2016

A geometric hashing technique for star pattern recognition

Joshua Gerhard

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

Recommended Citation

Gerhard, Joshua, "A geometric hashing technique for star pattern recognition" (2016). *Graduate Theses, Dissertations, and Problem Reports*. 5663.
<https://researchrepository.wvu.edu/etd/5663>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

A GEOMETRIC HASHING TECHNIQUE FOR STAR PATTERN RECOGNITION

Joshua Gerhard

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements for the degree of
Master of Science
in
Mechanical Engineering

John A. Christian, Chair Ph.D.

Jason Gross, Ph.D.

Alfred Lynam, Ph.D.

Thomas Evans, Ph.D.

Department of Mechanical and Aerospace Engineering

Morgantown, West Virginia

2016

Keywords: Geometric Hashing, Pattern Recognition, Star Identification

Copyright © 2016

ABSTRACT

A Geometric Hashing Technique for Star Pattern Recognition

Joshua Gerhard

With the present interest in deep space missions by NASA and other space agencies, various techniques are continuously being developed and tested for accurate spacecraft attitude determination. One particularly important problem is the situation where the spacecraft has no *a priori* attitude information — the so called “lost-in-space” problem. This thesis presents a novel method for recognizing star patterns in an image with no *a priori* attitude information, with applications to spacecraft star trackers and other similar attitude sensors. Specifically, a geometric hashing approach is proposed that describes star patterns by the four interior star angles that form the perimeter of a star quad. A technique is presented for labeling these four interior star angles to make the hash code both unique as well as invariant to sensor attitude or star observation order. Using this approach, an index of star quad hash codes is created from a star catalog and the result is stored in a k -d tree. When an image of a star field is obtained, observed star quads are created and the index is searched for the best match using a nearest neighbor algorithm. A verification process is used to improve robustness. This thesis gives an extensive overview of the process that was created to recognize and disregard possible false positives in the star identification algorithm, while still producing accurate results and correctly identifying stars at a high success rate. Performance of the new star identification approach is demonstrated through a Monte Carlo analysis that considered 100,000 random attitudes. It is found that the proposed approach successfully identifies star patterns in 99.79% of the cases and decides that no match can be reliably made in 0.21% of the cases. Incorrect star match were observed in only 0.001% of the test cases. The effect of these very infrequent misidentifications on the resulting attitude estimate were found to be benign.

To all my family and friends

Acknowledgements

First and foremost, I would like to thank God for all that He has blessed me with and all that He continues to bless me with. Without his grace and mercy, I would not be the person I am today and none of this would have been possible. For that, I am forever grateful.

I would like to thank my advisor, Dr. John Christian. Thank you for giving me the opportunity to do research in the field of spacecraft navigation. Thank you for expanding my knowledge to a level I never thought possible and for your constant help and guidance. You are, without a doubt, one of the smartest people I have ever met and truly an expert in the field of spacecraft navigation.

I would like to thank my family. To my parents, Jack and Fawn Gerhard, thank you for all that you have blessed me with. From the never ending love, to the countless prayers, I thank you. Thank you for the support and guidance throughout my entire life. Thank you to my brother, Jeremiah, for all that you do for me. Thank you to my sister, Rachel, and her family for the continuous love and prayers. Thank you to my grandparents, Joan Gerhard, and Lou and Vicki Salerno, for the all the prayers, love and support.

To my girlfriend Jade, thank you for all the support and encouragement over the last few years. Thank you for being there for me through thick and thin, and for the love you have given me, when I least deserved it.

Thank you to the members of the Applied Space Exploration Lab for all the help during my time of research.

Finally, I would like to thank all my friends. Though there are too many to mention, thank you to all who have been there and supported me over the years.

For all my friends and family, I thank you. I am blessed beyond belief and sincerely grateful.

Contents

Abstract	ii
Acknowledgements	iv
List of Figures	vii
List of Tables	ix
Acronyms	x
Nomenclature	xi
1 Introduction	1
1.1 Objective	3
1.2 Contributions	4
1.3 Thesis Organization	4
2 Background	6
2.1 Star Trackers	6
2.2 Star Pattern Algorithms	7
2.3 Bright Star Catalogs	10
2.4 Geometric Hashing	13
2.5 k -d Trees and Nearest Neighbor Searching	14

2.6	Attitude Estimation	16
2.6.1	Measurement Model	16
2.6.2	Maximum-Likelihood Attitude Estimation and the Wahba Problem	18
2.6.3	Derivation of Singular Value Decomposition	22
3	Construction of Star Pattern Index	28
3.1	Selection of Bright Star Catalog	28
3.2	Unit Vectors from star catalog information	28
3.3	Star Angles	30
3.4	Geometric Hashing with Star Quads	32
3.5	Generation of Quad Index from Star Catalog	35
3.5.1	Uniform Sampling of the Celestial Sphere	36
3.5.2	Creation of Measurably Unique Quads	38
3.6	Storing Index in k -d Tree for Fast Searching	39
4	Star Pattern Recognition	40
4.1	Star Measurements in the Sensor Frame	40
4.2	Generating Observed Hash Codes and Searching the Index	42
4.3	Solving Wahba's Problem to Calculate Attitude	43
4.4	Pattern Verification Process	44
5	Performance Assessment via Monte Carlo Analysis	47
5.1	Setup/Index Construction	47
5.2	Simulation	48
5.3	Simulation Results	49
6	Conclusion	55

List of Figures

1.1	The star tracker attitude estimation process broken down into four main steps. This thesis focuses on the star ID step.	2
2.1	Figure to show a star tracker identifying patterns of stars and using star measurements to estimate attitude. Modeled after figure from [1]	7
2.2	Visual representation of the RA and DEC of an arbitrary star location.	11
2.3	3D k -d tree used to help visualize structure of k -d tree	16
3.1	Figure to help visualize how to calculate unit vector measurements from Right Ascension and Declination information. Figure taken from [2] with permission from Authors.	29
3.2	Visualization of brightest stars on the celestial sphere.	30
3.3	Distribution of star magnitudes from the Yale Bright Star Catalog. .	31
3.4	Angle between the unit vector of star i and the unit vector of star j . .	31
3.5	A set of two simple criteria may be used to uniquely assign labels A through D to the four stars in a quad. The first criteria states that stars A and B share the largest interior star angle. The second criteria states the stars A and C share the smallest interior star angle. . . .	33
3.6	Dodecahedron inscribed within a unit sphere that represents a possible polyhedron that can be used to create uniformly spaced vertices. . . .	36

3.7	A spiral on the surface of unit sphere can be used to create k points with near uniform spacing.	38
4.1	Flow chart of the entire star ID process.	41
4.2	A successful star ID requires that one and only one catalog star be within a specified tolerance of the observed star, as shown in the left most frame. If no stars are within the specified tolerance (center frame) or more than one star is within the specified tolerance (right frame) then no star ID is assigned. In all three frames, the solid black dot represents the observed star and the gray dots denote the catalog stars projected onto the image.	45
5.1	Figure showing the spiral of points used to construct the index. The red circles show the location of the pointing directions and the black stars show the locations of bright stars.	48
5.2	Four randomly generated pointing directions. The red circles represent the boresight direction of the camera, the green stars represent stars within the camera FOV, and the blue stars represent the 10 brightest stars in the camera FOV.	50
5.3	Attitude errors in x direction for all 100,000 Monte Carlo cases. . . .	52
5.4	Attitude errors in y direction for all 100,000 Monte Carlo cases. . . .	52
5.5	Attitude errors in z direction for all 100,000 Monte Carlo cases. . . .	53
5.6	Magnitude of the attitude errors for all 100,000 Monte Carlo cases. . .	53
5.7	Cumulative density function of number of quads checked to reach a decision (either confirm a match or to determine that no match is possible).	54

List of Tables

2.1	Table of star measurement information taken from Yale Bright Star catalog [3].	12
2.2	Table of star measurement information taken from Yale Bright Star catalog [3], where the RA and DEC have been converted into radians.	13
5.1	Matching success results of Monte Carlo simulation	49
5.2	Attitude error statistics for Monte Carlo simulation	51

Acronyms

NASA	National Aeronautics and Space Administration
ASEL	Applied Space Exploration Laboratory
WVRTC	West Virginia Robotic Technology Center
WVU	West Virginia University
LIDAR	LIght Detection And Ranging
WFOV	Wide Field Of View
FOV	Field Of View
SVD	Singular Value Decomposition
ID	Identification
RA	Right Ascension
DEC	Declination
PDF	Probability Density Function
MLE	Maximum Likelihood Estimation

Nomenclature

Symbol	Units	Description
α		Right Ascension
δ		Declination
$\mathbf{u}_{i_S}$		i -th Sensor Frame Unit Vector
$\tilde{\mathbf{u}}_{i_S}$		i -th Noisy Sensor Frame Unit Vector
$\mathbf{u}_{i_I}$		i -th Inertial Frame Unit Vector
$\mathbf{T}_S^I$		Rotation Matrix from Inertial to Sensor Frame
ϵ		Measurement Error
$E[\bullet]$		Expected Value Operator
$\delta\mathbf{T}_i$		Rotational Error Matrix
$\delta\phi$		Angle Vector of Rotation Error
R_i		Measurement Covariance
σ^2		Variance
tr		Trace Operator
$\mathbf{B}$		Attitude Profile Matrix

Chapter 1

Introduction

With the emergence of new privately owned companies and research groups within the last few years, space exploration has experienced substantial growth in research and development. These interests have brought forth the determination to make advances and new discoveries in the space industry that were once thought impossible.

Currently, the Applied Space Exploration Lab (ASEL) at West Virginia University (WVU), led by Dr. John A. Christian, is making great strides in the area of spacecraft navigation. The ASEL frequently collaborates in this area with the West Virginia Robotic Technology Center (WVRTC). With research being conducted on camera-based and LIDAR-based optical navigation as well as attitude determination, the ASEL has helped make WVU an ever growing competitor in space navigation research [2, 4–12]. One problem of particular interest in the area of spacecraft navigation is known as the “lost-in-space” problem. The lost-in-space problem can basically be categorized as solving for spacecraft attitude when no *a priori* information is available. Although there are various techniques used to solve the lost-in-space problem, perhaps one of the most typical ways is the star tracker method.

The typical lost-in-space attitude estimation process for star trackers can be broken down into a few primary steps, as shown in Fig. 1.1.

Figure 1.1: The star tracker attitude estimation process broken down into four main steps. This thesis focuses on the star ID step.

