2010:014

MASTER'S THESIS

Attitude Determination Software for a Star Sensor

Raveesh Kandiyil

Luleå University of Technology

Master Thesis, Continuation Courses
Space Science and Technology
Department of Space Science, Kiruna

2010:014 - ISSN: 1653-0187 - ISRN: LTU-PB-EX--10/014--SE

University of Würzburg

Faculty of Mathematics and Computer Science

Department of Robotics and Telematics

Master Thesis

Author: Raveesh Kandiyil

Supervisors:

Prof.Dr.Hakan Kayal

Dr.Anita Enmark

Attitude determination software for a star sensor

Raveesh Kandiyil
Department of Robotics and Telematics
JMUW Germany
raveesh_03@yahoo.co.in

September 2009

Declaration

I,Raveesh Kandiyil, hereby declare that this submission is my own work and that, to the best of my knowledge and belief all resources used are properly acknowledged in the attached list.

Würzburg Date ____________________ ____________________

Signature

Acknowledgements

First and foremost I offer sincere gratitude to my supervisor, Prof.Dr.Hakan Kayal, for offering me a great opportunity to work on this thesis and providing me with excellent guidance and support throughout the work.

I would like to thank my LTU supervisor Dr.Anita Enmark for her valuable suggestion through e-mails.

I am obliged to Mr. Oleksii Balagurin for spending time with me for discussing the technical issues and for his kind support.

I would like to take this opportunity to thank Prof.Dr.Klaus Schilling, Dr.Sven Molin, Dr. Victoria Barabash and several other people who gave me the wonderful opportunity for this SpaceMaster programme.

I am grateful to Henry Hallam (Cambridge University) and ASIAS Team (York University) for providing star images and freely available MATLAB codes through MATLAB community

I would also like to thank the Lehrstuhl fur Informatik VII , University of Würzburg for giving me the facilities for my thesis work.

Finally, I thank my parents and friends for supporting me throughout all my studies at University.

Abstract

There are a number of universities which are developing pico and nano satellite for research and studies and will continue to increase in the future. Most of them also have specific attitude determination methods and associated sensors, but still a very accurate attitude determination sensor system is not available for small satellites other than sun sensors and magnetometers. Considering the accuracy and performance star sensor is the best choice, but the major drawbacks are the cost, effort and weight of the star sensor. The objective of the thesis work is to develop attitude determination software for a small low cost star sensor which can be used in pico and nano satellites. Modern miniaturization of electronics and hardware and low cost due to mass production has paved new way to develop a miniature sensor for small satellites in university projects. The work is mainly concentrated on developing a prototype software for the current star sensor project undergoing at the University of Würzburg, known as STELLAR. The prototype software is build in MATLAB 7 on a normal PC and tested for publicly available various star images and generated images.

Contents

List of Figures

List of Tables

Symbols and Abbreviation

$\times$	Cross product
$\cdot$	Dot product
$\|\|\|\|$	Normalisation of vector
3D	Three dimensional
$\mathbf{e_i}$	Eigen vector
$\mathbf{f}$	Focal length of camera
$\lambda_{\mathbf{i}}$	Eigen values
M	Magnitude
$\mathbf{q}$	quaternions
$\mathbf{R_B^I}$	Rotation matrix from body to inertial coordinate frame
tr	Trace of a matrix
APS	Active Pixel Sensor
AU	Astronomical unit
CCD	Charge Coupled Device
CMOS	Complementary metal oxide
Dec	Declination of the Star
ECEF	Earth Centred Earth Fixed
ECI	Earth Centred Inertial
FoV	Field of View
LEO	Low Earth Orbit
LISA	Lost in Space Algorithm
LUT	Look up table
Mag	Magnitude
PIXEL	Picture element of an image
QUEST	An Attitude estimation algorithm

RA	Right Ascension of the Star
TF	Triangular feature
TRIAD	An Attitude estimation algorithm

Chapter 1

Introduction

1.1 Introduction

The space industry has been booming for the past two decades and new technologies are being tested throughout different space agencies, organization and universities. Due to the advent of the miniaturization of electronic components and devices there are several small satellite projects in the universities across the globe and also corresponding hardware development for these projects. Most of them are used for research purposes and studying new technologies. The reason why universities are engaging in space technology is to provide hands on experience to students and study various technologies. New technologies for communication protocol, solar power generation, imaging and attitude determination are currently under research. Among this attitude determination is one of the important areas of study since it is the system responsible for the orientation of the satellites and spacecraft in the space. Main tasks involved are development of new sensors and onboard software.

