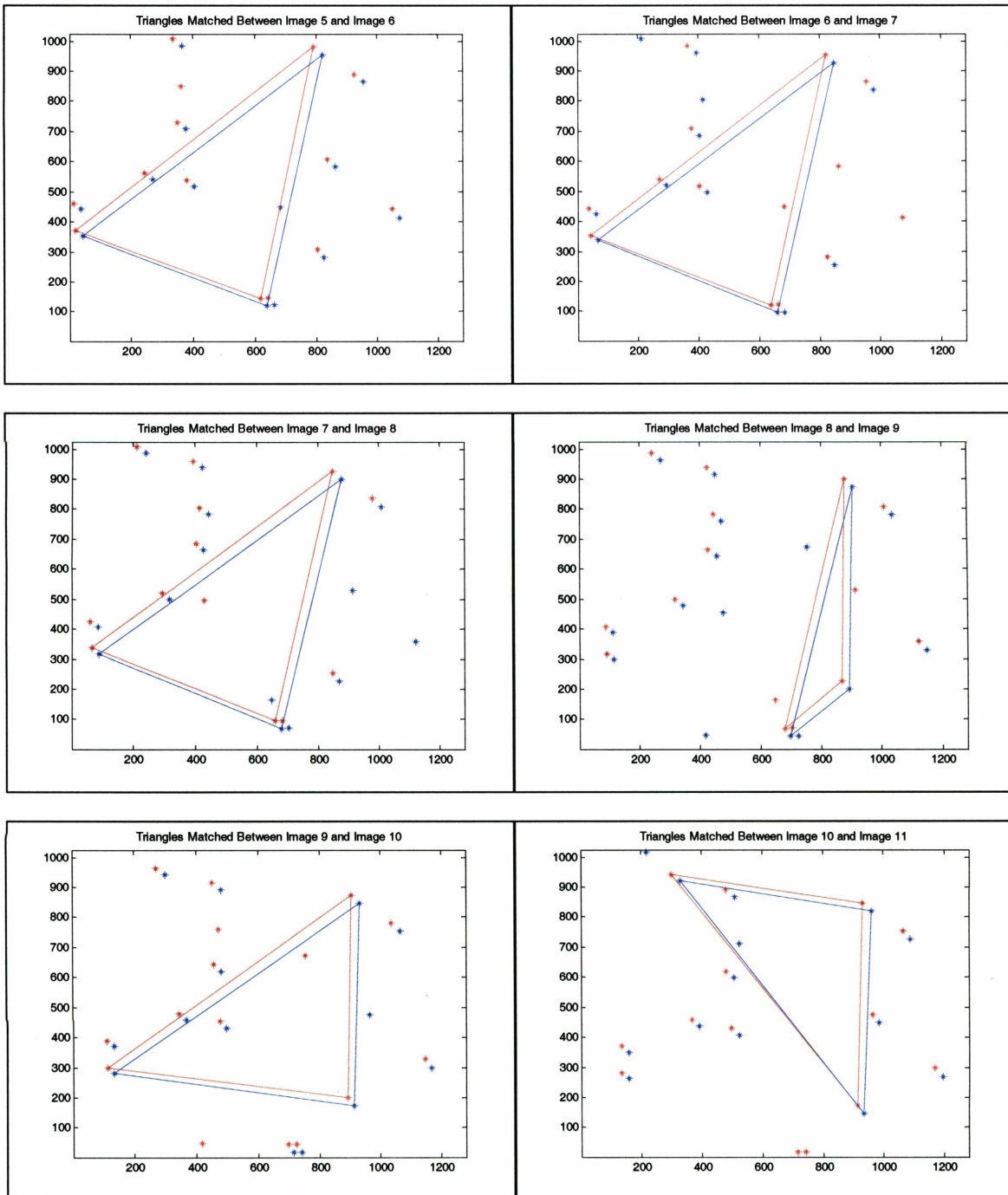


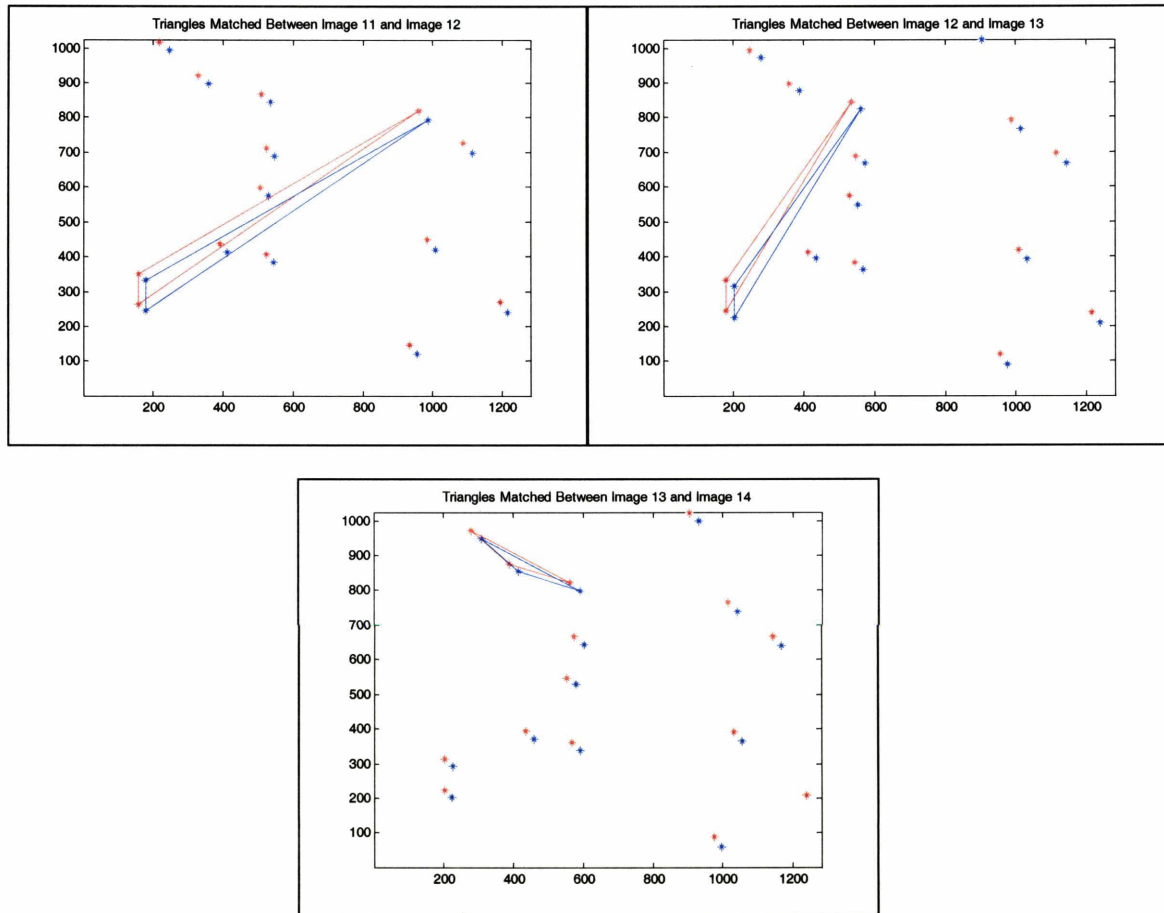
contain the dimmest sets of stars (recall that the max  $M_V$  was allowed to reach 6), and therefore the stars with the lowest centroid accuracies. The errors in images 2 and 3 increase for higher noise threshold levels because at higher thresholds, fewer of these dim stars exceed the threshold, and so the number of stars available for attitude information decreases, which also causes a decrease in attitude confidence.

## 7.2 Triangle Match Performance

The sequence of fourteen images was also used to test the performance of the triangle match algorithm. For this test, starCentroid3 was used and the noise threshold was set to the  $1\sigma_{\text{noise}}$  level. Figure 7.31 shows the sequence of triangles that were selected and correctly matched between the consecutive images. It took approximately 9.06s for the triangle match algorithm to successfully determine attitude information for all fourteen images.







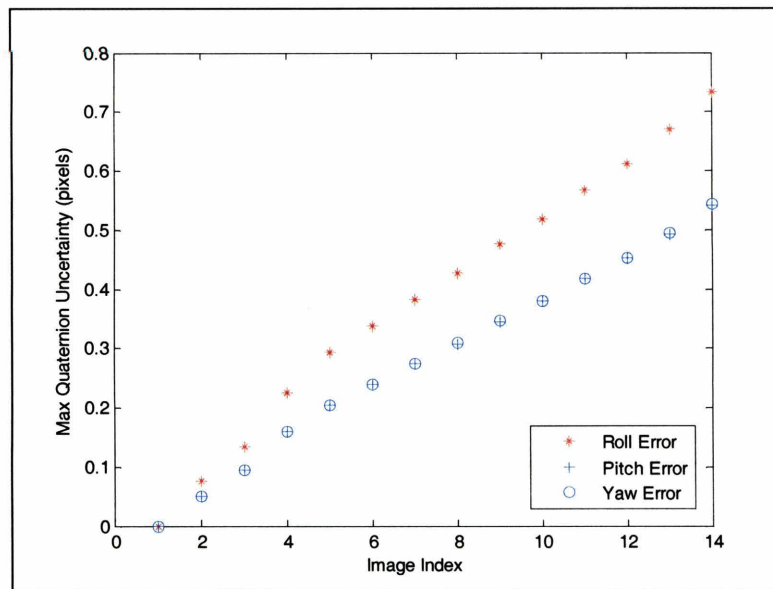
**Figure 7.31** Triangles Selected for Matching Using the Triangle Match Algorithm

Examination of these images in sequence shows how easily stars fall on and off of the image frame based not only on the star tracker’s motion, but more often due to the low SNR values (given that some of the stars are as dim as  $M_V = 6$ ). Because of the low  $M_V$  allowed, the noise threshold had to be set to a low number of sigmas. This enabled the possibility that some of the bright spots deemed “stars” were actually noise spikes.

Nonetheless, the triangle match successfully matched three stars between consecutive images for this series of images. The stars have been correctly identified by the centroider, as verified by comparing Figure 7.31 to the images that use the same centroider and noise threshold level in the centroid accuracy section above. In some cases, the same stars are selected to be part of the matching triangle; in general ones that are selected for multiple images are some of the brighter stars in the image. This is due to the looping structure used to select the stars that form the triangles, which always begins with the brightest stars, but conveniently varies the indices so

that no single star is chosen for more than three consecutive in case that star does not actually appear in both images to be matched. This trend is more readily seen in the n-vertex match trade study.

Besides the ability to accurately track stellar triads, the triangle match algorithm must be able to provide reliable information to the QUEST algorithm. This ability is measured by calculating the uncertainty associated with the attitude quaternion produced for each image, which is displayed in Figure 7.32.



**Figure 7.32 Maximum Attitude Quaternion Uncertainty Using Triangle Match Algorithm**

This plot shows that the amount of uncertainty accumulates with each added image. This is due to the fact that each attitude quaternion is associated with an error component, which is defined as

$$\varepsilon_j^2 = \varepsilon_i^2 + \varepsilon_{i,j}^2 \quad (7.1)$$

where  $\varepsilon_j$  is the error propagation from the 1<sup>st</sup> to the j<sup>th</sup> image,  $\varepsilon_i$  is the error propagation from the 1<sup>st</sup> to the i<sup>th</sup> image, and  $\varepsilon_{i,j}$  is the error that results from the motion between the i<sup>th</sup> and j<sup>th</sup> images. Therefore, in frame-to-frame matching, the greater the number of consecutive images that are matched, the more the error accumulates.