The attitude estimation process typically begins by recording an image of stars in a given sensor frame. After the image is taken, a centroiding algorithm is used to generate measurement information for candidate stars expressed in the camera’s sensor frame (these are called *candidate stars* because they have not yet been identified and some centroids may not actually be stars). Next, a star pattern identification process is used to match the observed candidate stars in the image to a database of predetermined star patterns. Finally, the observed candidate stars and the predetermined star patterns can be used to calculate the estimated spacecraft attitude.

In short, the star pattern identification process for star trackers can be used with no *a priori* information to solve for the spacecraft attitude, by recording measurements based on the locations of stars in a sensor frame, generating a pattern from the recorded measurement information, searching through a predetermined index of possible pattern matches, and determining an approximate attitude estimate based

on the best matched pattern information. Assuming the correct stars were matched, the computed attitude is thus the solution to the lost-in-space problem.

The star tracker method to solve the lost-in-space problem is constantly being researched to determine new and better techniques that produce more accurate results. The primary focuses of this thesis will be on the star pattern identification process.

1.1 Objective

The main objective of this thesis will be to investigate and develop a technique that solves the lost-in-space problem with no *a priori* knowledge by determining the attitude of a spacecraft using a star tracker that is integrated with an intelligently designed pattern recognition algorithm.

This thesis presents a star ID algorithm that matches observed stars in an image to known stars in a catalog, while also identifying outliers that are not stars. We focus on the situation where no *a priori* attitude information is available — the so called “lost-in-space” problem — in order to enable reliable initialization or star tracking with high angular velocity. The key to any lost-in-space star ID algorithm is to create a star pattern description that is attitude invariant. Consequently, this thesis considers methods for (1) constructing a database of attitude invariant pattern descriptors and then (2) querying this database to find the best match to an observed pattern. We further aim to build a pattern descriptor and structure the database in a manner that permits a fast search with a very low probability of producing a false positive (an incorrect star pattern match).

Though the use of star trackers integrated with pattern recognition algorithms have been previously investigated and developed, there are numerous techniques with their own specific features and results. The objective of this thesis is to not only correctly identify patterns of stars at a high success rate and accurately calculate

attitude using the correctly identified stars, but also be able to identify cases where stars could be falsely identified.

With the objective in mind, this thesis will give an in depth explanation of how these goals were achieved. Along with this in depth explanation, a Monte Carlo Simulation will be used to show the performance of the developed star pattern recognition technique.

1.2 Contributions

This thesis presents a new star ID algorithm for the lost-in-space attitude determination problem. The new algorithm presented here makes use of a novel pattern descriptor based on the concept of geometric hashing. Although this research is primarily designed with the intent to be integrated with a star tracker to successfully solve the lost-in-space problem for spacecraft attitude determination, it could also be used for other applications as well. With some slight alterations, this algorithm could also be used to identify star patterns for a manual attitude reinitialization device [2].

Though this research is intended for applications in the space industry, it is not limited to being only used for spacecraft attitude determination. The pattern recognition process developed in this thesis by creating geometric hash codes based on known point measurement information, could be used in other pattern recognition applications.

1.3 Thesis Organization

This thesis is organized in the following manner: Chapter 2 will give an in depth look at the background that will aid in a better understanding of the material presented in this thesis. It will begin with a discussion on star trackers and how they are used to determine spacecraft attitude. This will lead into a literature review of previously

developed star pattern recognition algorithms, followed by a discussion about star catalogs, Geometric Hashing, k -d trees, nearest neighbor searching, and finally ending with the derivation of the attitude estimation process. Chapter 3 will explain how an intelligently designed index of patterns can be constructed to be used for star pattern recognition. It will begin by explaining how bright star catalogs can be used to obtain useful star information, followed by a detailed discussion of Hash Code construction, techniques that can be used to generate an index in a near uniform manner, and end with the use of k -d trees to store and search the predetermined index. Chapter 4 will explain the pattern recognition technique and decision process developed to correctly identify star patterns. Chapter 5 will explain the setup and show the results from a Monte Carlo Simulation based on the proposed research. Chapter 6 will be a conclusion of this thesis.

Chapter 2

Background

2.1 Star Trackers

Star trackers are a common spacecraft attitude sensor that use observations of known stars to estimate the vehicle's orientation [13, 14]. Typical star trackers are optical sensors that work by collecting an image of a star field, matching the observed star pattern to an onboard catalog of known stars, and then computing the attitude that best aligns the observed star directions (in the sensor frame) with the catalog star directions (in the inertial frame). Thus, a critical step in star tracker operation is fast and robust star identification (ID). Fig. 2.1 gives a visual representation of the general process explained above on how star trackers work, and this process will be further explained.

Though occasionally a star tracker will have a WFOV, most have a relatively small FOV, typically ranging between $10 - 20^\circ$. Because of this, the number of stars that appear in an image taken from a star tracker is usually rather low.

Once a star tracker records an image, the next step is to find the locations of stars within the image. This process is typically done through use of a star centroiding algorithm and there are multiple well-known techniques [15–17]. After the centroid

Figure 2.1: Figure to show a star tracker identifying patterns of stars and using star measurements to estimate attitude. Modeled after figure from [1]

pixel coordinate is determined from the centroiding algorithm, a calibration process is used to determine the sensor frame line-of-sight directions to each star. One particular method of this process has been proposed by [11].

With the now known observed star measurements in hand, patterns within these measurements are computed. There are various techniques that have been developed for computing patterns of observed stars that are matched to an index of predetermined patterns, and some of these techniques will be discussed later in this thesis. Once the observed patterns are generated, a querying of a predetermined pattern index is conducted. Typically a searching algorithm is used to quickly search the predetermined index for a correct match. Once an observed pattern is matched to a predetermined indexed pattern and a correct match of the stars has been made, the attitude of the spacecraft can be determined using the observed and cataloged star measurements.

2.2 Star Pattern Algorithms

Star trackers have been used for attitude determination for decades. Consequently, there are numerous techniques that have been proposed to construct and search star

patterns from a catalog [18–32]. Though all star identification algorithms share the same final goal of successfully identifying stars, the approaches taken to reach this goal can be significantly different. Thus, a brief overview of the more well known star identification methods is now presented.

In 1977, Junkins *et al.* [18] developed a routine that uses a pattern recognition technique along with an extended Kalman filter to estimate an improved attitude estimate based on *a priori* knowledge at each epoch.

Groth [19] developed a pattern matching algorithm in 1986 that uses triangles formed from various point triplets to generate two lists. These lists can then be used to match observed triangles that share pairs of coordinates in each list.

Liebe [22] begins by using the brightest stars from a selected star catalog, constructing an index by selecting a reference star, and then storing the distance between the two nearest neighboring stars. The angle between the first and second neighboring stars is then calculated and stored. This process is done for each referenced star and the index can then be searched to correctly identify stars.

The grid algorithm presented by Padgett and Kreutz-Delgado [27] begins by generating patterns of groups of stars and then selecting a reference star to be the center of the grid. Stars considered in a particular pattern must fall within a predetermined pattern radius. The grid alignment is determined based on the closest neighbor to the reference star. Once the grid is reoriented into the correct alignment, the star locations are used to construct a pattern descriptor (in the form of a vector) that can be used for star identification.

The Pyramid algorithm developed by Mortari *et al.* [26] identifies star patterns through the use of unique triangles composed of three stars. Once a unique triangle is identified, an attempt is made to match a fourth star to increase confidence in the star IDs. Mortari *et al.* use a predetermined k-vector to initially search for a unique three star triangle [23, 24, 33]. The purpose of the k-vector approach is to eliminate

the need for searching the entire index.

In the early 2000's, Cole and Crassidis investigated an approach using three stars to construct a three star spherical triangle [28]. The area and polar moment information regarding each spherical triangle is stored in a catalog for matching. Cole and Crassidis later investigated a related approach, where planar triangles were used instead of spherical triangles [29]. The planar triangle method was found to produce better results.

Kolomenkin *et al.* [30] developed a star identification algorithm known as the geometric voting algorithm. This approach forms a catalog of inter-star angles and then uses a voting scheme based on the measured angular separation between star pairs. Every star pair in the image is considered and, if an observed star pair matches a pair in the catalog to within a specified tolerance, each observed star is given a vote for both of the corresponding star IDs in the catalog. Once all the votes are acquired, a clustering algorithm is used to verify if a particular star is correctly identified by comparing the number of votes relative to the maximum number of star votes.

McBryde and Lightsey [32] investigated using different pattern recognition techniques as a means of attitude determination for CubeSats. One of the main objectives was to determine the best star identification algorithm already developed to be used with their star centroiding method. McBryde and Lightsey give a general overview and comparison of four well known star identification algorithms as possible candidates to be implemented. These included the Planar Triangle star identification technique [29], Voting method [30], Pyramid method [26], and Grid algorithm [27].

Pham *et al.* [31] describe a method that uses only the brightest stars from the selected star catalog. Training patterns are constructed by centering a reference star in the boresight direction and then projecting stars onto an image plane. The data is then stored in an index structured first by the number of neighboring stars in the image, followed by the planar distance to each neighboring star. This data forms

a new index that is stored in a tree structure for faster searching. It is mentioned that the framework for Pham *et al.* [31] was modeled after other star identification algorithms [22, 25, 30].

Finally, although not used for star trackers, Lang *et al.* [34] introduce a star identification algorithm for the automatic alignment of astronomical images. Their algorithm constructs a geometric hash code using star patterns projected onto the image that have been normalized by the distance between the two furthest stars in the pattern. This creates a normalized coordinate frame in which the remaining star coordinates may be expressed. Clusters of four stars are then used to generate geometric hash codes that may be efficiently searched.

Thus, we find that there are numerous star identification algorithms, all implementing different methods for star matching and all having varying statistical robustness and computational speeds. The primary concern of this thesis is to not only be able to correctly identify stars and generate an accurate attitude, but to also identify the cases where an incorrect solution is possible. By using the angles associated with four stars, or “quads,” to construct geometric hash codes, we can successfully identify stars and also identify cases where a false positive could occur.

2.3 Bright Star Catalogs

Star trackers determine attitude by aligning the observed stars in an image with a catalog of known star directions. Thus catalog selection and preprocessing is a critical first step. This catalog contains known star information in a given reference frame that can be used to construct an index that is created based on frame invariant characteristics of stars.