There are several sensors such as sun sensors, horizon sensors, star sensors, and rate integration gyros which are currently used in spacecraft systems. When considering the most accurate sensor, the star sensor is the best choice which can provide an accuracy level up to sub-arc range and minimum fault tolerance. They have no moving parts, and they do not experience loss-of-accuracy from systematic drift. While using the normal attitude determination sensor additional components like gyroscopes and high end on board attitude determination software are required which itself takes huge processing power and memory from the onboard computer. The star sensors alone can provide an accurate attitude solution thus reducing onboard efforts. Usually the construction of a star sensor requires effort, time and high price and at a university level this is a challenging project. Before implementing on the hardware system the software has to be designed and tested on a normal PC system. The thesis work deals with the development of the initial version software for a star sensor and the software is tested on PC.

1.2 Attitude determination

Attitude determination is the technique by which the orientation of the spacecraft or satellite in the space is determined. The satellite or the spacecraft has to be correctly pointed in the space since, the solar panels has to be pointed towards the sun, the antenna towards earth and several other sensors to specified directions for its proper functioning. This is one of the important tasks for a complete mission. The attitude is determined by the knowing the direction towards which the satellite is currently pointing (a reference point) and the previously known coordinates of the reference point in inertial coordinates from catalogs or other systems. The knowledge of the coordinates in both body frame and inertial frame gives a unique attitude solution for the satellite. The three axis attitude determination is a complex task in terms of computations and is more complex when satellite has no apriori information of the attitude. There are several sensors which are used to determine attitude like sun sensors, star sensors, magnetometers, gyroscope etc. Out of this star sensor is the one which produces the best attitude estimate even though the complexity of hardware and software is high.

1.3 Different Attitude determination Sensors

1.3.1 Sun Sensors

The sun sensor is the most common attitude determination sensor used in space missions. It senses the presence of sun in the field of view of camera and measures the direction to which the camera is pointed. The relative brightness of the sun when compared to the other stellar objects makes it easy to be detected. Moreover the angular radius of the sun is very small, nearly 0.267 at 1AU. When compared with earth globe the angular radius is so small and it is excellent for low earth space missions. Usually there is a wide range of sun sensor due to difference in operation and designs. There are basically three types of sensors: analog sensors, sun presence sensors, and digital sensors [1]. Digital sensors provide accurate measurement and are more expensive. Most of the times the analog sensors are sufficient to provide attitude information.

1.3.2 Magnetometers

The earth's magnetic field can be used as an excellent means to find out the direction of satellite. Magnetometers are deployed to measure the magnitude and vector direction of the magnetic field of earth and provide an attitude solution. They are light weight and require only very less power for operation. The earth's magnetic field extends to several thousand kilometres but due to unknown precise nature it can't be used at an orbits of high altitude. They are less expensive and easy to operate but is less accurate. So missions which need less accuracy and low cost mostly deploy magnetometers

1.3.3 Star Sensors

Star sensors are the most accurate system for measuring the attitude of a spacecraft which can measure up to sub-arc ranges. Star sensors measure the direction vectors of the star detected in the body frame of the spacecraft and determines attitude. The drawbacks are the instrumentation complexity, elevated price, extensive software requirements and relatively low reliability when compared to other attitude sensor assemblies. The other drawbacks of the star sensors are due to errors from bright sources such as moon, planets and sun. The fixed pattern noise avoidance techniques can reduce these errors to a very good extent. But still they are preferred over most mission due to the accuracy of the measurement.

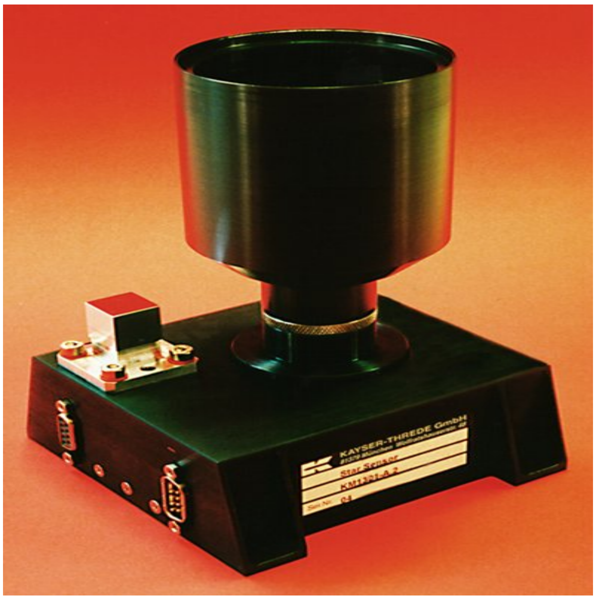


Figure 1.1: Star Sensor image courtesy [2]

1.4 Thesis Overview

1.4.1 Thesis Objective

The proposed thesis work is to conduct initial studies about the field of view and magnitude requirements for the development of star sensor software and also for star catalogs (section 3.5). The developed software will be capable of detecting star triads (section 3.4) in image and producing an accurate attitude solution. Also different problems affecting the software will be discussed in detail. A study about two different attitude determination algorithms are also performed and out of this one which suits for the thesis work is selected at the end.