In the case of these fourteen images, the direction of maximum uncertainty was always found in the roll direction. The uncertainty in pitch and yaw is determined solely by the centroid uncertainties, whereas in roll, the distance between the stars becomes a factor. For pitch and yaw, the uncertainty is roughly equal to  $\Delta\theta_{pix}$ , the centroid uncertainty in pixels. For roll, it is roughly  $\Delta\theta_{pix}/d$  where  $d$  is the distance between two stars and this ratio is in radians. Multiplying by  $pp/f$  to convert to pixels, and equating to  $\Delta\theta_{pix}$ ,  $d = pp/f$  is found to be the distance at which the roll uncertainty equals that of the pitch and yaw. For the current LIST and FAR-MST hardware, this is approximately 2300 pixels, which is twice the width of the imager. Therefore the uncertainty in roll tends to be twice as much as that for pitch and yaw. The uncertainties in the pitch and yaw direction followed similar trends, but do not accumulate as much over time because each image contributes a smaller amount of uncertainty.

### 7.3 Rate Match Performance

The rate match algorithm was tested on the sequence of fourteen images to examine whether or not stellar locations could be accurately projected based on current rate information. The results of the rate matching are displayed in Figure 7.33 through Figure 7.44. As in the triangle match tests, `starCentroid3` and a noise threshold level of  $1\sigma_{noise}$  were implemented. It took approximately 15.12s for the rate match algorithm to successfully determine attitude information for all fourteen images.