There are numerous star catalogs available that can be integrated into a star tracker, depending on the pattern recognition algorithm requirements [3, 35–39]. Each

of these catalogs contain various information that ranges from the number of stars, to the reference frame that was used to construct the catalog. Though the various catalogs range in size and recorded star information, typically the star ID, Magnitude, Right Ascension, and Declination of each star is presented in the catalog.

The star ID is simply the identification given to each star in the catalog. A star's ID is typically assigned in ascending order starting with the first cataloged star. For example, the Yale Bright Star Catalog [3] orders stars based on their location about the Right Ascension.

Star Magnitude is a unitless number that represents how bright a star appears. This number is defined such that smaller star magnitudes, correspond to brighter stars. For example a star with a magnitude of 6 will appear dimmer than a star with a magnitude of 5. Star catalogs are usually constructed by only recording measurement information of stars that have a magnitude less than some threshold value.

Figure 2.2: Visual representation of the RA and DEC of an arbitrary star location.

The Right Ascension (RA) and Declination (DEC) are used to describe the ce-

lestial location of a particular star. From Fig. 2.2 it can be seen that the RA is measured as the eastward angle —along the equator of the celestial sphere— from the vernal equinox, while DEC is measured as the angle above or below the equator of the celestial sphere [40].

Generally, RA is measured in *Hours, Minutes, Seconds* ($hh:mm:ss.s$) and DEC is measured in *Degrees, Minutes, Seconds* ($dd^\circ:mm:ss.s$). Table. 5.2 shows how the RA and DEC information is presented in a star catalog.

Table 2.1: Table of star measurement information taken from Yale Bright Star catalog [3].

Star ID	Right Ascension	Declination	Magnitude
1	00 : 05 : 09.9	+45 : 13 : 45	6.7
2	00 : 05 : 03.8	−00 : 30 : 11	6.29
3	00 : 05 : 20.1	−05 : 42 : 27	4.61
4	00 : 05 : 42.0	+13 : 23 : 46	5.51
5	00 : 05 : 42.0	+58 : 26 : 12	5.96

RA is defined on the interval $[0, 360)$ deg, which is equivalent to $[0, 24)$ hrs. Thus, 1 hr of arc is equivalent to 15 deg. Each hour of arc is divided into 60 minutes, and each minute of arc is divided into 60 seconds. Consequently, the RA in degrees may be computed from the $hh:mm:ss.s$ format by

$$\begin{aligned}
 RA[deg] = & \left(15 \frac{deg}{hr}\right) hh + \left(15 \frac{deg}{hr}\right) \left(\frac{1hr}{60min}\right) mm \\
 & + \left(15 \frac{deg}{hr}\right) \left(\frac{1hr}{60min}\right) \left(\frac{1min}{60sec}\right) ss.s
 \end{aligned} \tag{2.1}$$

This RA may easily be converted from degrees to radians if desired.

DEC is defined differently then RA. DEC is defiend on the interval $[-90, 90]$ deg. Each degree is divided into 60 minutes (sometimes called *arcminutes*), and each arcminute is divided into 60 seconds (sometimes called *arcseconds*).

Table. 2.2 shows the same information as Table. 2.1, only with the RA and DEC

converted into radians.

Table 2.2: Table of star measurement information taken from Yale Bright Star catalog [3], where the RA and DEC have been converted into radians.

Star ID	Right Ascension	Declination	Magnitude
1	0.0225	0.7894	6.7
2	0.0221	−0.0088	6.29
3	0.0233	−0.0996	4.61
4	0.0249	0.2338	5.51
5	0.0273	1.0199	5.96

2.4 Geometric Hashing

Geometric Hashing is a technique that was developed for use in computer vision. The process is explained in more depth in [41]. In short, the process of Geometric Hashing is simply a pattern recognition technique that can identify geometric features despite their orientation and relative position. This makes Geometric Hashing a very robust way to identify star patterns, since a particular pattern can be arbitrarily rotated and still be matched.

The process of Geometric Hashing begins by selecting an arrangement of points that form a pattern that represents a geometric feature. Using point pairs to form a basis, the pattern of points is structured in such a way that is invariant to transformations. This newly structured pattern is then used to describe the particular geometric feature it is formed from.

Once the transformation invariant patterns are formed, they can be stored in a predetermined index that contains the basis and pattern information, where the size of the index is directly dependent on the number of desired geometric features to be identified. Once an index is formed from the predetermined transformation invariant patterns, an observed geometric feature’s point pattern is searched within the index

for a correctly identified indexed pattern match. If an observed pattern is correctly matched to an indexed pattern, the observed geometric feature can be identified using the observed and indexed pattern information.

This Geometric Hashing approach for pattern recognition can be integrated with Star Trackers for star pattern recognition. When it comes to constructing patterns that represent geometric star features, a frame invariant characteristic must first be determined. The process for constructing patterns of transformation invariant characteristics is a crucial step for the Geometric Hashing process to be applicable. Because the angle between any random star pair remains the same, whether measured in the inertial frame or sensor frame, it is a transformation invariant characteristic that can be used in a Geometric Hashing approach to identify star patterns.

For every desired pattern of stars, the transformation invariant angular information between the star pairs within the pattern can be stored in the index. This stored information that is constructed in order to represent a pattern of stars is known as a Hash Code. The number of Hash Codes within the index depends on the number of star patterns to be identified. A crucial step in the index construction process, is to design the Hash Codes in a uniquely distinguishable way such that a correct pattern can be correctly identified when sensor noise is present.

Once an index of Hash Codes representing star patterns is constructed, it must be stored in such a way that an observed pattern can be quickly searched for a correct pattern match.

2.5 k -d Trees and Nearest Neighbor Searching

The index of Hash Codes will be large and contain thousands of Hash Codes that represent star quads. Because of this, the index must be stored in such a way that an efficient and accurate search of the index be conducted for an observed pattern

match. One particular way to satisfy these requirements is to store the Hash Code data index in a k -d Tree [42].

K -d Trees are binary trees that store data in a k -dimensional space. They are structured such that each data set –in this case each Hash Code– is stored as a point within the k dimensional space. The dimensionality of the space is dependent on the complexity of the points. For example, points represented by x , y , and z coordinates would be stored in a $3D$ K -d tree.

It can be seen from Fig. 2.3 that k -d trees split the data using multiple hyperplanes. The location and direction of the hyperplanes are based on the density of points within each split subspace. Points that are very close to one another are typically grouped together. These hyperplanes separate the points in such a way that the entire k -dimensional space does not need to be searched, but the search can be limited to smaller subspaces. K -d trees have a building complexity on the order of $O(n \log n)$. Though this is not particularly fast, the tree is built off-line and only needs to be constructed once.

Perhaps one of the best ways to search through a k -d tree of stored data is via a nearest neighbor search [43]. The nearest neighbor search algorithm begins at the top of the tree and follows the partitions down the various levels –or branches– until the best matched point is found. With every partition, more and more points can be excluded from the search. This leads to a near real time searching scheme that has complexity on the order of $O(\log n)$. The robust and efficient nature of storing data in a k -d tree and searching the tree using a nearest neighbor search algorithm is what makes this process very beneficial to use in the proposed star pattern recognition algorithm.

Figure 2.3: 3D k -d tree used to help visualize structure of k -d tree

2.6 Attitude Estimation

This section reviews the key components of the classic attitude determination from unit vectors problem.

2.6.1 Measurement Model

Once an observed pattern of stars in the sensor frame is matched to an indexed pattern of stars in the inertial frame, the attitude estimation process can be conducted. Using the now known sensor frame and inertial frame star information, the following measurement model can be considered.

$$\mathbf{u}_{i_S} = \mathbf{T}_S^I \mathbf{u}_{i_I} \quad (2.2)$$

Where $\mathbf{T}_S^I$ is the rotation matrix from the inertial frame to the sensor frame, $\mathbf{u}_{i_S}$ is the i -th unit vector measurement in the sensor frame, and $\mathbf{u}_{i_I}$ is the i -th unit vector measurement in the inertial frame.

Due to measurement noise experienced when recording observed sensor frame star unit vectors, the noisy observed sensor frame unit vector measurements can be written as

$$\tilde{\mathbf{u}}_{i_S} = \mathbf{u}_{i_S} + \epsilon \quad (2.3)$$

where ϵ is the measurement error due to noise. Assuming an unbiased measurement, the expected value of the measurement error is

$$E[\epsilon] = 0 \quad (2.4)$$

The new noisy measurement model can now be written as the following

$$\tilde{\mathbf{u}}_{i_S} = \mathbf{T}_S^I \mathbf{u}_{i_I} + \epsilon \quad (2.5)$$

This can be rearranged and solved for ϵ

$$\epsilon = \tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I} \quad (2.6)$$

We note, however, that $\mathbf{u}_{i_I}$, $\mathbf{u}_{i_S}$, and $\tilde{\mathbf{u}}_{i_S}$ are all constrained to have unity norm (since they're both unit vectors). This places a constraint on ϵ . Thus, to explicitly conserve the length of $\tilde{\mathbf{u}}_{i_S}$, we can describe the measurement error as a small rotation

$$\tilde{\mathbf{u}}_{i_S} = \boldsymbol{\delta} \mathbf{T}_i \mathbf{u}_{i_S} \quad (2.7)$$

where $\boldsymbol{\delta} \mathbf{T}$ is a rotation matrix describing the error in the measured direction of $\mathbf{u}_{i_S}$

and can be written in terms of the matrix exponential as

$$\boldsymbol{\delta T} = \exp([- \boldsymbol{\delta \phi} \times]) \quad (2.8)$$

where $\boldsymbol{\delta \phi}$ is the rotation error expressed as an angle vector.

If small angular error is assumed, the rotational error matrix $\boldsymbol{\delta T}$ can be estimated to first order using a Taylor series expansion and the higher order terms can be ignored

$$\boldsymbol{\delta T} \approx \mathbf{I}_{3 \times 3} + [- \boldsymbol{\delta \phi} \times] \quad (2.9)$$

Substituting Eq. 2.9 into Eq. 2.7, it can be seen that

$$\tilde{\mathbf{u}}_{i_S} = (\mathbf{I}_{3 \times 3} + [- \boldsymbol{\delta \phi} \times]) \mathbf{u}_{i_S} \quad (2.10)$$

Expanding Eq. 2.10 and using the cross operator yields

$$\tilde{\mathbf{u}}_{i_S} = \mathbf{u}_{i_S} + [\mathbf{u}_{i_S} \times] \boldsymbol{\delta \phi} \quad (2.11)$$

from which we find that

$$\epsilon = [\mathbf{u}_{i_S} \times] \boldsymbol{\delta \phi} \quad (2.12)$$

2.6.2 Maximum-Likelihood Attitude Estimation and the Wahba Problem

Now that the an accurate measurement model has been determined, a maximum-likelihood estimation (MLE) problem is used to solve for the optimal rotation matrix.

The process begins with the following Gaussian probability density function (PDF)

$$p(\mathbf{T}_S^I | \tilde{\mathbf{u}}_{i_S}) = \frac{1}{2\pi^{\frac{n}{2}} \|\mathbf{R}_i\|^{\frac{1}{2}}} \exp \left[-\frac{1}{2} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T \mathbf{R}_i^{-1} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \right] \quad (2.13)$$

where we define the probability of observing the optimal rotation matrix $\mathbf{T}_S^I$, given the observations $\tilde{\mathbf{u}}_{i_S}$.

We now define the constant D_i

$$D_i = \frac{1}{2\pi^{\frac{n}{2}} \|\mathbf{R}_i\|^{\frac{1}{2}}} \quad (2.14)$$

Thus, Eq. 2.13 can be simplified into

$$p(\mathbf{T}_S^I | \tilde{\mathbf{u}}_{i_S}) = D_i \exp \left[-\frac{1}{2} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T \mathbf{R}_i^{-1} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \right] \quad (2.15)$$

Because each observation is independent, the probability of obtaining $\mathbf{T}_S^I$ given $\tilde{\mathbf{u}}_{i_S}$ is

$$p(\mathbf{T}_S^I | \tilde{\mathbf{u}}_{1_S}, \tilde{\mathbf{u}}_{2_S}, \dots, \tilde{\mathbf{u}}_{n_S}) = \prod_{i=1}^n p(\mathbf{T}_S^I | \tilde{\mathbf{u}}_{i_S}) \quad (2.16)$$

The PDF can now be written as

$$\prod_{i=1}^n p(\mathbf{T}_S^I | \tilde{\mathbf{u}}_{i_S}) = D \exp \left[-\frac{1}{2} \sum_{i=1}^n (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T \mathbf{R}_i^{-1} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \right] \quad (2.17)$$

where D is simply

$$D = \prod_{i=1}^n D_i \quad (2.18)$$

We now wish to maximize the likelihood of obtaining the optimal rotation matrix given the observations

$$\max p(\mathbf{T}_S^I | \tilde{\mathbf{u}}_{i_S}) = D \exp \left[-\frac{1}{2} \sum_{i=1}^n (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T \mathbf{R}_i^{-1} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \right] \quad (2.19)$$

It can be seen that maximizing $p(\mathbf{T}_S^I | \tilde{\mathbf{u}}_{i_S})$ will yield the same solution as maximizing $\ln(p(\mathbf{T}_S^I | \tilde{\mathbf{u}}_{i_S}))$. This leads to the following objective function, where the natural logarithm has been distributed

$$\max J(\mathbf{T}_S^I) = \ln(D) - \frac{1}{2} \sum_{i=1}^n (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T \mathbf{R}_i^{-1} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \quad (2.20)$$

We can now write the maximization problem as the following minimization problem, where the first term has been dropped because it is not dependent on the state

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T \mathbf{R}_i^{-1} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \quad (2.21)$$

It is known from Eq. 2.6 that

$$\epsilon = \tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I} \quad (2.22)$$

Eq. 2.22 can be substituted into Eq. 2.21 to simplify

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n \epsilon_i^T \mathbf{R}_i^{-1} \epsilon_i \quad (2.23)$$

In order to further simplify the MLE problem, the next step is to compute the

measurement covariance. The measurement covariance is found using the following

$$\mathbf{R} = E[(\tilde{\mathbf{u}}_{i_S} - E[\tilde{\mathbf{u}}_{i_S}])(\tilde{\mathbf{u}}_{i_S} - E[\tilde{\mathbf{u}}_{i_S}])^T] \quad (2.24)$$

Expanding out and simplifying results in the following

$$\mathbf{R} = E[\epsilon\epsilon^T] \quad (2.25)$$

Substituting Eq. 2.12 into measurement covariance equation

$$\mathbf{R} = E[[\mathbf{u}_{i_S} \times] \boldsymbol{\delta\phi} \boldsymbol{\delta\phi}^T [\mathbf{u}_{i_S} \times]^T] \quad (2.26)$$

Distributing the expected value operator and simplifying

$$\mathbf{R} = -[\mathbf{u}_{i_S} \times] E[\boldsymbol{\delta\phi} \boldsymbol{\delta\phi}^T] [\mathbf{u}_{i_S} \times] \quad (2.27)$$

If it is assumed that there is no correlation between measurements

$$E[\boldsymbol{\delta\phi} \boldsymbol{\delta\phi}^T] = \sigma_\phi^2 \mathbf{I}_{3 \times 3} \quad (2.28)$$

Substituting Eq. 2.28 back into Eq. 2.27 and simplifying

$$\mathbf{R} = -\sigma_\phi^2 (\mathbf{I}_{3 \times 3} - \mathbf{u}_{i_S} \mathbf{u}_{i_S}^T) \quad (2.29)$$

Because the measured covariance matrix $\mathbf{R}$ is only of rank 2 and not full rank, the inverse can not be computed. Due to this fact, the pseudo-inverse must be used in the derivation in place of the inverse. The pseudo-inverse can be found using the following

$$\mathbf{R}^\dagger = \frac{1}{\sigma_\phi^2} (\mathbf{I}_{3 \times 3} - \mathbf{u}_{i_S} \mathbf{u}_{i_S}^T) \quad (2.30)$$

Eq. 2.30 can now be substituted into Eq. 2.23

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n \epsilon_i^T \left[\frac{1}{\sigma_\phi^2} (\mathbf{I}_{3 \times 3} - \mathbf{u}_{i_S} \mathbf{u}_{i_S}^T) \right] \epsilon_i \quad (2.31)$$

Because the error is orthogonal to the unit vector measurements, meaning

$$\mathbf{u}_{i_S}^T \epsilon_i = 0 \quad (2.32)$$

Eq. 2.31 can be expanded and then simplified into

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n \frac{1}{\sigma_\phi^2} \epsilon_i^T \epsilon_i \quad (2.33)$$

Substituting back in for ϵ from Eq. 2.22

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n \frac{1}{\sigma_\phi^2} (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I})^T (\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}) \quad (2.34)$$

Eq. 2.34 is typically written as follows

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n \frac{1}{\sigma_\phi^2} \|\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}\|^2 \quad (2.35)$$

This equation is more commonly known as the Wahba Problem [44]. It is used to solve for the optimal rotation matrix that best matches vector observations between two different coordinate frames.

2.6.3 Derivation of Singular Value Decomposition

There are many proposed solutions to solving Wahba's problem [45–50]. For this thesis, it was decided to use the singular value decomposition (SVD) method originally presented by Markley [46]. It is important to note that the following equations are modeled after [46].

The SVD process begins with the equation for the Wahba Problem

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n \frac{1}{\sigma_\phi^2} \|\tilde{\mathbf{u}}_{i_S} - \mathbf{T}_S^I \mathbf{u}_{i_I}\|^2 \quad (2.36)$$

This equation can be expanded into the following

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n \frac{1}{\sigma_\phi^2} (\tilde{\mathbf{u}}_{i_S}^T \tilde{\mathbf{u}}_{i_S} - \tilde{\mathbf{u}}_{i_S}^T \mathbf{T}_S^I \mathbf{u}_{i_I} - \mathbf{u}_{i_I}^T \mathbf{T}_S^{I^T} \tilde{\mathbf{u}}_{i_S} + \mathbf{u}_{i_I}^T \mathbf{T}_S^{I^T} \mathbf{T}_S^I \mathbf{u}_{i_I}) \quad (2.37)$$

Because the measurements are all unit vectors and we know that the dot product of a unit vector with itself is equal to one, Eq. 2.37 can be simplified into

$$\min J(\mathbf{T}_S^I) = \frac{1}{2} \sum_{i=1}^n \frac{1}{\sigma_\phi^2} (2 - 2\tilde{\mathbf{u}}_{i_S}^T \mathbf{T}_S^I \mathbf{u}_{i_I}) \quad (2.38)$$

and then further simplified into

$$\min J(\mathbf{T}_S^I) = \lambda_o - \sum_{i=1}^n \frac{1}{\sigma_\phi^2} \tilde{\mathbf{u}}_{i_S}^T \mathbf{T}_S^I \mathbf{u}_{i_I} \quad (2.39)$$

where

$$\lambda_o = \sum_{i=1}^n \frac{1}{\sigma_\phi^2} \quad (2.40)$$

Because the right hand term in Eq. 2.39 is a scalar, applying the trace operator will not change the result,

$$\min J(\mathbf{T}_S^I) = \lambda_o - \sum_{i=1}^n \frac{1}{\sigma_\phi^2} \text{tr}[\tilde{\mathbf{u}}_{i_S}^T \mathbf{T}_S^I \mathbf{u}_{i_I}] \quad (2.41)$$

This can be rewritten with the variables within the trace operator cyclically permuted. Also, the rotation matrix $\mathbf{T}_S^I$ can be removed from the summation because it is not dependent on the particular line of sight being considered (it is not dependent

on i)

$$\min J(\mathbf{T}_S^I) = \lambda_o - \text{tr}[\mathbf{T}_S^I \sum_{i=1}^n \frac{1}{\sigma_\phi^2} \mathbf{u}_{i_I} \tilde{\mathbf{u}}_{i_S}^T] \quad (2.42)$$

Define the attitude profile matrix as

$$\mathbf{B} = \sum_{i=1}^n \frac{1}{\sigma_\phi^2} \tilde{\mathbf{u}}_{i_S} \mathbf{u}_{i_I}^T \quad (2.43)$$

Thus we can rewrite the now simplified minimization equation as

$$\min J(\mathbf{T}_S^I) = \lambda_o - \text{tr}[\mathbf{T}_S^I \mathbf{B}^T] \quad (2.44)$$

In order to solve this minimization problem for the optimal solution of $\mathbf{T}_S^I$, the attitude profile matrix $\mathbf{B}$ must be decomposed using Singular Value Decomposition (SVD). This process begins by defining $\mathbf{B}$ as

$$\mathbf{B} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (2.45)$$

where the orthonormal eigenvectors of $\mathbf{B} \mathbf{B}^T$ form the columns of matrix $\mathbf{U}$, and the orthonormal eigenvectors of $\mathbf{B}^T \mathbf{B}$ form the columns of matrix $\mathbf{V}$. It is also defined that the singular values of $\mathbf{B}$ form the diagonal matrix $\mathbf{S}$. Because the matrix $\mathbf{V}$ is constructed from the orthonormal eigenvectors, it is defined as an orthogonal matrix, where

$$\mathbf{V}^T \mathbf{V} = \mathbf{V} \mathbf{V}^T = \mathbf{I}_{3 \times 3} \quad (2.46)$$

Thus, a $\mathbf{V}^T \mathbf{V}$ may be inserted into Eq. 2.45 without changing anything

$$\mathbf{B} = \mathbf{U} \mathbf{V}^T \mathbf{V} \mathbf{S} \mathbf{V}^T \quad (2.47)$$

The matrix $\mathbf{U}$ and $\mathbf{V}^T$ can be combined to form the orthogonal matrix $\mathbf{A}$, and $\mathbf{V} \mathbf{S} \mathbf{V}^T$ can be combined to form the positive semi-definite matrix $\mathbf{C}$. It is also impor-

tant to note that $\mathbf{C}$ will also be a symmetric matrix. This then simplifies the attitude profile matrix into

$$\mathbf{B} = \mathbf{A}\mathbf{C} \quad (2.48)$$

Eq. 2.48 can now be substituted back into Eq. 2.44 to obtain

$$\min J(\mathbf{T}_S^I) = -tr[\mathbf{T}_S^I \mathbf{C}^T \mathbf{A}^T] \quad (2.49)$$

where λ_o can be ignored because it is not dependent on the optimal rotation matrix $\mathbf{T}_S^I$.