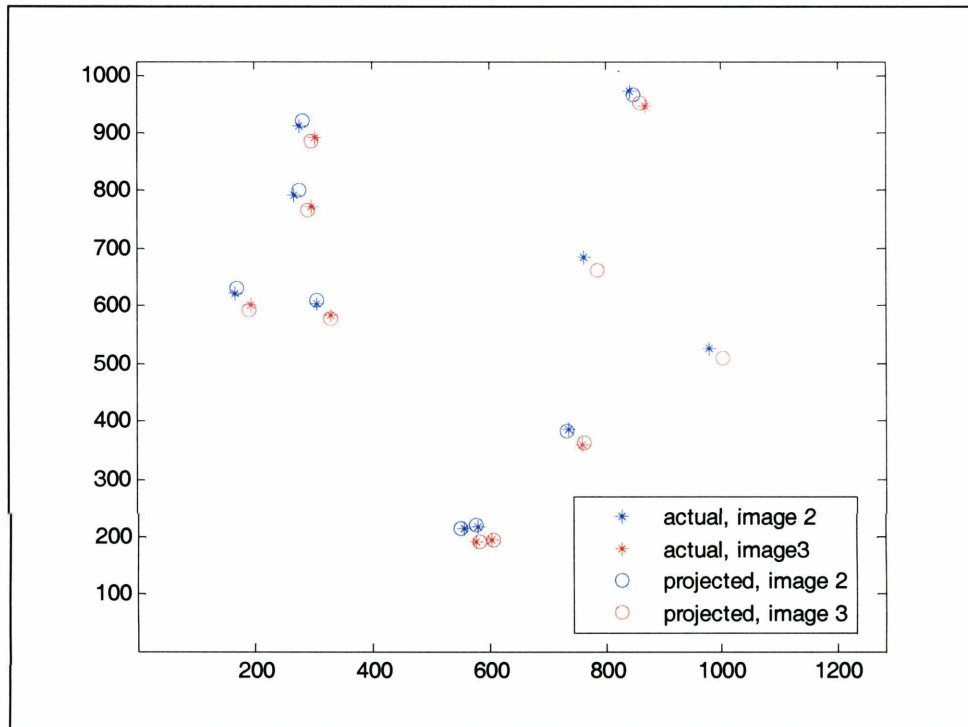


Figure 7.33 Rate Match Between Images 2 and 3

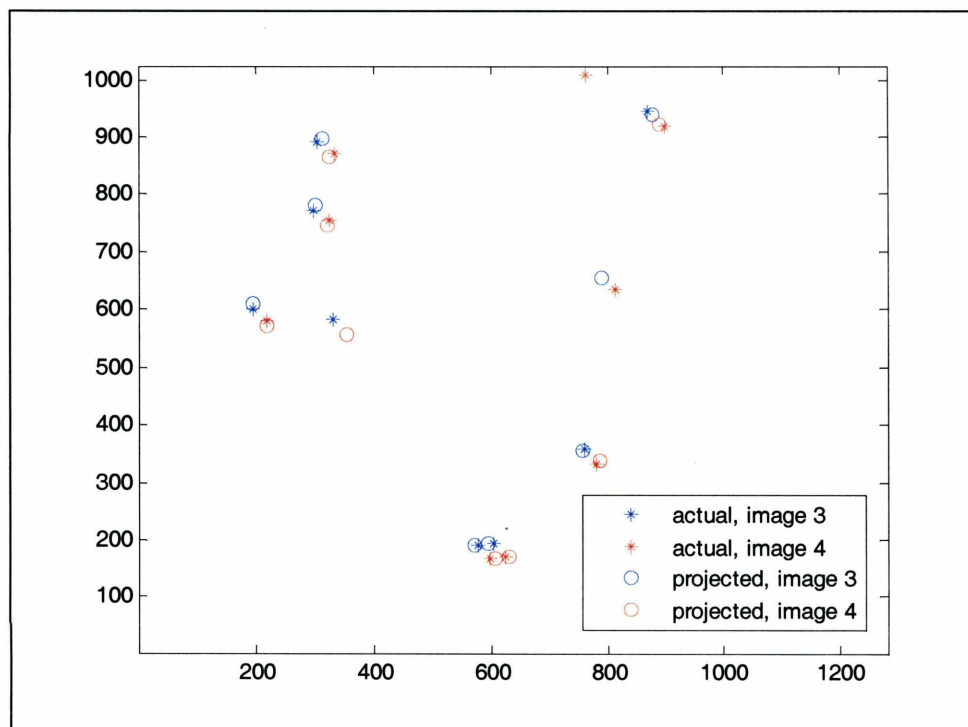


Figure 7.34 Rate Match Between Images 3 and 4



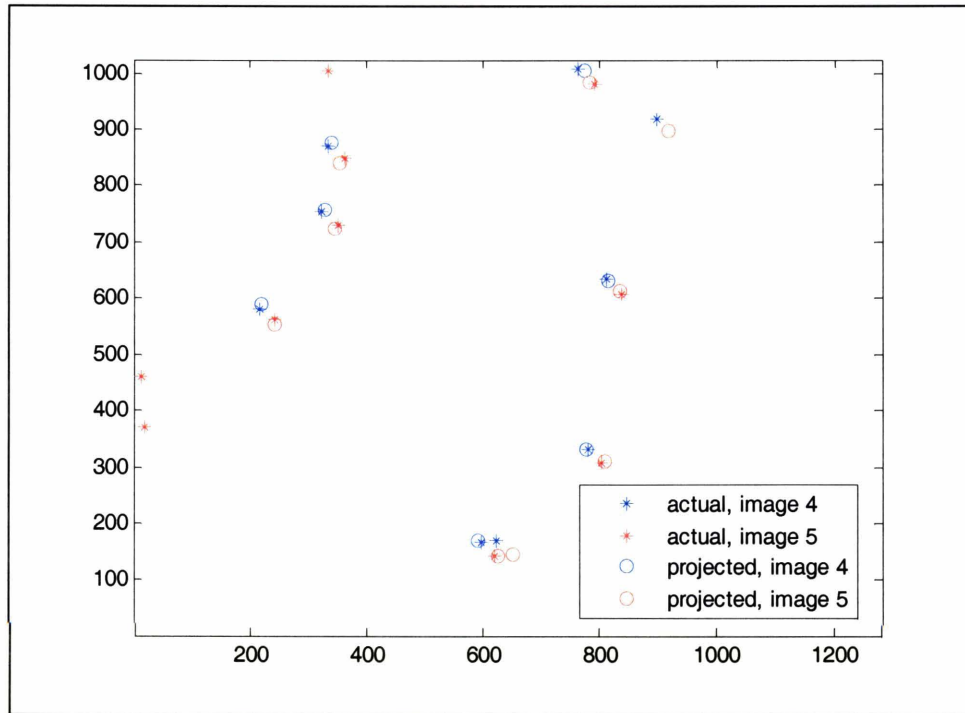


Figure 7.35 Rate Match Between Images 4 and 5

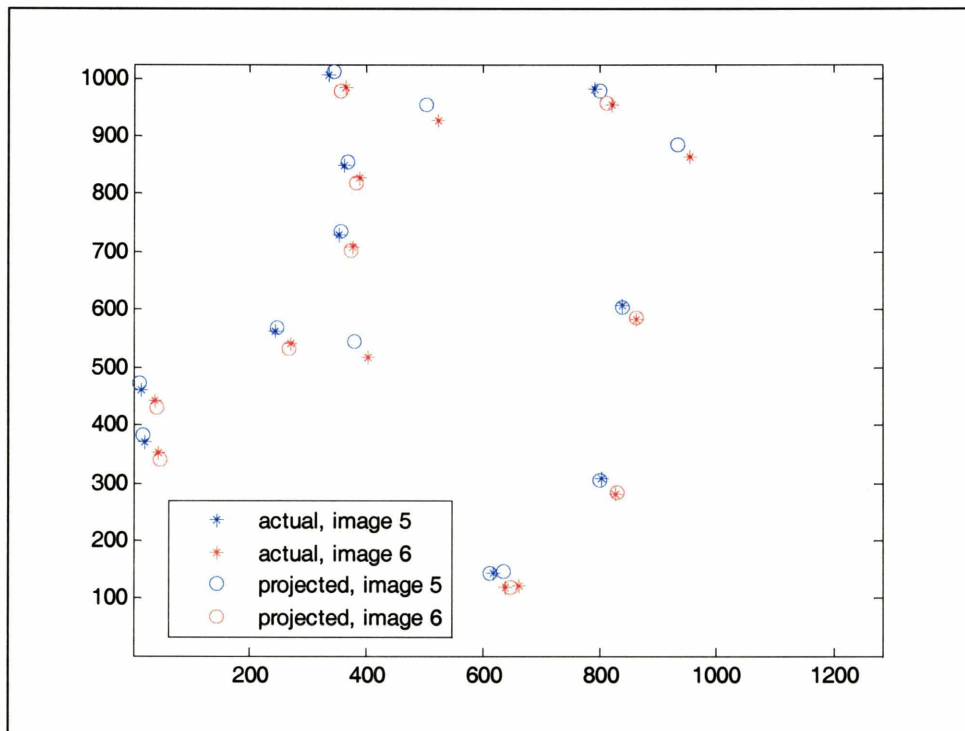


Figure 7.36 Rate Match Between Images 5 and 6

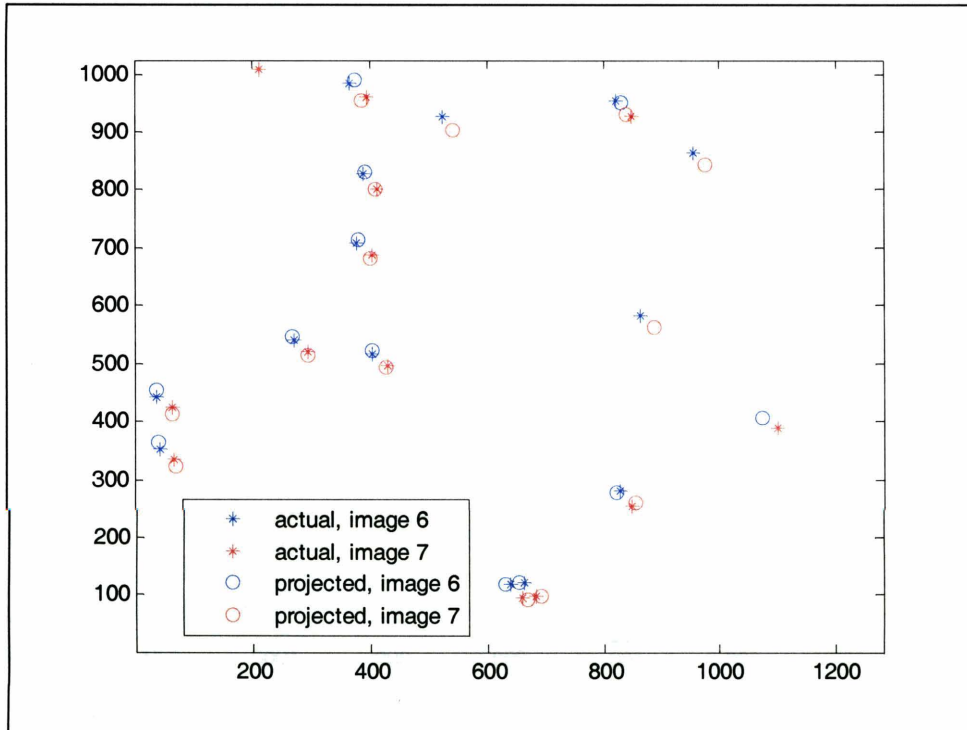


Figure 7.37 Rate Match Between Images 6 and 7

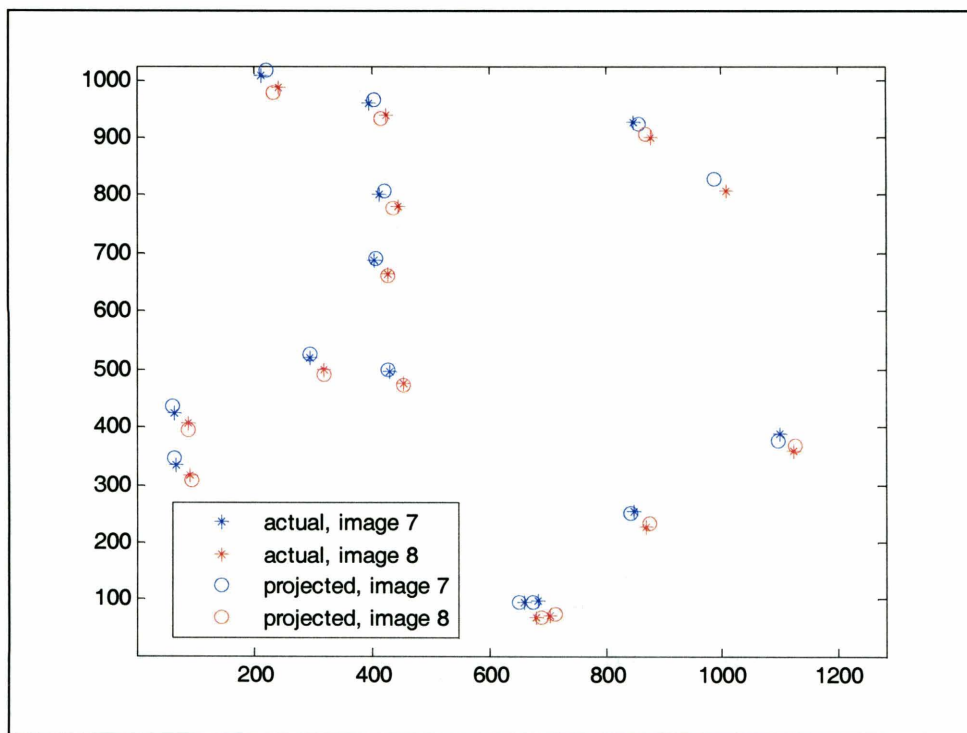


Figure 7.38 Rate Match Between Images 7 and 8



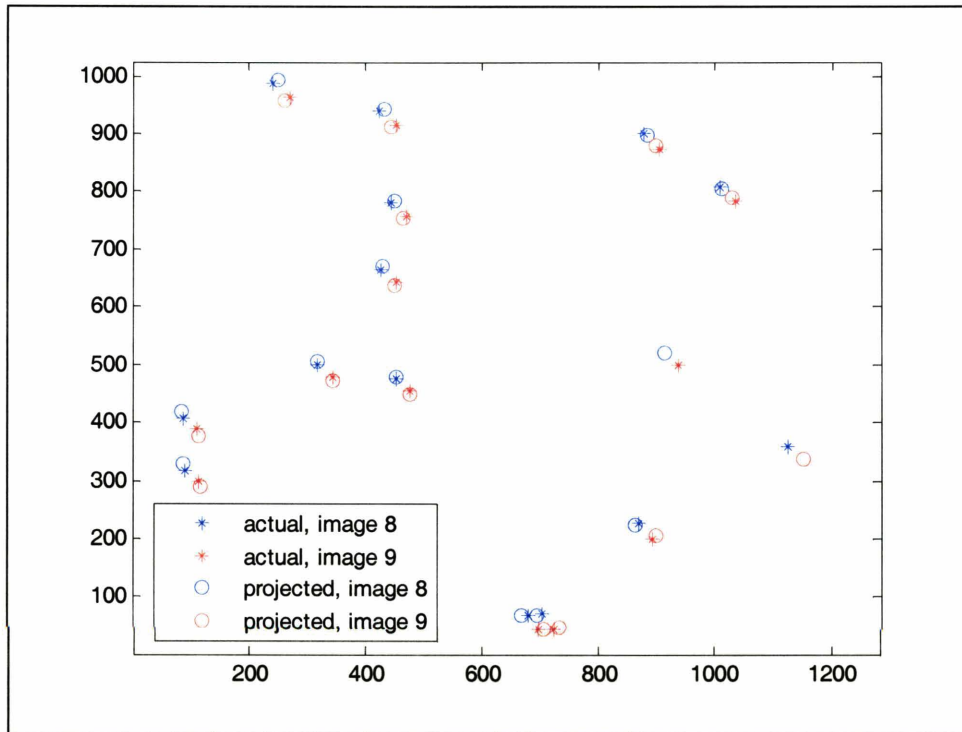


Figure 7.39 Rate Match Between Images 8 and 9

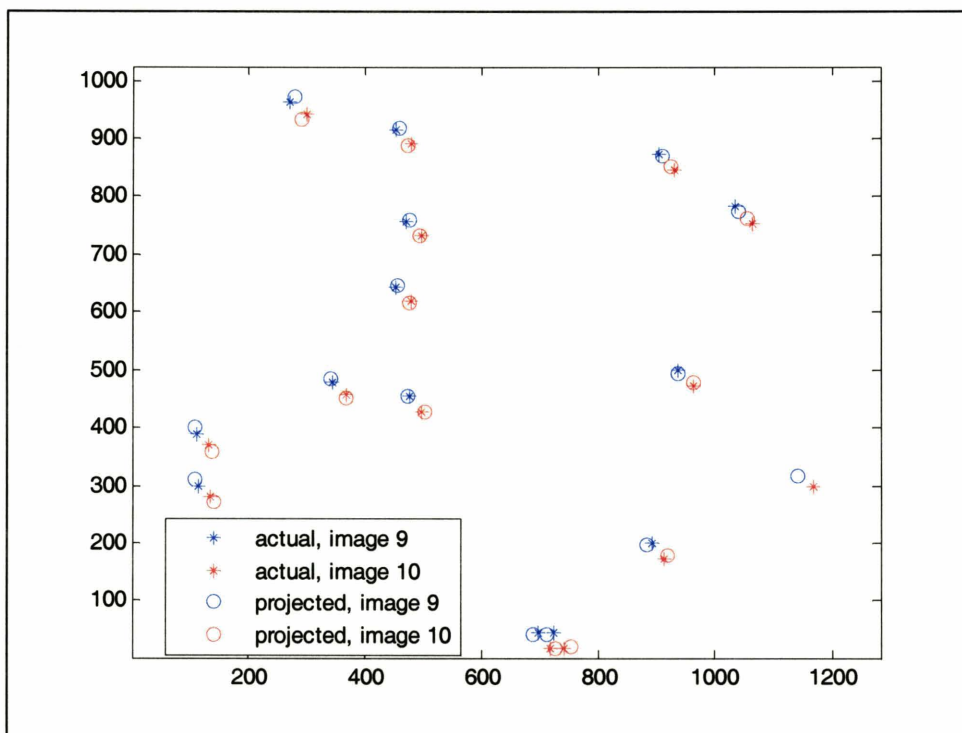


Figure 7.40 Rate Match Between Images 9 and 10

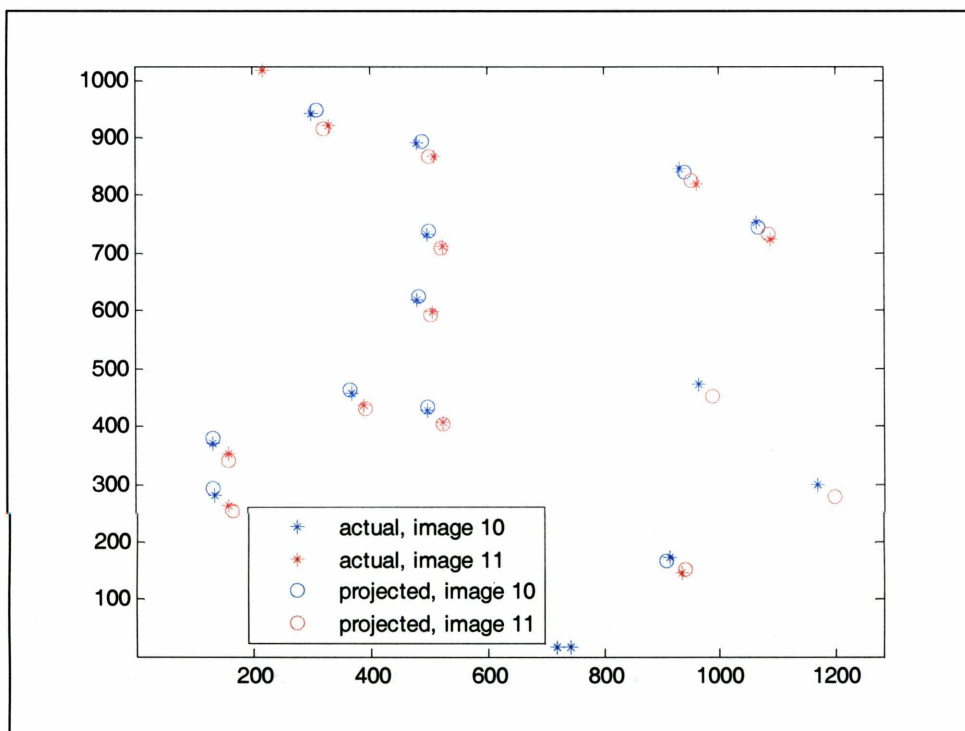


Figure 7.41 Rate Match Between Images 10 and 11

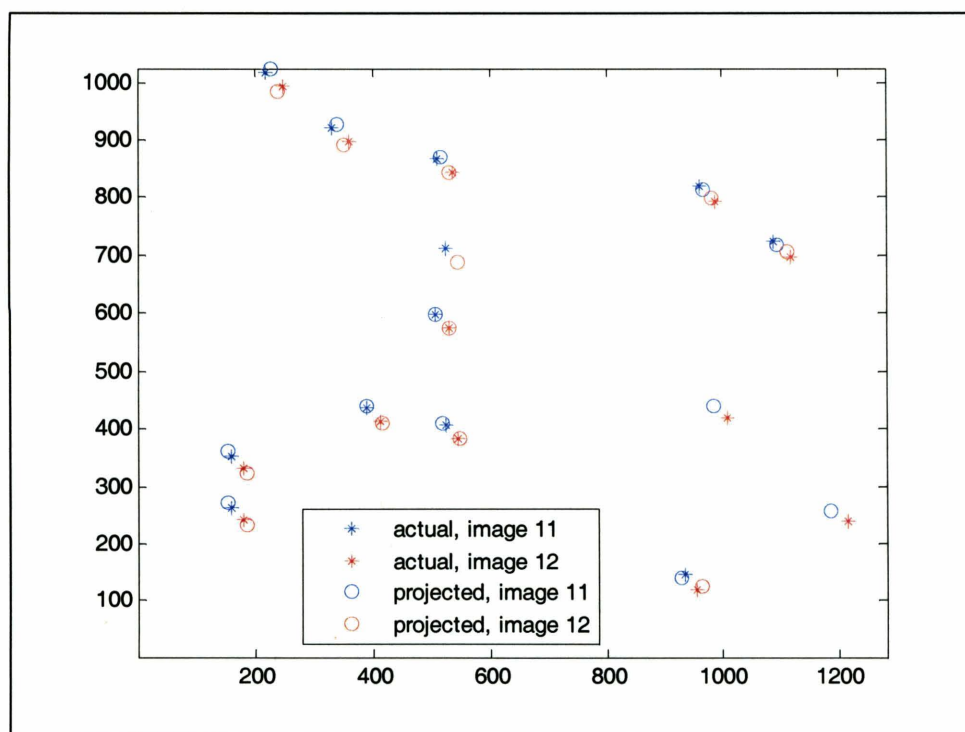


Figure 7.42 Rate Match Between Images 11 and 12

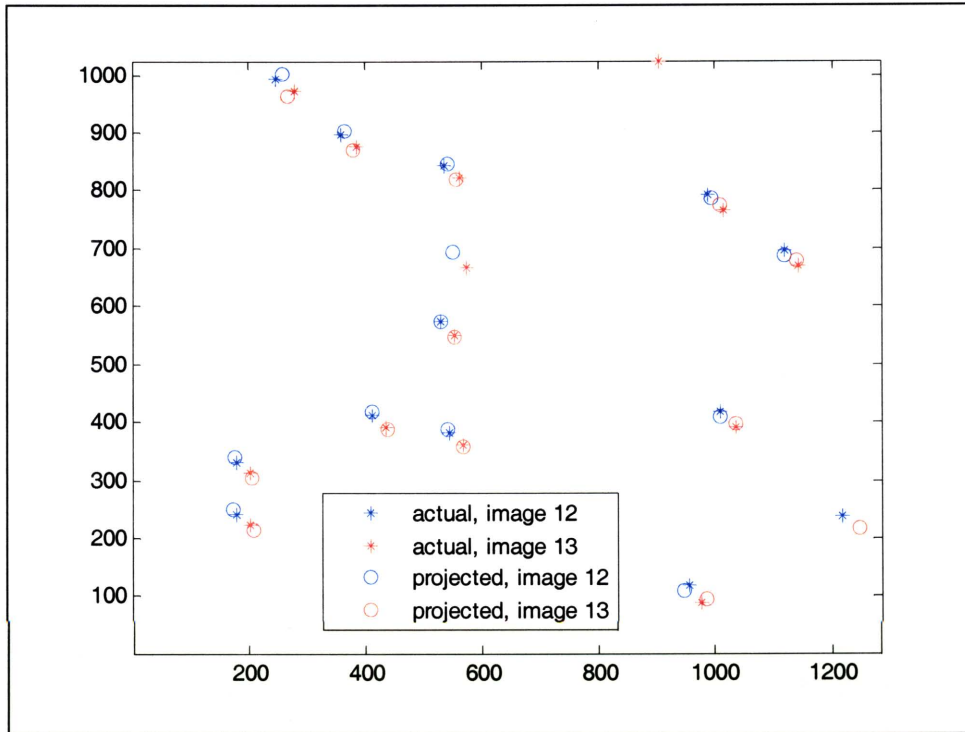


Figure 7.43 Rate Match Between Images 12 and 13

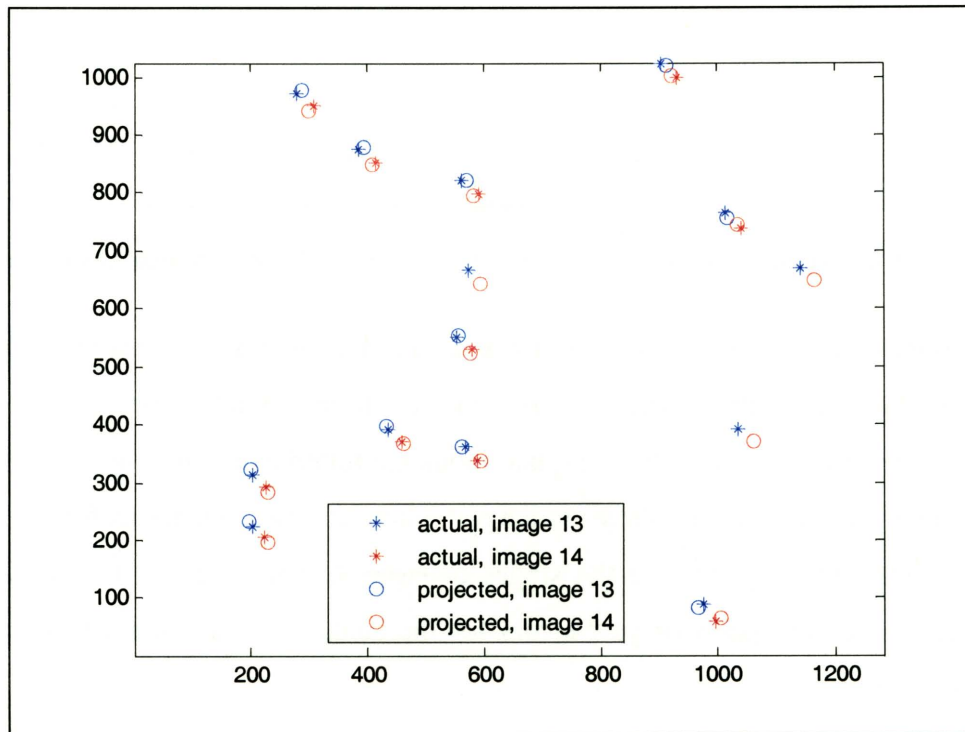
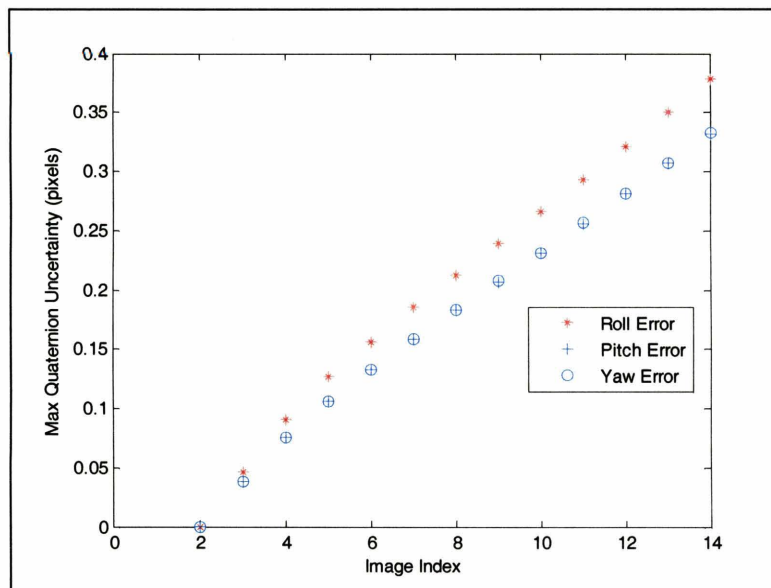


Figure 7.44 Rate Match Between Images 13 and 14

The rate match attempts to match as many stars as possible between images using the current rate information. Based on the plots above, the rate match was generally able to match a large percentage of the stars in one image to stars in the subsequent image. The cases where this did not occur were when an “old” star dropped below the noise floor or fell out of the image FOV or when a “new” star came out above the noise floor or entered the image FOV.

Attitude quaternion errors were examined based on the results of the error covariance matrix values in the roll, pitch, and yaw directions and are displayed in Figure 7.45.



**Figure 7.45 Maximum Attitude Quaternion Uncertainty Using Rate Match Algorithm**

As with the triangle match, the rate match errors accumulate as more images are considered in sequence. However, due to the predictive ability of the rate match, all of the stars are found and used, and therefore more accurate attitude quaternions are found using the rate match than using the triangle match; the lower uncertainties in the roll, pitch, and yaw directions for the rate match are visibly apparent when comparing the slopes of Figure 7.32 and Figure 7.45. Again, the roll errors are greater than the pitch or yaw errors for the same reasons discussed in the triangle match performance section.

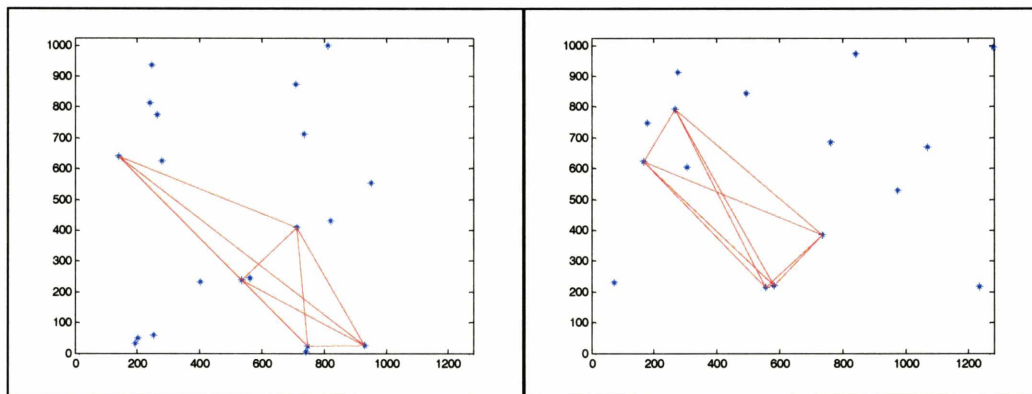
## 7.4 Attitude Quaternion Comparison Using n-vertex Match