$\mathbf{V}\mathbf{S}\mathbf{V}^T$ is now substituted in for $\mathbf{C}$, and the values within the trace operator can once again be cyclically permuted until the following equation is formed

$$\min J(\mathbf{T}_S^I) = -tr[\mathbf{V}^T \mathbf{A}^T \mathbf{T}_S^I \mathbf{V} \mathbf{S}] \quad (2.50)$$

The next step in the SVD process is to convert the minimization problem into a maximization problem. This can be done by simply multiplying the minimization objective function by -1 . This changes Eq. 2.50 into the following maximization problem

$$\max J(\mathbf{T}_S^I) = tr[\mathbf{V}^T \mathbf{A}^T \mathbf{T}_S^I \mathbf{V} \mathbf{S}] \quad (2.51)$$

In order to maximize $J(\mathbf{T}_S^I)$, we begin by defining

$$\mathbf{X} = \mathbf{V}^T \mathbf{A}^T \mathbf{T}_S^I \mathbf{V} \quad (2.52)$$

Thus, we obtain the following

$$\max J(\mathbf{T}_S^I) = tr[\mathbf{X}\mathbf{S}] = \sum_{i=1}^3 \mathbf{X}_{ii} \mathbf{S}_i \quad (2.53)$$

where

$$\sum_{i=1}^3 \mathbf{X}_{ii} \mathbf{S}_i = \mathbf{X}_{11} \mathbf{S}_1 + \mathbf{X}_{22} \mathbf{S}_2 + \mathbf{X}_{33} \mathbf{S}_3 \quad (2.54)$$

From Eq. 2.52, it is observed that $\mathbf{X}$ is an orthogonal matrix. Because $\mathbf{X}$ is orthogonal, the rows and columns each sum to 1. Also, because the matrix $\mathbf{B}$ is positive definite, $\mathbf{S}_i \geq 0$. From this, it can be seen in Eq. 2.53 and Eq. 2.54 that in order to maximize $J(\mathbf{T}_S^I)$ we must set $\mathbf{X}_{ii} = 1$ and $\mathbf{X}_{ij} = 0$.

The next step is to ensure that the resulting estimate of $\mathbf{T}_S^I$ is right-handed. This is done by taking the determinate of $\mathbf{X}$. Where

$$\det(\mathbf{X}) = \det(\mathbf{V}^T) \det(\mathbf{A}^T) \det(\mathbf{T}_S^I) \det(\mathbf{V}) \quad (2.55)$$

or simply

$$\det(\mathbf{X}) = \det(\mathbf{V})^2 \det(\mathbf{A}^T) \det(\mathbf{T}_S^I) \quad (2.56)$$

By the definition of an orthogonal rotation matrix, it is known that

$$\det(\mathbf{T}_S^I) = 1 \quad (2.57)$$

Because $\mathbf{V}$ is orthogonal, it is known that

$$\det(\mathbf{V}) = \pm 1 \quad (2.58)$$

Using Eq. 2.57 and Eq. 2.58, it can be seen that

$$\det(\mathbf{X}) = \det(\mathbf{A}^T) \quad (2.59)$$

substitute in $\mathbf{A} = \mathbf{U}\mathbf{V}^T$ from before,

$$\det(\mathbf{X}) = \det(\mathbf{U}) \det(\mathbf{V}^T) = \det(\mathbf{U}) \det(\mathbf{V}) \quad (2.60)$$

Thus, it can be shown that the optimal solution of $\mathbf{X}$ is

$$\hat{\mathbf{X}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}) \det(\mathbf{U}) \end{bmatrix} \quad (2.61)$$

Using Eq. 2.52, where the optimal solution $\hat{\mathbf{X}}$ is substituted in for $\mathbf{X}$, the equation can be solved for $\mathbf{T}_S^I$.

$$\mathbf{T}_S^I = \mathbf{A} \mathbf{V} \hat{\mathbf{X}} \mathbf{V}^T \quad (2.62)$$

Finally, once again substituting $\mathbf{A} = \mathbf{U} \mathbf{V}^T$ and canceling terms, the optimal rotation matrix can be written as

$$\hat{\mathbf{T}}_S^I = \mathbf{U} \hat{\mathbf{X}} \mathbf{V}^T \quad (2.63)$$

or ultimately

$$\hat{\mathbf{T}}_S^I = \mathbf{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}) \det(\mathbf{U}) \end{bmatrix} \mathbf{V}^T \quad (2.64)$$

Thus from the above derivation, it can be shown how to obtain the optimal rotation matrix that maps vector measurements in one frame to vector measurements in another frame, using the Singular Value Decomposition method as presented by Markley [46].

Chapter 3

Construction of Star Pattern Index

3.1 Selection of Bright Star Catalog

The approach used here focuses on relatively bright stars, as these are generally easier to find, have a higher signal-to-noise ratio, and are of sufficient quantity for high quality image alignment. Using only the brightest stars also reduces the size of the database that must be searched. If superior attitude precision is required, a more complete star catalog could be used after the initial alignment is complete and stars can easily be identified by their apparent location in the image.

3.2 Unit Vectors from star catalog information

The Yale Bright Star catalog contains information regarding over 9,000 of the brightest stars in the sky [3]. The catalog consists of various information including Right Ascension, Declination and apparent magnitude. From this information, it is possible to calculate the line-of-sight direction (unit vector) for each star in the inertial frame,

Figure 3.1: Figure to help visualize how to calculate unit vector measurements from Right Ascension and Declination information. Figure taken from [2] with permission from Authors.

$$\mathbf{u}_{i_I} = \begin{bmatrix} \cos \delta_i \cos \alpha_i \\ \cos \delta_i \sin \alpha_i \\ \sin \delta_i \end{bmatrix} \quad (3.1)$$

where $\mathbf{u}_{i_I}$ is the unit vector in the inertial frame, α_i is the Right Ascension, and δ_i is the Declination of the i -th star. To compact notation in later discussions, define the set of all possible line-of-sight directions, $\mathbb{D}$, as

$$\mathbb{D} = \{ \mathbf{u} \in \mathbb{R}^3 : \|\mathbf{u}\| = 1 \} \quad (3.2)$$

which is simply the set describing the surface of the unit sphere.

The index must have enough sufficiently distinct patterns to produce the desired results, while not containing so many patterns that the search time becomes too long. In the event this approach is used in an actual space mission, the entire star ID

process must be completed in real time. Understanding this, it was decided to use a subset of the brightest stars taken from the catalog.

Although the choice is somewhat arbitrary, we decided to build a pattern database using only stars of magnitude 5.5 or brighter. As a point of reference, the constellation Cancer (the dimmest constellation in the Zodiac) is composed of five main stars — and the dimmest of these has an apparent magnitude of approximately 4.67. By setting the threshold to only consider stars with an apparent magnitude of 5.5 or brighter, the catalog used to construct our pattern database ultimately consists of the 2,887 brightest stars from the Yale Bright Star Catalog. Figure 3.2 gives a view of the distribution of the star directions on the celestial sphere and Figure 3.3 shows the distribution of star magnitudes in the Yale Bright Star Catalog.

Figure 3.2: Visualization of brightest stars on the celestial sphere.

3.3 Star Angles

Typically, the angles between stars or some variation of these angles are used in the construction of the predetermined index. As previously mentioned, this is because

Figure 3.3: Distribution of star magnitudes from the Yale Bright Star Catalog.

the angle between any two stars is frame invariant, meaning that the angle will be the same whether measured in the inertial or sensor frame.

Figure 3.4: Angle between the unit vector of star i and the unit vector of star j .

Fig. 3.4 shows a visual representation of the angle between two arbitrary stars. From Eq. 3.1, the unit vectors $\mathbf{u}_i$ and $\mathbf{u}_j$ can be calculated. It is well known that the

inner product of any two unit vectors is simply the cosine of the angle between them. This can be written in equation form as

$$\mathbf{u}_i \mathbf{u}_j^T = \cos \theta \quad (3.3)$$

With the angle between stars now known, the angles associated with star pairs can be used in a Geometric Hashing approach to construct star patterns.

3.4 Geometric Hashing with Star Quads

Geometric hashing is a common pattern recognition technique used within the computer vision community [41]. Although there are many variants, the fundamental idea is to use the arrangement of points within the pattern to generate a basis for describing the pattern that is invariant to particular transformations. A further explanation of Geometric Hashing can be found in Section 2.4. The concept of geometric hashing has previously been applied to star pattern recognition for the registration of astronomical images [34]. This thesis extends this idea for use with a calibrated star tracker by generating a geometric hash code directly from measurements on $\mathbb{D}$ instead of from a normalization of the starfield projected onto the image.

The hash codes considered here consist of angular information about a grouping of four distinct stars which we refer to as a “quad.” For every quad that is considered, there is a corresponding hash code stored in the index.