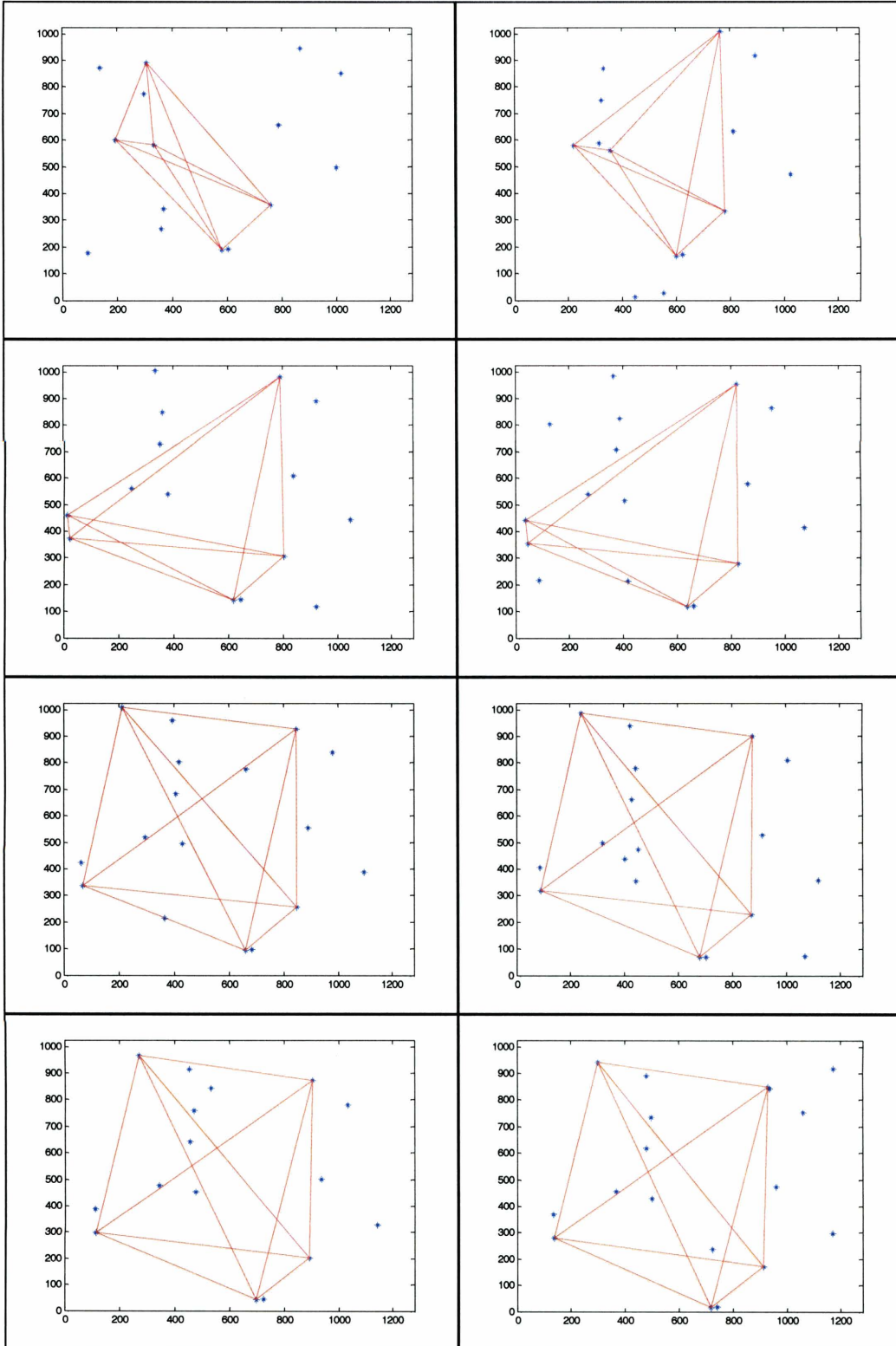
The n-vertex match using 3, 4, and 5 vertices was tested to evaluate the LIS abilities of the reduced star catalog and pattern matching algorithms. The quaternion that defines the motion of the spacecraft between images  $i$  and  $j$  is

$$q_j = q_i \Delta q_{i,j} \quad (7.2)$$

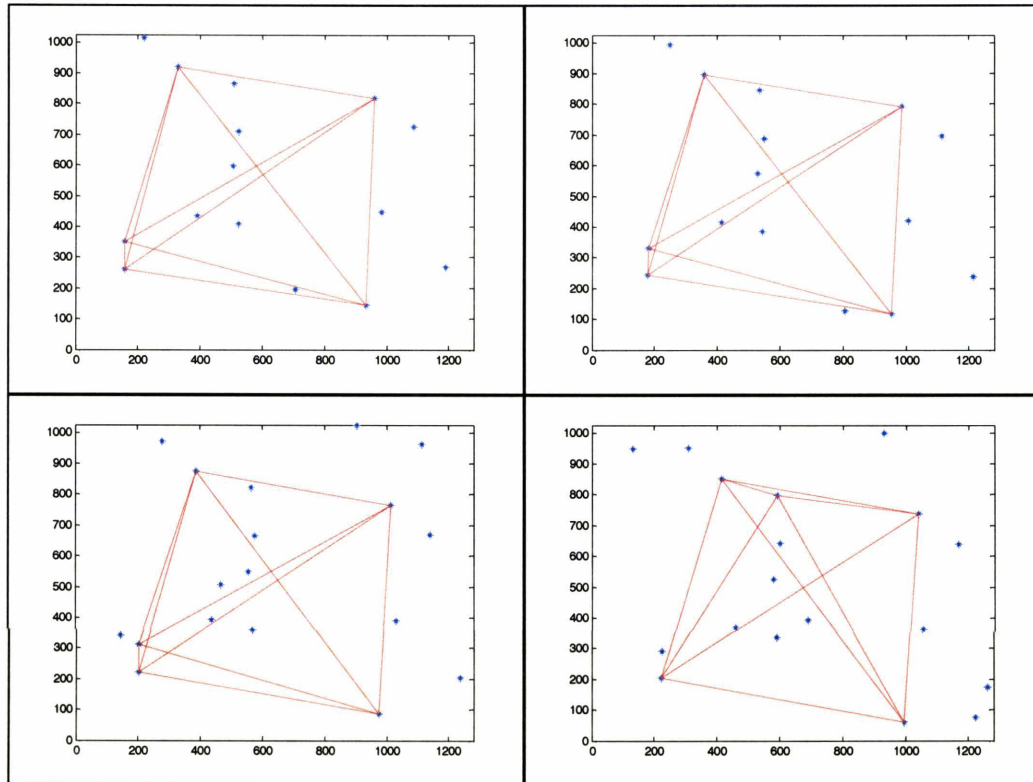
where  $q_i$  and  $q_j$  are the attitude quaternions of the  $i^{\text{th}}$  and  $j^{\text{th}}$  images, respectively, and  $\Delta q_{i,j}$  is the quaternion that defines the motion of the spacecraft from the  $i^{\text{th}}$  image to the  $j^{\text{th}}$  image. Each attitude quaternion has an error associated with it, as first described in the triangle matching performance section.

Three sets of data were collected for the n-vertex method, each using a value of  $n$  equal to 3, 4, or 5. As with the triangle match and rate match performance trade studies, starCentroid3 with a noise threshold level of  $1\sigma_{\text{noise}}$  was implemented. It took approximately 8.30s for the n-vertex match algorithm to successfully determine attitude information for all fourteen images; the time was relatively independent of the  $n$  value. The stars selected to match to the catalog when  $n$  equaled 5 for each of the fourteen images is displayed in Figure 7.46.









**Figure 7.46 Pentagon Selected for Matching Using the n-vertex Match Algorithm with  $n = 5$**

The three versions of the n-vertex match algorithm ( $n = 3$ ,  $n = 4$ , and  $n = 5$ ) successfully matched the appropriate number of stars to the star catalog for all fourteen images. Again, as seen in the triangle match, several of the same star polygons are used in consecutive images, which is result of the “smart” looping structure in the algorithms. The same two bright stars are present in images 5 through 14, and are always selected as two of the five stars shown in Figure 7.46.

While it is advantageous to have images with bright stars, from a trade study perspective this may lead to overly confident results because most regions of the sky do not contain stars of this magnitude, much less multiple stars of this magnitude. Nonetheless, as shown below, bright stars enable very accurate attitude quaternions to be generated.

Using the error covariance equations, errors in roll, pitch, and yaw were found for each of the fourteen images for the three n-vertex match tests. These results are displayed in Figure 7.47, Figure 7.48, and Figure 7.49.

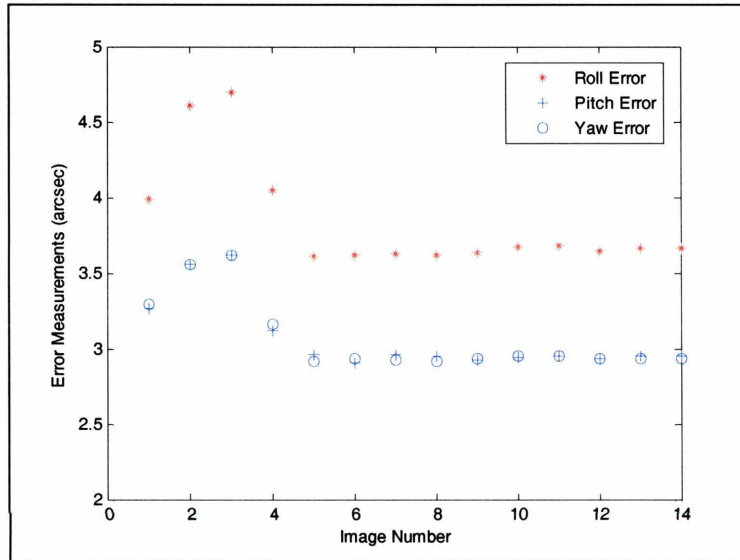


Figure 7.47 Uncertainty in Image Attitude Quaternion for n-vertex Match when  $n = 3$

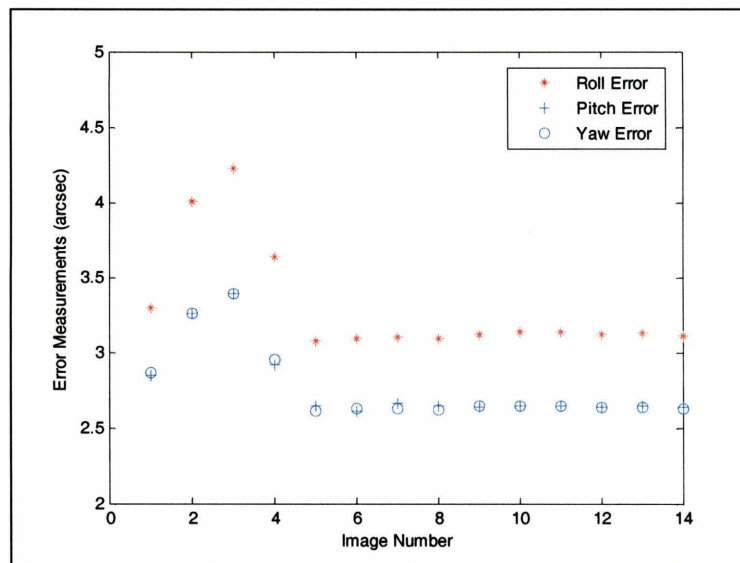
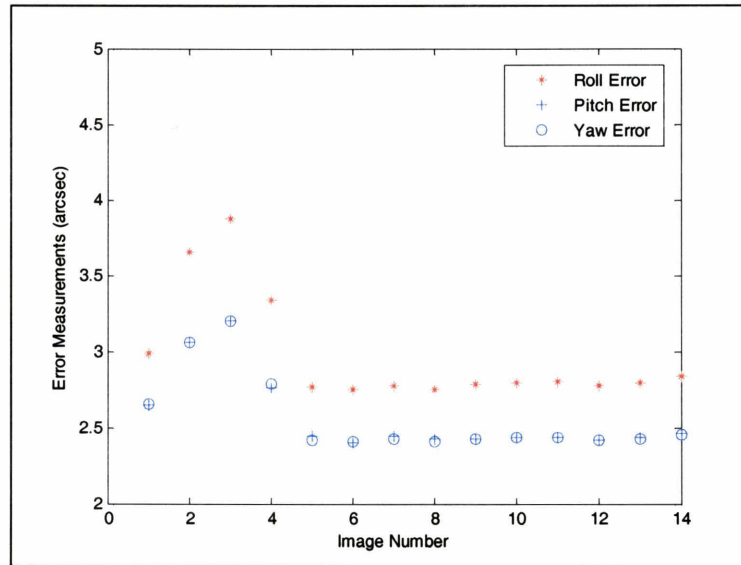


Figure 7.48 Uncertainty in Image Attitude Quaternion for n-vertex Match when  $n = 4$





**Figure 7.49** Uncertainty in Image Attitude Quaternion for  $n$ -vertex Match when  $n = 5$

As shown above, the errors in the pitch and yaw directions are similar to one another and smaller than the errors in the roll direction, which is to be expected. Also shown above is the improvement in quaternion accuracy when brighter stars are present. It was not until frame # 4 that one of the two bright stars appeared in the FOV and not until frame # 5 that both bright stars were present. Once these two stars were present, the error measurements in all three directions decreased and remained fairly constant. The key trend from these results is the fact that the errors decrease when using more vertices. The primary difference between these images and the images before is that the uncertainty errors do not accumulate over consecutive images. This is due to the fact that the images are always matched to a star catalog with known interstellar angles.