To begin, select a set of four unique stars. Remembering from Section 3.3, the angle between any two star line-of-sight directions is simply

$$\theta_{ij} = \theta_{ji} = \cos^{-1}(\mathbf{u}_i^T \mathbf{u}_j) \quad (3.4)$$

where $\mathbf{u}_i \in \mathbb{D}$ and $\mathbf{u}_j \in \mathbb{D}$ are the unit vectors of the two observed stars. The interior

Figure 3.5: A set of two simple criteria may be used to uniquely assign labels A through D to the four stars in a quad. The first criteria states that stars A and B share the largest interior star angle. The second criteria states the stars A and C share the smallest interior star angle.

star angle θ_{ij} is computed for all six possible combinations of stars from the quad. Note that we do not distinguish between the ordering of the subscripts and $\theta_{ij} = \theta_{ji}$ simply refers to the angle between the i -th and j -th star. Define the set of all six interior star angles as $\mathbb{A}$,

$$\mathbb{A} = \{\theta_{12}, \theta_{13}, \theta_{14}, \theta_{23}, \theta_{24}, \theta_{34},\} \quad (3.5)$$

Importantly, these interior star angles are always the same for a particular pattern, regardless of the frame in which the line-of-sight directions are expressed.

From the set of six calculated interior star angles, find the four angles that form the perimeter of the quad. These four angles may be used to encode the shape and size of the pattern and hence form a four-dimensional (4D) hash code. The remaining difficulty in assigning such a hash code is deciding the order in which these four interior angles should be listed. Thus, we apply the concept of geometric hashing to form a pattern descriptor that is invariant to both sensor attitude and the order in which the candidate star centroids were detected in the image.

As an example, consider the quad selected from amongst many stars in Fig. 3.5. Suppose that a set of four stars are extracted from the full set of candidate stars observed in a particular image. These four stars are labeled 1 through 4 as they

are selected. This labeling is, of course, completely arbitrary. Thus, we now aim to relabel the stars with the letters A through D in a way that will be unique regardless of the order in which the initial star selection was performed.

The unique labeling of A through D may be performed using two criteria. The first criterion is that star A and star B form the *largest* of the six possible interior star angles. The second criterion is that star A and star C form the *smallest* of the four remaining interior star angles that contain one of the two stars identified by the first criterion. The common star amongst the two criteria is labeled star A, and the remaining labels follow directly.

Therefore, suppose we use the indices i , j , k , and l to describe the star labels 1 through 4. Defining the set of four interior star angles used for the second criteria as $\mathbb{B}$,

$$\mathbb{B} = \{\theta_{ik}, \theta_{il}, \theta_{jk}, \theta_{jl} \mid \theta_{ij} = \max(\mathbb{A})\} \quad (3.6)$$

the unique star labeling may be described mathematically as

$$\theta_{ij} = \max(\mathbb{A}) \wedge \theta_{ik} = \min(\mathbb{B}) \quad (3.7)$$

→ label observation i as star A

→ label observation j as star B

→ label observation k as star C

→ label remaining observation as star D

Referring back to the example in Fig. 3.5, we may better visualize how these criteria are met. In this example, the largest interior star angle is between star 1 and 4. From the first criterion, we know that star A will be either 1 or 4. The smallest interior angle containing either star 1 or 4 in this example is between star 2 and 4, meaning that the second criteria tells us that star A will be either 2 or 4. The combination of the two criteria tells us that star 4 should be labeled as star A.

Further, the first criteria tells us that star B is either 1 or 4. Since star 4 is known to be A, it is easy to see that star 1 should be labeled as star B. Likewise, using the second criteria we find that star 2 should be labeled star C. Finally, the remaining star must clearly be labeled as star D.

After the set of stars in a quad have been given a unique sequence of labels A through D it becomes straightforward to create a hash code. As discussed earlier, the hash code will be built from the four exterior angles describing the quad, $\{\angle AC, \angle AD, \angle BC, \angle BD\}$, though this order is arbitrary as long as all codes are constructed the same way. Finally, note that each hash code is a unique description of a pattern since the labeling A through D is invariant to sensor attitude or the initial order in which the four stars were selected.

3.5 Generation of Quad Index from Star Catalog

With a geometric hashing technique for describing star patterns in hand, the next step is to build an index of quads from the star catalog. This index must have good coverage of the entire celestial sphere and enough unique quads to be able to recognize patterns in any given image. At the same time, however, the total number of quads must be limited to stay within reasonable memory limitations and to be searchable in real time.

Bright stars are not evenly distributed throughout the sky (see Fig. 3.2) and the star tracker will have a finite field of view (FOV). With these observation in mind, it is desired to create the index that uniformly samples the celestial sphere at a resolution derived from the sensor FOV. Further, the quads built at each sampling location should be constrained in size based on the sensor FOV.

3.5.1 Uniform Sampling of the Celestial Sphere

A simple uniform sweep of azimuth-elevation combinations will not produce a uniform sampling of the celestial sphere. This approach would sample more densely at higher elevation angles and less densely at lower elevation angles. Thus, two different techniques were considered for generating the reference locations on the celestial sphere for building the star quad index. These two methods are now discussed.

One way to evenly sample the surface of a sphere is through an inscribed regular polyhedron, such as that shown in Fig. 3.6. The vertices of such polyhedrons lie on the unit sphere and are uniformly spaced.

Figure 3.6: Dodecahedron inscribed within a unit sphere that represents a possible polyhedron that can be used to create uniformly spaced vertices.

The choice of polyhedron is driven by the desired number of vertices and is application dependent. The number of required vertices tends to be smaller in cases where the star tracker has a larger FOV. Ideally, the FOV about each vertex being used to construct an index of star patterns will be the same size, if not larger, than the FOV of the camera being used. This will help to insure that no matter what observed bright stars are selected in the sensor frame, a pattern of these stars is highly likely to exist in the index. The inscribed polyhedron method, however, tends to be practically limited to a small number of vertices. Additionally, the number of vertices

cannot be arbitrary chosen for regular polyhedra.

For applications where the camera being used to identify star patterns has a rather small FOV (as is the case with most star trackers), the number of vertices needed to construct the index must be rather large in order to adequately sample the entire sky. Inscribed polyhedra are not practical for such a large number of vertices.

Therefore, a second technique is proposed. Creating a spiral of points (Fig. 3.7) has been shown to be a simple and effective way to generate a set of uniformly spaced points on a unit sphere [51]. Generation of the point spiral is straightforward.

Consider the following parameterization of a unit vector,

$$\mathbf{u}_n = \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = \begin{bmatrix} \sin \phi_n \cos \theta_n \\ \sin \phi_n \sin \theta_n \\ \cos \phi_n \end{bmatrix} \quad (3.8)$$

where θ_n is the azimuth of the n -th point and $90^\circ - \phi_n$ is the elevation angle of the n -th point. As was mentioned earlier, a uniform sampling of these two angles would sample the surface of the sphere more heavily near the poles. However, a *nonuniform* sampling of θ_n and ϕ_n in the shape of a spiral will produce an approximately *uniform* sampling of the surface of the sphere.

To generate such a spiral, first define $k \in \mathbb{Z}$ as the integer number of points to be sampled on the surface of the sphere. At the n -th sample location define a_n as

$$a_n = \cos(\phi_n) = 1 - \frac{2n-1}{k} \quad (3.9)$$

Thus, ϕ_n can readily be computed as

$$\phi_n = \cos^{-1}(a_n) \quad (3.10)$$

Likewise, defining the constant b as

$$b = \sqrt{k\pi} \quad (3.11)$$

the value of θ_n may also be computed

$$\theta_n = b\phi_n \quad (3.12)$$

The spiral method for producing k number of vertices works well when using a camera with a small FOV since it is easy to sample at an arbitrary resolution.

Figure 3.7: A spiral on the surface of unit sphere can be used to create k points with near uniform spacing.

3.5.2 Creation of Measurably Unique Quads

With the coordinates of the k number of vertices determined, the index of hash codes can be created. At each vertex, consider the m brightest stars within a specified FOV. Hash codes can be created for all quads that are formed from the m brightest stars. That is, all four star combinations of the m brightest stars can be used to form a hash code.

It is required, however, that all created quads satisfy the following two conditions: (1) a star cannot be used to generate a catalog quad if it is within a specified tolerance of another bright star and (2) no two stars in a quad can be closer than $\theta_{\min}$ (that is, $\min(\mathbb{A}) \geq \theta_{\min}$). The first criteria forces stars in the index to be measurably unique and the second criteria forces the minimum quad edge length to be larger than a minimum size. Together, these two conditions help to ensure unique quads are used to generate hash codes and help minimize cases where two created hash codes are similar enough that they may be misidentified.

At each vertex location, a set of quads is formed from all four star combinations of the m brightest stars within a specified FOV that satisfy the two conditions mention above. An index of hash codes is created by accumulating this set of quads from every vertex location.

3.6 Storing Index in k -d Tree for Fast Searching

The procedure described above will generate a very large number of hash codes, each of which represents a particular star quad. This information must now be stored in a manner that permits it to be efficiently searched. In this thesis, it was chosen to store the index of hash codes in a k -d tree structure [42]. A more in-depth discussion of k -d trees and k -d tree searching can be found in Section 2.5.

Although building the k -d tree has a complexity of $O(n \log n)$, the task of building the index is only done once and occurs before use (so it may be computed preflight). Searching the k -d tree, however, is very fast and only has a complexity of $O(\log n)$.

Chapter 4

Star Pattern Recognition

We now seek to identify star patterns in an image of a star field. This is accomplished by forming hash codes from observed star quads in the image and then attempting to match this to the hash code index stored in a k -d tree. The overall process has a number of steps and is summarized in Fig. 4.1. This process will be explained in more detail in the following sections.

4.1 Star Measurements in the Sensor Frame

Once an image is acquired, the first step is to find stars within this image. This is done through a star centroiding process, for which there are a variety of very well-known techniques [15–17]. The centroid pixel coordinates may be related to line-of-sight directions in the sensor frame through a straightforward calibration process [11].

Figure 4.1: Flow chart of the entire star ID process.

4.2 Generating Observed Hash Codes and Searching the Index

The process of generating hash codes in an observed image is nearly identical to that of constructing hash code index. First, all possible quads are created for the m brightest observed stars in the image. The observed stars in each quad are labeled A through D, following the process discussed in Section 3.4. Each measured quad is then screened to ensure it meets the two criteria from Section 3.5.2. The result is a set of candidate observed quads.

Measurement noise will generally prevent the hash code of an observed quad from exactly matching a quad in the index. Thus, we simply seek to find the hash code in the index that is closest to the hash code for an observed quad. The best match is found quickly via a nearest neighbor search [43] on the k -d tree containing the index of hash codes.

Once an observed quad is matched to the index, the correspondence between the observed stars in the quad and the stars in the star catalog is known (this correspondence is known because you know which catalog stars were used to create the best-match hash code in the index). Thus, we now have a set of unit vectors in the sensor frame (derived from the observed star coordinates in the image) and a corresponding set of unit vectors in the inertial frame (obtained from the star catalog). The task is now to find the optimal attitude transformation that rotates the catalog unit vectors in the inertial frame into best agreement with the observed unit vectors in the sensor frame.

4.3 Solving Wahba's Problem to Calculate Attitude

Once a hash code in the sensor frame is matched to an indexed hash code in the inertial frame, the optimal attitude may be calculated by solving Wahba's problem [44]. The derivation and explanation for solving Wahba's problem using the SVD method can be found in Section 2.6. Wahba's problem seeks the rotation matrix, $\mathbf{T}$, that best rotates the catalog star directions onto the observed star directions by minimizing the following cost function:

$$\min J(\mathbf{T}) = \frac{1}{2} \sum_{i=1}^n c_i \|\mathbf{u}_{i_S} - \mathbf{T}\mathbf{u}_{i_I}\|^2 \quad (4.1)$$

where $\mathbf{T}$ is the rotation matrix from the inertial frame to the sensor frame, $\mathbf{u}_{i_S}$ is the i -th unit vector measurement in the sensor frame, and $\mathbf{u}_{i_I}$ is the i -th unit vector measurement in the inertial frame. Though there are many solutions to solving Wahba's problem [45–50], it was decided to use the singular value decomposition (SVD) method originally presented by Markley [46]. See section 2.6 for details.

The SVD method begins with the attitude profile matrix $\mathbf{B}$,

$$\mathbf{B} = \sum_{i=1}^n c_i \mathbf{u}_{i_S} \mathbf{u}_{i_I}^T \quad (4.2)$$

where $\mathbf{u}_{i_S}$ and $\mathbf{u}_{i_I}$ are the same as in Eq. 4.1. Since all the measurements are collected by the same sensor, we can assume the same weighting for them all, $c_i = 1$

Now that $\mathbf{B}$ has been defined in Eq. 4.2, the SVD can be performed,

$$\mathbf{B} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (4.3)$$

Once $\mathbf{U}$ and $\mathbf{V}$ are determined, the optimal estimate of the rotation matrix $\mathbf{T}$ can

be solved using the following equation

$$\hat{\mathbf{T}} = \mathbf{U}\hat{\mathbf{X}}\mathbf{V}^T \quad (4.4)$$

The matrix $\hat{\mathbf{X}}$ is simply

$$\hat{\mathbf{X}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V})\det(\mathbf{U}) \end{bmatrix} \quad (4.5)$$

which is chosen to ensure that $\hat{\mathbf{T}}$ is right handed (which simply requires that $\det(\mathbf{T}) = +1$).

4.4 Pattern Verification Process

Suppose that an observed quad was successfully matched to an indexed quad. Further suppose that the resulting star correspondences were used to solve for an attitude estimate via the SVD solution to Wahba's problem. A final verification step is required to ensure a robust star ID has occurred. This verification process has two components: (1) confirm one-to-one ID by reprojection of the catalog and (2) matching of at least two additional non-quad stars for final attitude confirmation. These components are now discussed in detail.

The first verification check confirms a successful one-to-one correspondence of the transformed catalog with the observed stars. The attitude solution generated from the quad can be used to find the expected location of catalog stars in the image. For the first check to be passed, there must be one and only one projected catalog star within a specified tolerance of each of the four quad stars. The various possible scenarios for any particular star are shown graphically in Fig. 4.2.

The second check in the verification process considers stars that are not part

Figure 4.2: A successful star ID requires that one and only one catalog star be within a specified tolerance of the observed star, as shown in the left most frame. If no stars are within the specified tolerance (center frame) or more than one star is within the specified tolerance (right frame) then no star ID is assigned. In all three frames, the solid black dot represents the observed star and the gray dots denote the catalog stars projected onto the image.

of the quad. As with the first check, the rotated catalog stars are projected onto the observed image. Now, the $m - 4$ stars not used to create the current quad are considered. If any of these stars has one and only one catalog star within a specified tolerance, the corresponding catalog ID is attributed to that star. Otherwise no ID is assigned. The second verification check is passed if two or more non-quad stars are successfully matched with a catalog ID. Importantly, not all observed stars in an image need be identified — which makes the approach robust to non-star objects appearing in the image. All that is required is that the four quad stars, along with two other verification stars, be actual stars that exist in the star catalog.

If both verification checks are not passed for the current quad, another observed quad from the m observed stars is selected, the attitude is recomputed, and the verification process is repeated. If all quads are checked and no combination of stars passes both verification checks then no star IDs are assigned.

Once a quad makes it through the verification process, all of the correctly matched selected stars can then be used to obtain a better estimate of attitude via the SVD solution to Wahba’s problem. This is an important step since the initial attitude esti-

mate was based on only the four quad stars. A significantly better attitude estimate may be achieved by using all of the identified stars in the image.

Chapter 5

Performance Assessment via Monte Carlo Analysis

5.1 Setup/Index Construction

A Monte Carlo simulation was conducted to determine how well the star ID technique presented here works for a star tracker with a 20° FOV. A detailed description of the simulation is now presented.

The first step is to construct the index of catalog hash codes. As mentioned previously, the Yale Bright Star Catalog was selected and only stars with an apparent magnitude of 5.5 or brighter were retained. The index was created using the spiral method for sampling the celestial sphere. Using 300 near uniformly spaced pointing directions and a 20° FOV about each pointing direction, quads were generated using all 4 star combinations of the 20 brightest stars in each FOV. Fig. 5.1 shows the spiral of points that were used to generate the index.

If less than 20 stars are in the FOV, then the total number of stars in the FOV were used. No bright star within 80 arcsec of another bright star was used to construct a quad and the quads were checked to ensure $\min(\mathbb{A}) \geq \theta_{\min} = 100$ arcsec. Since

Figure 5.1: Figure showing the spiral of points used to construct the index. The red circles show the location of the pointing directions and the black stars show the locations of bright stars.

the simulated star tracker is assumed to be capable of measuring star line-of-sight directions with $\sigma = 10$ arcsec, this ensures that each quad has a minimum edge length at least an order of magnitude greater than the sensor noise. If any hash codes were generated more than once, the duplicate was removed from the index. This process resulted in an index of 938,635 unique hash codes.

5.2 Simulation

With the index constructed, the simulation to match hash codes was conducted. The simulation was set up with three possible outcomes. These outcomes were “Correct Match,” “No Match”, and “False Match”.

In order to simulate an image taken by a star tracker, a simulated camera was placed in a random pointing direction and measurements to stars within the FOV

were recorded. The angular error of each star line-of-sight direction was assumed to be $\sigma = 10$ arcsec.

For each simulated image in the Monte Carlo analysis, the 10 brightest stars in the FOV were used as observed star measurements. All possible 8 star combinations of these 10 brightest stars are then calculated. All 8 stars are checked to ensure they are further than 100 arcsec apart from each other. If stars used in one of the 8 star combinations are less than 100 arcsec apart, that particular combination is not used.

Once the 8 star combinations are calculated for a particular simulated image, we select the first 8 star combination and generate all possible quads. We then attempt to match each quad and, if successful, attempt to verify the match using the two component verification process. If no match is found, the next set of 8 stars is selected and the process is repeated. If all combinations are considered and no match is found, that particular image is labeled as a “No Match.” Images whose pattern is matched are labeled either “Correct Match” if all stars are correctly identified or “False Match” if any one star is misidentified. If a particular image has less than 8 usable stars, it is labeled as a “No Match”.

5.3 Simulation Results

The Monte Carlo analysis was ran for 100,000 different cases, each time generating a new random pointing direction and selecting the 10 bright stars in the sensor frame to simulate a real lost-in-space application. Results are summarized in Table 5.1.

Table 5.1: Matching success results of Monte Carlo simulation

	Correct Match	False Match	No Match
Number of Cases	99,786	1	213

It can be seen from Table 5.1 that star patterns were successfully recognized and star IDs assigned in 99.79% of the cases. The algorithm determined that a reliable

Figure 5.2: Four randomly generated pointing directions. The red circles represent the boresight direction of the camera, the green stars represent stars within the camera FOV, and the blue stars represent the 10 brightest stars in the camera FOV.

match could not be made in 0.21% of the cases. Though we generated 1 “False Match” (representing only 0.001% of the test cases), it is important to note that the Monte Carlo was set up to record a “False Match” when *any* single star is falsely matched — regardless of whether or not the remaining stars were correctly matched. In this particular “False Match” case, 7 of the 8 observed stars were still correctly identified and an accurate estimate of attitude was still calculated. The single misidentified star ID occurred when trying to match a star with a neighbor star just beyond 80 arcsec threshold, and the measurement noise made the observed star appear closer to the nearby star than the actual star. As a result, this misidentification had no discernible impact on the resulting attitude estimate.

The calculated attitude errors for all 100,000 cases (including the single “False Match” case) are summarized in Figs. 5.3–5.6 and the statistics are enumerated in Table. 5.2.

Table 5.2: Attitude error statistics for Monte Carlo simulation

	X Direction	Y Direction	Z Direction
Max Error (arcsec)	154.74	322.75	206.97
Mean Error (arcsec)	0.0161	−0.0602	−0.0173
Standard Deviation (arcsec)	20.66	20.81	20.78
Skewness	0.0024	0.0487	0.0102
Kurtosis	6.18	6.96	6.15

Additionally, Fig. 5.7 shows the number of quads that were checked before a correct match was determined. Importantly, the correct match was found using the first quad hash code generated in 59.3% of the cases. Thus, despite the sophisticated scheme for exploring different quad hash code combinations, a correct match is made with just the first quad in well over half the cases.

Figure 5.3: Attitude errors in x direction for all 100,000 Monte Carlo cases.

Figure 5.4: Attitude errors in y direction for all 100,000 Monte Carlo cases.

Figure 5.5: Attitude errors in z direction for all 100,000 Monte Carlo cases.

Figure 5.6: Magnitude of the attitude errors for all 100,000 Monte Carlo cases.

Figure 5.7: Cumulative density function of number of quads checked to reach a decision (either confirm a match or to determine that no match is possible).