# Chapter 8

## CONCLUSIONS AND FUTURE WORK

### 8.1 Conclusions

The primary objective of LIST and FAR-MST star trackers is to provide sufficient attitude information to micro-satellites and spacecraft without the higher mass, size, power consumption, and cost requisites. The work presented in this thesis has demonstrated that it is possible to achieve adequate relative and absolute attitude information by means of appropriate hardware selection and algorithm establishment. Algorithm performance has been quantified by applying hardware parameters defined by a double lens optics system and a CMOS imager to an image sequence representing a specific region of the celestial sphere.

#### 8.1.1 The Optimal Centroider

Accurate centroiding is essential for star trackers because pattern matching success rates and attitude quaternion accuracy are highly dependent on centroiding precision. Three different centroiders were created to quickly and accurately locate the stars' centers on an imager. The two weighted sum algorithms, `starCentroid2` and `starCentroid3`, were compared against a maximum likelihood estimator. Although the MLE produced slightly more precise centroids, the significant increase in time requirements for the MLE made this centroider impractical for LIST and FAR-MST star trackers.

The two weighted sum centroiders were compared against one another to determine which performed the most accurately and in the shortest amount of time. The performance of each was highly correlated with image SNR, as well as the selected noise threshold level. For the series of fourteen images used in the trade study, speed and accuracy results help dictate which centroider will perform best under various conditions that include the number of stars in the image and the selected noise threshold level. The overall optimal centroider under four sets

of conditions is based on the results of the speed and accuracy benchmarking and is displayed in Table 8.1

**Table 8.1 Optimal Centroider Based on Speed and Accuracy Performance Results**

# of stars in image	noise threshold level (# $\sigma_{\text{noise}}$ )	optimal centroider		
		<i>speed</i>	<i>accuracy</i>	<i>overall</i>
few (< 4)	high (> $3\sigma_{\text{noise}}$ )	starCentroid2	starCentroid2	starCentroid2
few (< 4)	low (< $1.5\sigma_{\text{noise}}$ )	starCentroid3	either	starCentroid3
many (> 4)	high (> $3\sigma_{\text{noise}}$ )	starCentroid2	either	starCentroid2
many (> 4)	low (< $1.5\sigma_{\text{noise}}$ )	starCentroid3	starCentroid3	starCentroid3

The optimal centroider based solely on speed was taken directly from the results in Figure 7.1 through Figure 7.5. The speed results were found to be independent of the number of stars in the images, but were dependent on the level of the noise threshold, and so the number of stars in the images was not considered from a speed standard of optimization. The accuracy tests did not specifically examine differences between few or many numbers of stars in the images; however this number was considered when determining the optimal centroider from an accuracy perspective. At high noise threshold levels, when few stars are present in the image, starCentroid2 is the optimal choice because starCentroid2 was able to locate more stars on the image, and was not found to centroid noise spikes at the higher threshold levels. On the other hand, when many stars are available and the noise threshold is low, starCentroid3 is the optimal choice because although starCentroid3 finds fewer stars than starCentroid2, at the lower noise threshold levels, it was not found to centroid noise spikes.

Determining the optimal centroider from an accuracy perspective for the remaining two conditions is more difficult because conflicting results are possible. In the case where few stars are present and the noise threshold level is low, starCentroid2 is advantageous because it shows a higher likelihood of actually finding the stars, but also has a higher potential for centroiding noise spikes. Alternatively starCentroid3 will not centroid noise spikes, but has a higher probability of not finding enough stars to perform attitude determination. The case when many stars are available and the noise threshold level is set high is the ideal case for star trackers because it has the best probability of centroiding many stars and not noise spikes; both

starCentroid2 and starCentroid3 would perform well under these conditions. The overall optimal centroider was determined by combining the results from the speed and accuracy standpoints; the results of the optimal speed centroiders were selected for the overall optimal centroider in the cases where the optimal accuracy centroiders could choose from either starCentroid2 or starCentroid3.

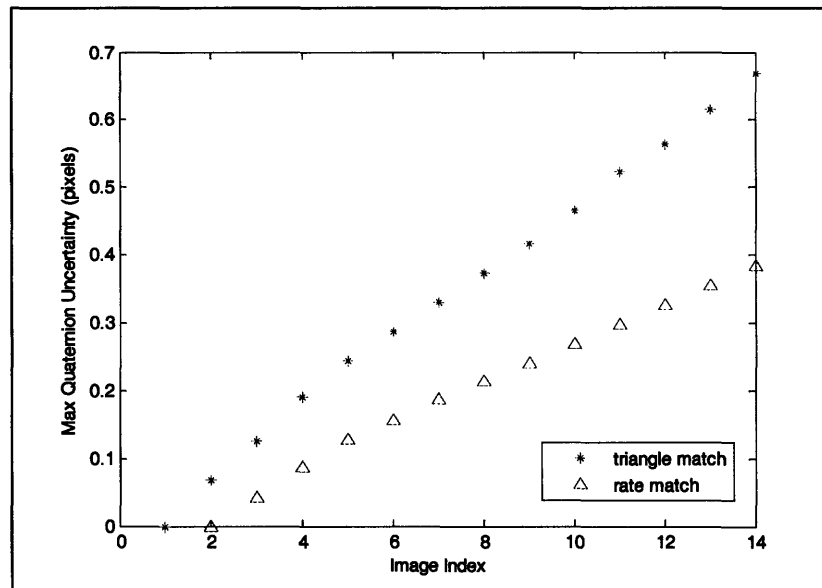
Regardless of the centroider, centroiding was found to be more accurate for brighter stars, where noise electrons fail to contribute as much to the overall stellar intensity. This was expressed by the covariance values in Figure 7.26 through Figure 7.30, as images that contained brighter stars (images 5 through 14) produced better centroids, and therefore smaller uncertainties in attitude quaternions.

### **8.1.2 Pattern Matching**

The primary pattern matching algorithms selected for LIST and FAR-MST include the triangle match, rate match, and n-vertex match algorithms with n equal to 3, 4, and 5. The triangle and rate match algorithms were implemented in frame-to-frame matching, while the three versions of the n-vertex match were implemented in LIS scenarios. Thus, following implementation of QUEST, the triangle and rate matching techniques led to relative attitude information, while the n-vertex match algorithm led to absolute attitude information.

Each matching algorithm has advantages and disadvantages, which are exemplified through the algorithms' ability to accurately determine attitude information and overall speed of execution. All five algorithms were able to determine attitude information for each of the fourteen images. The difference between the algorithms was the level of uncertainty associated with each set of attitude quaternions. The n-vertex match held the lowest attitude uncertainties, and therefore provided the best attitude information. This was due to the fact that stars were matched to a star catalog, as opposed to a previous image. Because the cataloged stars have known interstellar angles that are not corrupted by noise variations, the star catalog data sent to QUEST is much more accurate than that sent from an image.

The triangle match and rate match produce accumulating uncertainty errors over consecutive images, and the uncertainties were the largest in the roll direction. Figure 8.1 compares the uncertainties of each image's attitude quaternion for the triangle match and rate match. Comparisons of the pitch and yaw uncertainties produce similar results.



**Figure 8.1 Comparison of Triangle Match and Rate Match Quaternion Uncertainties in Roll Direction**

The predictive ability of the rate match to find all of the matchable stars proved to be advantageous because the uncertainty errors of the rate match accumulated at a slower rate than those of the triangle match, as seen by the rate match's smaller slope.

Apart from accuracy, computational requirements for the pattern matching algorithms were considered. Table 8.2 compares the time requirements of each algorithm to threshold, centroid, and pattern or rate match the stars and output an attitude quaternion. These times were collected using Matlab versions of the algorithms; the actual algorithms will be run in C, which will produce much faster times than those listed below. The Matlab times have been included to show the relative time requirements of the different matching algorithms.

**Table 8.2 Time Comparisons for Matching Algorithms**

algorithm:	triangle match	rate match	3-vertex match	4-vertex match	5-vertex match
time (s):	9.06	15.12	8.30	8.27	8.33

The fastest times correspond to the n-vertex matching algorithms, although the triangle match algorithm is also relatively fast. Although the n-vertex match algorithms require matching the stars in the current image to an onboard star catalog, the use of the k-vector prevents the long time constraints that one would think would be required to match the image stars to those in the catalog. The rate match is the most inefficient method from a computational standpoint. The rate match is costly because star locations in the earlier image must be projected forward in time and star locations in the later image must be projected back in time in order for the current attitude quaternion to assist in correctly matching stars.

These conclusions show that the ideal pattern matching algorithm is the n-vertex match. This technique provides the least amount of uncertainty in the attitude quaternion determination while simultaneously requiring the least amount of computational time. The tradeoff is the requirement of the onboard star catalog, which is only problematic if memory space is limited. However, creating a reduced star catalog, with properties similar to those defined for the FAR-MST catalog, requires a minimal amount of additional space. If, on the other hand, an onboard star catalog is not present, the user must decide whether accuracy or speed is most important in selecting between the triangle match and rate match.

The primary reason for multiple pattern matching algorithms is so that the star tracker can function in a dynamic manner; LIST and FAR-MST are not restricted to one algorithm onboard. The pattern matching algorithm of choice for frame-to-frame matching will be selected based on the number of stars in the FOV and whether or not current rate information is available. If the matching algorithm originally selected fails, then star tracker will attempt to implement a different matching algorithm. An example of dynamic implementation is displayed in Figure 8.2

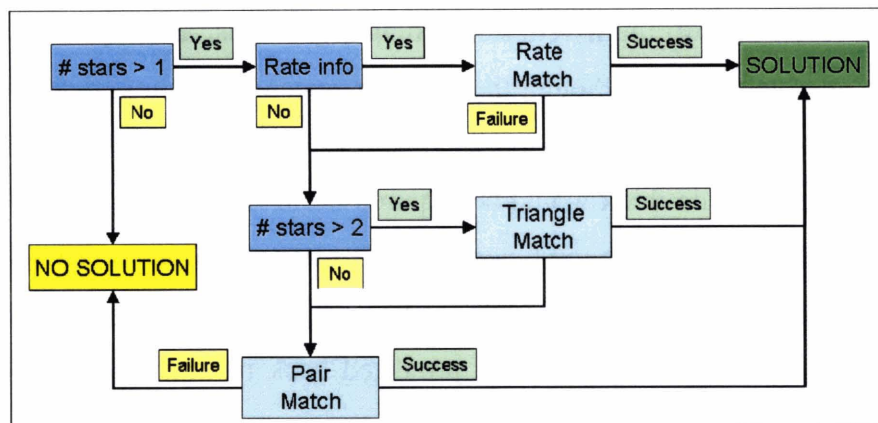


Figure 8.2 Dynamic Implementation of Frame-to-Frame Pattern Matching Algorithms

### 8.1.3 Attitude Determination

The QUEST algorithm was selected over other attitude determining methods primarily because of its speed of execution. The low computational requirements are a direct result of the fact that QUEST finds an approximation to the actual attitude quaternion that is based on the centroiding and pattern matching results. QUEST is highly efficient in that it provides sufficient accuracy in terms of attitude information for LIST and FAR-MST star trackers.