Chapter 6

Conclusion

As the interest in deep space missions continues to grow, new and improved techniques must be made in the area of spacecraft navigation. This thesis presents a novel star pattern identification algorithm that recognizes unique star patterns and calculates spacecraft attitude using the identified star measurements. It adopts a geometric hashing approach that stores hash codes constructed from the interior angles of star quads in a predetermined index. Once the index is created, it is stored for fast searching with accurate results using a k -d tree.

Perhaps one of the main goals of this thesis is to not only be able to identify stars, but also be able to identify cases where a false positive star ID could occur. It can be said from the results presented here that this goal was achieved. The thesis proposes an intelligently designed decision process that can determine cases of pattern ambiguity in the identification process that could result in a false positive in star identifications. Once the stars are correctly identified, the spacecraft attitude can be accurately determined. Practical issues related to pattern index construction and pattern verification are also discussed.

Performance Assessment via Monte Carlo analysis demonstrates that the system can correctly identify star patterns and compute an attitude estimation in 99.79% of

the cases and determines that no match can reliably be made in 0.21% of the cases. False matches occur very infrequently (0.001% in the present simulation) and these mismatches are typically benign from an attitude estimation standpoint.

The proposed star pattern recognition algorithm could be integrated in an actual star tracker and used for real world applications. Another benefit of this thesis research is that this pattern recognition process is not limited to star trackers. The proposed decision process could also be used in other pattern recognition applications.

Bibliography

- [1] Fisher, D. and Beck, S., “Technology Star Camera,” Jet Propulsion Laboratory, http://nmp.nasa.gov/st6/TECHNOLOGY/star_camera.html, [Cited March 9, 2016].
- [2] Gioia, C., Gerhard, J., and Christian, J., “Manual Optical Attitude Re-Initialization for a Crew Vehicle in Space,” *ION International Technical Meeting, Dana Point, CA*, 2015.
- [3] Hoffleit, D. and Warren, W., *The Bright Star Catalog, 5th Revised Edition (Preliminary Version)*, 1991.
- [4] Liounis, A., Daniel, S., and Christian, J., “Autonomous Navigation System Performance in the Earth-Moon System,” *AIAA Space 2013 Conference and Exposition, San Diego, CA*, 2013.
- [5] Sell, J., Rhodes, A., Woods, J., Christian, J., and Evans, T., “Pose Performance of LIDAR-Based Navigation for Satellite Servicing,” *AIAA Astrodynamics Specialist Conference, San Jose, CA*, 2014.
- [6] Bittner, D., Christian, J., and Bishop, R., “Development of an Alignment Technique for a Large Number of Redundant Inertial Measurement Units,” *65th International Astronautical Congress, Toronto, ON*, 2014.

- [7] Straub, M. and Christian, J., “Autonomous Optical Navigation for Earth-Observing Satellites Using Coastline Matching,” *AIAA Guidance, Navigation, and Control Conference, Orlando, FL*, 2015.
- [8] Christian, J., “Optical Navigation Using Planet’s Centroid and Apparent Diameter in Image,” *Journal of Guidance, Control, and Dynamics*, Vol. 38, No. 2, 2015, pp. 192–204.
- [9] Liounis, A., Gerhard, J., and J.A., C., “Performance Assessment of Horizon-Based Optical Navigation Techniques,” *38th Annual AAS Guidance, Navigation, and Control Conference, Breckenridge, CO*, 2015.
- [10] Woods, J., Christian, J., and Evans, T., “A 6-DOF Pose Initialization Strategy for LIDAR-Based Non-Cooperative Navigation,” *38th Annual AAS Guidance, Navigation, and Control Conference, Breckenridge, CO*, 2015.
- [11] Christian, J., Benhacine, L., and Hikes, J., “Geometric Calibration of the Orion Optical Navigation Camera using Star Field Images,” *39th Annual AAS Guidance, Navigation, and Control Conference, AAS 16-116, Breckenridge, CO*, 2016.
- [12] Christian, J. and Robinson, S., “Observations on the Geometry of Horizon-Based Optical Navigation,” *39th Annual AAS Guidance, Navigation, and Control Conference, AAS 16-116, Breckenridge, CO*, 2016.
- [13] Liebe, C. C., “Accuracy Performance of Star Trackers — A Tutorial,” *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 38, No. 2, 2002, pp. 587–599.
- [14] Markley, F. L. and Crassidis, J. L., *Fundamentals of spacecraft attitude determination and control*, Vol. 33, Springer, New York, NY, 2014.

- [15] Thomas, S., Fusco, T., Tokovinin, A., Nicolle, M., Michau, V., and Rousset, G., “Comparison of centroid computation algorithms in a Shack–Hartmann sensor,” *Monthly Notices of the Royal Astronomical Society*, Vol. 371, No. 1, 2006, pp. 323–336.
- [16] Coralie, J. and Philip, D., “Optical Navigation Capabilities for Deep Space Missions,” *23rd AAS/AIAA Space Flight Mechanics Conference, Kauai, HI*, 2013.
- [17] Dennison, E. and Stanton, R., “Ultra-precise star tracking using charge coupled devices (CCDs),” *24th Annual Technical Symposium*, International Society for Optics and Photonics, 1980, pp. 54–63.
- [18] Junkins, J. L., White, C. C., and Turner, J. D., “Star Pattern Recognition for Real Time Attitude Determination,” *Journal of the Astronautical Sciences*, Vol. 25, No. 3, 1977, pp. 251–270.
- [19] Groth, E. J., “A Pattern-Matching Algorithm for Two-Dimensional Coordinate Lists,” *The Astronomical Journal*, Vol. 91, 1986, pp. 1244–1248.
- [20] Sheela, B., Shekhar, C., Padmanabhan, P., and Chandrasekhar, M., “New Star Identification Technique for Attitude Control,” *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, 1991, pp. 477–480.
- [21] Kosik, J. C., “Star Pattern Identification Aboard an Inertially Stabilized Aircraft,” *Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, 1991, pp. 230–235.
- [22] Liebe, C. C., “Pattern Recognition of Star Constellations for Spacecraft Applications,” *Aerospace and Electronic Systems Magazine, IEEE*, Vol. 7, No. 6, 1992, pp. 34–41.

- [23] Mortari, D., “A Fast On-Board Autonomous Attitude Determination System Based on a New Star-ID Technique for a Wide FOV Star Tracker,” *Advances in the Astronautical Sciences*, Vol. 93, 1996, pp. 893–904.
- [24] Mortari, D., “Search-Less Algorithm for Star Pattern Recognition,” *The Journal of the Astronautical Sciences*, Vol. 45, No. 2, 1997, pp. 179–194.
- [25] Mortari, D., Junkins, J. L., and Samaan, M., “Lost-In-Space Pyramid Algorithm for Robust Star Pattern Recognition,” *Guidance and control 2001*, 2001, pp. 49–68.
- [26] Mortari, D., Samaan, M. A., Bruccoleri, C., and Junkins, J. L., “The Pyramid Star Identification Technique,” *Navigation*, Vol. 51, No. 3, 2004, pp. 171–183.
- [27] Padgett, C. and Kreutz-Delgado, K., “A Grid Algorithm for Autonomous Star Identification,” *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 33, No. 1, 1997, pp. 202–213.
- [28] Cole, C. L. and Crassidis, J., “Fast Star Pattern Recognition Using Spherical Triangles,” *AIAA/AAS Astrodynamics Specialist Conference and Exhibit. Providence, Rhode Island: AIAA*, 2004.
- [29] Cole, C. L. and Crassidis, J. L., “Fast Star-Pattern Recognition Using Planar Triangles,” *Journal of guidance, control, and dynamics*, Vol. 29, No. 1, 2006, pp. 64–71.
- [30] Kolomenkin, M., Pollak, S., Shimshoni, I., and Lindenbaum, M., “Geometric Voting Algorithm for Star Trackers,” *Aerospace and Electronic Systems, IEEE Transactions on*, Vol. 44, No. 2, 2008, pp. 441–456.

- [39] Mason, B. D., Wycoff, G. L., Hartkopf, W. I., Douglass, G. G., and Worley, C. E., “The 2001 US naval observatory double star CD-ROM. I. The Washington double star catalog,” *The Astronomical Journal*, Vol. 122, No. 6, 2001, pp. 3466.
- [40] Seidelmann, P. K., *Explanatory supplement to the astronomical almanac*, University Science Books, 2005.
- [41] Wolfson, H. and Rigoutsos, I., “Geometric Hashing: An Overview,” *IEEE Computational Science and Engineering*, Vol. 4, No. 4, 1997, pp. 10–21.
- [42] Bentley, J. L., “Multidimensional Binary Search Trees Used for Associative Searching,” *Communications of the ACM*, Vol. 18, No. 9, 1975, pp. 509–517.
- [43] Friedman, J. H., Bentley, J. L., and Finkel, R. A., “An algorithm for finding best matches in logarithmic expected time,” *ACM Transactions on Mathematical Software (TOMS)*, Vol. 3, No. 3, 1977, pp. 209–226.
- [44] Wahba, G., “A Least Squares Estimate of Satellite Attitude,” *SIAM review*, Vol. 7, No. 3, 1965, pp. 409–409.
- [45] Lerner, G. M., “Three-Axis Attitude Determination,” *Spacecraft Attitude Determination and Control*, Vol. 73, 1978, pp. 420–428.
- [46] Markley, F. L., “Attitude Determination Using Vector Observations and the Singular Value Decomposition,” *The Journal of the Astronautical Sciences*, Vol. 36, No. 3, 1988, pp. 245–258.
- [47] Markley, F. L. and Mortari, D., “How to Estimate Attitude from Vector Observations,” *Proceedings of the AAS/AIAA Astrodynamics Specialist Conference*, Vol. 103, 1999, pp. 1979–1996.
- [48] Mortari, D., “ESOQ: A Closed-Form Solution to the Wahba Problem,” *Journal of the Astronautical Sciences*, Vol. 45, No. 2, 1997, pp. 195–204.

- [49] Shuster, M. D. and Oh, S., “Three-Axis Attitude Determination From Vector Observations,” *Journal of Guidance, Control, and Dynamics*, Vol. 4, No. 1, 1981, pp. 70–77.
- [50] Wertz, J. R., *Spacecraft Attitude Determination and Control*, Vol. 73, Springer Science & Business Media, 2012.
- [51] Bauer, R., “Distribution of Points on a Sphere with Application to Star Catalogs,” *Journal of Guidance, Control, and Dynamics*, Vol. 23, No. 1, 2000, pp. 130–137.

Appendix