## 8.2 Future Work

The current hardware and software properties defined for LIST and FAR-MST star trackers have the potential for improvements through the accomplishment of future work. The focus of this thesis revolves around the software algorithms for centroiding, pattern matching, and attitude determination, and so the future work concentrates on these areas.

### 8.2.1 Quantifying Multiple Star Pairs

It is widely acknowledged in the literature that, in a given image, multiple star pairs can have the same interstellar angle because of centroid inaccuracies and noise variations. It is also possible, though much more unlikely, for identical star triads to exist within the same image. However, it would be interesting to quantify these amounts. This could be accomplished by selecting a FOV, setting an interstellar distance tolerance to account for centroid errors and noise, and looping over known regions of the sky using a star catalog such as the NASA



SKY2000. FOV and tolerances can be varied to quantify the minimum or average number of star pairs, triads, etc that occur over the entire sky, and perhaps help predict which regions in the sky pose the most threat. This analysis would be useful to the star tracker when selecting the current pattern matching algorithm to implement. For example, if particular regions of the sky are found to contain multiple star pairs within the same FOV, the user would know to implement the triangle match or rate match. However, if other regions of the sky do not contain multiple star pairs, perhaps the pair match could be implemented to reduce computational time.

### **8.2.2 Additional Benchmarking**

The performance of the centroiding, pattern and rate matching, and attitude quaternion determination algorithms is based on the results from a single set of fourteen images. These images were generated by a maneuver profile that was designed to rotate at a slow, constant rate. Additional trade studies can be undertaken that would improve the confidence level of these performance results. These trade studies can include implementing the algorithms on different regions of the sky, on an increased number of images, and on a maneuver profile that “tumbles” in various directions at different rates.

One of the problems with the selected region of the sky was that two very bright stars were present in images 5 through 14. Had these regions not contained such bright stars, performance results would most likely be reduced. Therefore it would be beneficial to select a variety of different regions in the sky that contain dim stars, bright stars, few stars, and many stars. This knowledge will assist in determining the most appropriate centroiding algorithms and pattern/rate match algorithms. The alternative to selecting several different regions in the sky is to create a maneuver profile that spans several hundred, or perhaps a thousand, images. Testing an extensive sequence of images will enable more statistically valid results to be quantified, and enable a higher confidence level for performance characteristics overall. Finally, realistic operating conditions may require the star tracker to handle more erratic motions, and so it would be beneficial to test performance for random movements at higher tumble rates.

Although many of the hardware elements have been defined, such as the IBIS5 CMOS imager, variations on hardware parameters will be useful in understanding the hardware selection

effects on performance. For example, a double lens was assumed to simplify considerations otherwise caused by the curvature-of-fields effect. However, if a single lens is assumed, stars would be spread across different PSF width depending on their location on the imager. Centroiding and pattern matching algorithms would have to be aware of the field curvature and adjust the properties of the stars according to their locations on the imager. Additionally, lens aperture plays a large roll in attitude determination. Larger lenses collect more light, which enables them to capture dimmer stars, and require shorter integration times. It would be interesting to see the effects of varying the lens diameter between 0.5cm and 2cm. Finally, different FOV choices will lead to varying performance characteristics, as stars will be spread over varying number of pixels depending on the FOV selected.

### **8.2.3 Dynamic Thresholding**

All of the analyses performed thus far have selected a single noise threshold level for the entire sequence of images. It may be beneficial to implement a dynamic thresholding mode that would automatically set the noise threshold level based on the properties of the recent images. For example, if previous images were found to have sparse numbers of stars, a lower noise threshold could be selected to assist in digging out as many stars as possible, and if future tumbles point the star tracker towards regions of the sky with greater numbers of stars, a higher noise threshold could be set to eliminate potential noise spikes.

### **8.2.4 Predictive Centroiding**

Several works in the literature have suggested the use of a predictive centroider that uses current rate information to predict stellar locations for subsequent images. A predictive centroider would enable increased computational speeds, as only small windows of the pixel matrix would require searching during the centroiding process since approximate stellar locations would be known. Additionally, in situations where several stellar centroids are known and an onboard star catalog is available, additional stars in the image that are close to the noise floor can be more easily located, thus increasing the number of stars that can be passed through the QUEST algorithm to improve attitude accuracy.

---

### **8.2.5 Reducing Error Propagation in Frame-to-Frame Pattern Matching**

One of the largest setbacks of frame-to-frame pattern matching is the natural propagation of uncertainty errors. Although it would be beneficial to create additional algorithms to assist in mitigating error propagation, this is generally not a possibility. Therefore, the best approach is to attempt improvements in centroiding accuracy, which will automatically tighten the error bounds on the attitude quaternion. If this is not possible, it may be necessary to introduce an onboard star catalog to enable the stars to be uniquely identified with the stars in the catalog.

### **8.2.6 Attitude determination**

Numerous techniques were considered prior to the selection of the QUEST algorithm. Although QUEST provides adequate attitude quaternions, it would be interesting to compare the performance of QUEST to the performance of the q-method to quantify any increases in time requirements or performance quality.



# APPENDIX A

## PROPERTIES OF THE QUATERNION

### **Definition**

Quaternions in this paper are described with the first component being the scalar portion of the quaternion and the second through fourth components being the vector portion. A particular quaternion  $\mathbf{q}$  is defined as

$$\mathbf{q} \triangleq \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \hat{n} \sin\left(\frac{\theta}{2}\right) \end{bmatrix} \quad (8.1)$$

where  $\theta$  is the angle of rotation and  $\hat{n}$  is the normalized axis of rotation defined as

$$\hat{n} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ \hat{z} \end{bmatrix}. \quad (8.2)$$

### **Quaternion Multiplication**

Let two quaternions be given as

$$\mathbf{q}_1 = a_1 + b_1\hat{i} + c_1\hat{j} + d_1\hat{k} = a_1 + \vec{v}_1 \quad (8.3)$$

and

$$\mathbf{q}_2 = a_2 + b_2\hat{i} + c_2\hat{j} + d_2\hat{k} = a_2 + \vec{v}_2 \quad (8.4)$$

where  $\hat{i}$ ,  $\hat{j}$ , and  $\hat{k}$  are the unit normal vectors along the axes of the 3D Cartesian coordinate frame. The product of the two quaternions is then found by<sup>[79]</sup>

$$\mathbf{q}_1 \otimes \mathbf{q}_2 = \begin{bmatrix} a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2 \\ a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2 \\ a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2 \\ a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2 \end{bmatrix} = \begin{bmatrix} a_1 a_2 - (\vec{v}_1 \bullet \vec{v}_2) \\ a_1 \vec{v}_2 + a_2 \vec{v}_1 + (\vec{v}_1 \times \vec{v}_2) \end{bmatrix}. \quad (8.5)$$

### Quaternion Inverse

The quaternion conjugate of (8.3) is

$$\bar{\mathbf{q}}_1 = a_1 - b_1 \hat{i} - c_1 \hat{j} - d_1 \hat{k} = a_1 - \vec{v}_1. \quad (8.6)$$

Using this information, the inverse of a quaternion is given by<sup>[79]</sup>

$$\mathbf{q}_1^{-1} = \frac{\bar{\mathbf{q}}_1}{\mathbf{q}_1 \bar{\mathbf{q}}_1}. \quad (8.7)$$

### Frame Rotation vs. Vector Rotation

The standard rotation matrix rotates a reference frame about a desired axis. For example, in the 3D Cartesian coordinate frame, a rotation about the z-axis by  $\theta$  degrees the rotation matrix is defined as

$$\mathbf{R}_z = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.8)$$

which corresponds to a rotation quaternion of

$$\mathbf{q}_{R_z} = \frac{1}{\sqrt{2}} \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}. \quad (8.9)$$

The correct rotation of a vector by a quaternion depends on whether a frame rotation or a vector rotation is desired.<sup>[80]</sup> A frame rotation corresponds to a fixed vector with a reference frame that rotates around the vector, whereas a vector rotation corresponds to a vector that rotates inside a fixed reference frame.

Rotating a vector by a quaternion using a frame rotation requires

$$\vec{v}' = \mathbf{q}^{-1} \cdot \vec{v} \cdot \mathbf{q}. \quad (8.10)$$

Rotating a vector by a series of  $n$  quaternions using frame rotations requires

$$\vec{v}' = \mathbf{q}_n^{-1} \dots \mathbf{q}_2^{-1} \mathbf{q}_1^{-1} \cdot \vec{v} \cdot \mathbf{q}_1 \mathbf{q}_2 \dots \mathbf{q}_n. \quad (8.11)$$

On the other hand, rotating a vector by a quaternion using a vector rotation requires

$$\vec{v}' = \mathbf{q} \cdot \vec{v} \cdot \mathbf{q}^{-1}. \quad (8.12)$$

Rotating a vector by a series of  $n$  quaternions using vector rotations requires

$$\vec{v}' = \mathbf{q}_n \dots \mathbf{q}_2 \mathbf{q}_1 \cdot \vec{v} \cdot \mathbf{q}_1^{-1} \mathbf{q}_2^{-1} \dots \mathbf{q}_n^{-1}. \quad (8.13)$$

In this thesis, all of the rotations of vectors by quaternions implement vector rotations.





# REFERENCES

- [1] Lerner, G. M., "Horizon Scanners," *Spacecraft Attitude Determination and Control*, 1978, pp. 166-180.
- [2] Wertz, J. R., "Appearance of the Earth at Infrared Wavelengths," *Spacecraft Attitude Determination and Control*, 1978, pp. 90-98.
- [3] Goodrich Corporation, 2001. Multi-Mission Earth Sensor Model 13-410 Datasheet.
- [4] Hatcher, N. M., "A Survey of Attitude Sensors for Spacecraft," *National Aeronautics and Space Administration Office of Technology Utilization*, NASA SP-145, Washington, D.C., 1967.
- [5] Lerner, G. M., "Sun Sensors," *Spacecraft Attitude Determination and Control*, 1978, pp. 155-166.
- [6] AeroAstro, Inc., August 2005. Medium Sun Sensors Datasheet.
- [7] Blaylock, B., T., "Magnetometers," *Spacecraft Attitude Determination and Control*, 1978, pp. 180-184.
- [8] National Space Science Data Center Photo Gallery: Spacecraft.  
[http://nssdc.gsfc.nasa.gov/photo\\_gallery/photogallery-spacecraft.html](http://nssdc.gsfc.nasa.gov/photo_gallery/photogallery-spacecraft.html).
- [9] Parkinson, B.W. and Spilker, J. J, e.d. *Global Positioning System: Theory and Applications*, Vol. 1. Progress in Aeronautics and Astronautics, Vol. 163.

- 
- [10] Remote Sensing Tutorial Section 13: Collecting Data at the Surface – Ground Truth and Imaging Spectroscopy. [http://rst.gsfc.nasa.gov/Sect13/Sect13\\_4.html](http://rst.gsfc.nasa.gov/Sect13/Sect13_4.html).
- [11] HowStuffWorks: Brain, M. and Harris, T. “How GPS Receivers Work.” <http://electronics.howstuffworks.com/gps.htm>.
- [12] Wikipedia: Global Positioning System, May 2006. [http://en.wikipedia.org/wiki/Global\\_Positioning\\_System](http://en.wikipedia.org/wiki/Global_Positioning_System).
- [13] Ju, G. and Junkins, J. L., “Overview of Star Tracker Technology and its Trends in Research and Development,” *Advances in the Astronautical Sciences*, Vol. 115, 2003, pp 461–477.
- [14] Fallon, L., “Star Sensors” *Spacecraft Attitude Determination and Control*, 1978, pp. 184-195.
- [15] “Science with the Galileo Star Scanner.” <http://www.mindspring.com/~feez/home.htm>.
- [16] Federation of American Scientists: Intelligence Resource Program, Rivet Joint Aircraft [http://www.fas.org/irp/program/collect/rivet\\_joint.htm](http://www.fas.org/irp/program/collect/rivet_joint.htm).
- [17] Federation of American Scientists: Intelligence Resource Program, Senior Crown SR-71 <http://www.fas.org/irp/program/collect/sr-71.htm>.
- [18] About Edwards – History: First Flights at Edwards Air Force Base [http://www.edwards.af.mil/history/docs\\_html/aircraft/first\\_flights.html](http://www.edwards.af.mil/history/docs_html/aircraft/first_flights.html).
- [19] White paper: “Celestial Augmentation of Inertial Navigation Systems: A Robust Navigation Alternative” U.S. Naval Observatory, Washington, DC (USNO), SPAWAR System Center, San Diego, CA (SPAWARSYSCEN SD).

- 
- [20] NASA Astronomy Picture of the Day, August 17, 1997. Astro-1 In Orbit  
<http://antwrp.gsfc.nasa.gov/apod/ap970817.html>.
- [21] Rakoczy, J. M., and West, M. E., "Magnitude calibration of a fixed head star tracker using Astro-1 flight data" American Institute for Aeronautics and Astronautics Space Programs and Technologies Conference, Huntsville, AL, Mar. 24-27, 1992.
- [22] Anderson, K., "FAR-MST Phase I SBIR Final Report," AeroAstro, Inc., Proprietary, February 16, 2005.
- [23] AeroAstro, Inc., LIST Phase I Proposal, Proprietary.
- [24] AeroAstro, Inc., FAR-MST Phase I Proposal, Proprietary.
- [25] Goodrich Corporation, 2004. HD-1003 Star Tracker Attitude Sensor Datasheet.
- [26] Ball Aerospace and Technologies Corp. CT-633 Star Tracker Datasheet.
- [27] EADS Space. SODERN SED16 Star Tracker Datasheet.
- [28] Galileo Avonica. AA-STR APS Based Autonomous Star Tracker Datasheet.
- [29] AeroAstro, Inc., FAR-MST Phase II Proposal, Proprietary.
- [30] Shalom, E., Alexander, J. W., Stanton, R. H., "Acquisition and Track Algorithms for the ASTROS Star Tracker," *Advances in the Astronautical Sciences*, Vol. 57, February 1985, pp. 375–398.
- [31] Analog Devices, Inc. Embedded Processing & DSP. Blackfin ASDP-BF533 Datasheet.

- 
- [32] Anagnostopoulos, C., "Signal Readout in CID Image Sensors" *IEEE Journal of Solid-State Circuits*, Vol. SC-13, No. 1, February, 1978, pp. 11-15.
- [33] Burke, H. K. and Michon, G. J., "Charge-injection Imaging: Operating Techniques and Performance Characteristics," *IEEE Transactions on Electron Devices*, Vol. ED-23, February, 1976, pp. 189-196.
- [34] CID Technology, Liverpool, NY, "What is a CID?" [http://www.cis.rit.edu/research/CID/a\\_cid\\_is.htm](http://www.cis.rit.edu/research/CID/a_cid_is.htm).
- [35] Wikipedia: Charge-Coupled Device, May 2006 [http://en.wikipedia.org/wiki/Charge-coupled\\_device](http://en.wikipedia.org/wiki/Charge-coupled_device).
- [36] HowStuffWorks: What is the Difference between CCD and CMOS Imaging Sensors in a Digital Camera? <http://electronics.howstuffworks.com/question362.htm>.
- [37] Micron Technology, Inc., 2006. CMOS Image Sensors: CMOS Advantages. <http://www.micron.com/products/imaging/technology/advantages.html>.
- [38] PhotoCourse in Digital Photography, Chapter 1: The Digital Image and the Image Sensor, Section 1.4: CCD and CMOS Image Sensors. <http://209.196.177.41/01/01-04.htm>.
- [39] Fillfactory Image Sensors. IBIS4-1300, 1.3M Pixel Rolling Shutter CMOS Image Sensor Datasheet.
- [40] Fillfactory Image Sensors, November 2003. IBIS5-A-1300, Dual Shutter Mode, 1.3M Pixel, CMOS Image Sensor Datasheet.
- [41] Micron Technology, Inc., 2004. ½-Inch Megapixel CMOS Digital Image Sensor: MT9M001C12STM (Monochrome) Datasheet.

- 
- [42] Micron Technology, Inc., 2004. 1.3-Megapixel CMOS Active-Pixel Digital Image Sensor: MT9M413 Datasheet.
- [43] Kodak, April 2006. Kodak KAC-9638 CMOS Image Sensor Datasheet.
- [44] Kodak, February 2006. Kodak KAC-01301 CMOS Image Sensor Datasheet.
- [45] Micron, CMOS Image Sensors: Lens Selection and Suppliers  
<http://micron.com/products/imaging/technology/lens.html>.
- [46] University of Dayton, Department of Electrical and Computer Engineering: Double Gauss Lens  
<http://www.engr.udayton.edu/faculty/jloomis/eop601/designs/dgauss1/dgauss.html>.
- [47] AeroAstro, Inc., LIST Phase II Proposal, Proprietary.
- [48] Wikipedia: Airy Disc, April 2006. [http://en.wikipedia.org/wiki/Airy\\_disc](http://en.wikipedia.org/wiki/Airy_disc).
- [49] Olympus Microscopy Resource Center: Spring, K. R. "Anatomy of the Microscope – Cutoff Frequency and Airy Disk Size."  
<http://www.olympusmicro.com/primer/java/mtf/airydisksize/>.
- [50] Eric W. Weisstein. "Airy Functions." From *MathWorld*--A Wolfram Web Resource.  
<http://mathworld.wolfram.com/AiryFunctions.html>.
- [51] Wikipedia: Airy Function, April, 2006. [http://en.wikipedia.org/wiki/Airy\\_function](http://en.wikipedia.org/wiki/Airy_function).
- [52] Eric W. Weisstein. "Gaussian Function." From *MathWorld*--A Wolfram Web Resource.  
<http://mathworld.wolfram.com/GaussianFunction.html>.

- 
- [53] McFee, C. "Noise Sources in a CCD." [http://www.mssl.ucl.ac.uk/www\\_detector/opttheory/darkcurrent.html](http://www.mssl.ucl.ac.uk/www_detector/opttheory/darkcurrent.html).
- [54] Quantization Noise. [http://www.class.ese.wustl.edu/ese488/Lectures/Lecture5a\\_QNoise.doc](http://www.class.ese.wustl.edu/ese488/Lectures/Lecture5a_QNoise.doc).
- [55] Eric W. Weisstein. "Central Limit Theorem." From *MathWorld*--A Wolfram Web Resource. <http://mathworld.wolfram.com/CentralLimitTheorem.html>.
- [56] Ju, G., Pollock, T., and Junkins, J., "DIGISTAR: A Low Cost Micro Star Tracker" *AIAA Space Technology Conference and Exposition*, AIAA 99-4603, Albuquerque, NM., September 1999.
- [57] Samaan, M., Mortari, D., Pollock, T., and Junkins, J., "Predictive Centroiding for Single and Multiple FOVs Star Trackers," *AAS/AIAA Space Flight Mechanics Meeting*, San Antonio, TX, Jan. 2002, AAS 02-103.
- [58] Kruijff, M., Heide, E. J., de Boom, C. W., and Heiden, N., "Star Sensor Algorithm Application and Spin-Off," AIC-03-A.5.03.
- [59] Winick, K. "Cramer-Rao bounds on the performance of charge-coupled-devide optical position estimators," *Journal of the Optical Society of America*, Vol. 3, No. 11, November 1986, pp. 1809-1815.
- [60] NASA Sky2000 Master Star Catalog.
- [61] Mortari, D. "Search-Less Algorithm for Star Pattern Recognition," *The Journal of the Astronautical Sciences*, Vol. 45, No. 2, April-June 1997, p. 179-194.

- 
- [62] Gottlieb, D. M., "Star Identification Techniques," *Spacecraft Attitude Determination and Control*, 1978, pp. 259-266.
- [63] Groth, E., "A Pattern-Matching Algorithm for Two-Dimensional Coordinate Lists," *Astronomical Journal*, Vol. 91, 1986, pp. 1244-1248.
- [64] Padgett, C., Kreutz-Delgado, K., and Udomkesmalee, S., "Evaluation of Star Identification Techniques," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 2, March-April 1997, pp. 259-267.
- [65] Heide, E. J., Kruijff, M., Douma, S. R., and Oude Lansink, D., "Development and Validation of a Fast and Reliable Star Sensor Algorithm with Reduced Data Base," *IAF-98.A.6.05*, IAF Melbourne, 1998.
- [66] Cole, C. L., and Crassidis, J. L., "Fast Star Pattern Recognition Using Spherical Triangles," *AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, Providence, RI, Aug. 2004, AIAA Paper #2004-5389.
- [67] Jansen, J. D., "Use of spherical trigonometry to compute near-well flow through irregular grid block boundaries," *Computational Geosciences*, Vol. 6, 2002, pp. 195-206.
- [68] Markley, F. L., "Attitude Determination Using Vector Observations: A Fast Optimal Matrix Algorithm," *The Journal of the Astronautical Sciences*, Vol. 21, No. 2, April-June 1993, pp. 261-280.
- [69] Cole, C. L., and Crassidis, J. L., "Fast Star-Pattern Recognition Using Planar Triangles," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, January-February 2006, pp. 64-71.

- 
- [70] Samaan, M. A., Mortari, D., Junkins, J. L., "Recursive Mode Star Identification Algorithms," AAS Paper No. 01-149, pp. 677-692.
- [71] Mortari, D. et.al. "The Pyramid Star Identification Technique." Texas A&M University, College Station, Texas.
- [72] Crassidis, J. and Markley, F., "A Predictive Attitude Determination Algorithm," *Proceedings of the Flight Mechanics/Estimation Theory Symposium*, NASA-Goddard Space Flight Center, Greenbelt, MD, May 1997, pp. 249-263.
- [73] Wahba, G., "A Least Squares Estimate of Spacecraft Attitude," *SIAM Review*, Vol. 7, No. 3, 1965, p.409.
- [74] Hall, C. Ch 4: Attitude Determination. March 18, 2003.
- [75] Bar-Itzhack, I. "REQUEST: A Recursive QUEST Algorithm for Sequential Attitude Determination," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 5, September-October 1996.
- [76] Keat, J. Analysis of Least Squares Attitude Determination Routine DOAOP," Computer Sciences Corp., CSC/TM-77/6034, Silver Spring, MD, February 1977.
- [77] Rodrigues, M. O., "Des Lois Géométriques Qui Régissent les Déplacement d' un système Solide dans L'espace, et de la Variation des Coordonnées Provenant de ces Déplacements Considérés Independamment des Causes qui Peuvent les Produire," *Journal de Mathematiques Pures et Appliquees (Liouville)*, Vol. 5, 1840, pp. 380-440.
- [78] Markley, F. L., "Attitude Determination using Vector Observations and the Singular Value Decomposition," *The Journal of the Astronautical Sciences*, Vol. 36., No. 3, July-September 1988, pp. 245-258.



- 
- [79] Eric W. Weisstein. "Quaternion." From *MathWorld--A Wolfram Web Resource*.  
<http://mathworld.wolfram.com/Quaternion.html>.
- [80] Stevens, B. L. and Lewis, F. L., *Aircraft Control and Simulation*, 2<sup>nd</sup> ed., Wiley, Hoboken, NJ, 2003.