1.4.2 Thesis Framework

The whole work for the thesis is done in MATLAB, which is a powerful mathematical simulation tool used by most of the space engineers and aerospace companies. The images that are publicly available is centroided [1] with the help of algorithms and the star co-ordinates in the images are calculated according to the camera frame. The position of the camera frame is assumed since there is no original camera builds for the time being. The attitude is measured from comparing the inertial frame vectors and the vectors observed in the star camera frame and then the TRIAD algorithm [3] is used to find out a attitude solution in the form of quaternion. The algorithm is tested on various images to make sure that the performance matches the requirement. Once the software is successfully tested on a computer it can be converted to embedded code and loaded in the star sensor. The highlighted portion in the image represents the complete work for the thesis.

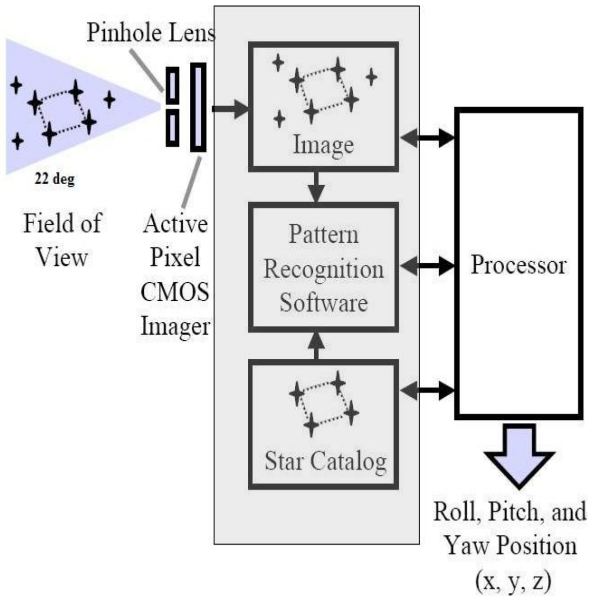


Figure 1.2: Thesis Frame work,Image[17]

Chapter 2

Co-ordinate frames

To begin with any attitude determination problem we should select an appropriate coordinate system . Any coordinate system can be used in defining orientation. But most of the times the problem is generated due to improper usage of coordinate frames. This section gives a brief overview about the different coordinate systems.

2.1 ECI

The Earth Centred Inertial coordinate frame is oriented with respect to the sun. The X-axis is the one which points along the equatorial plane towards the sun at the vernal equinox or the first point of Aries. The Z-axis points towards the earth's rotation axis and Y-axis forms the right handed system. The earth's axis and the vernal equinox precess around the pole of earth's orbit with a period of 26,000 yrs.

The precession of equinoxes result in a shift of the position of the vernal equinox relative to the fixed stars at a rate of 0.014 deg/yr. Because of this slow drift, celestial coordinates require a corresponding date to accurately define the position of the equinoxes. The most common systems are 1950 coordinates, J2000 coordinates and true of date or TOD. In this thesis work the co-ordinates used for stars are from J2000 Epoch.

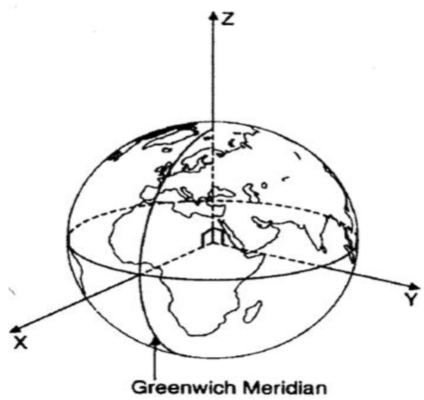


Figure 2.1: ECI Frame, Image [4]

2.2 Body (satellite)

The body frame is the coordinate system which is attached to the satellite or spacecraft body. This is fixed for each satellite. The X-axis is along the nominal velocity direction with Z- axis pointing towards the earth along nadir and the Y-axis forming the right handed coordinate system. Usually this coordinate system is used in problems dealing with the position of spacecraft instruments with reference to spacecraft body.

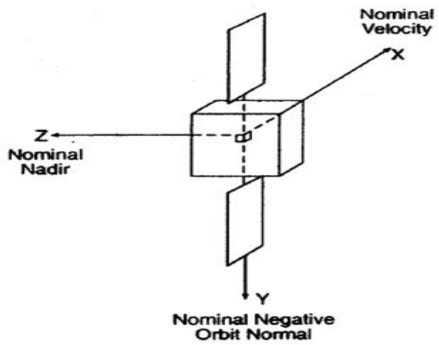


Figure 2.2: Body Coordinate frame, Image [4]

2.3 Camera frame

The camera frame is attached to the camera FoV and its bore sight lies along the Z-axis. The X-axis and Y-axis lies along the image plane axis and is fixed on the centre. Normally the star sensor is fixed on the spacecraft and

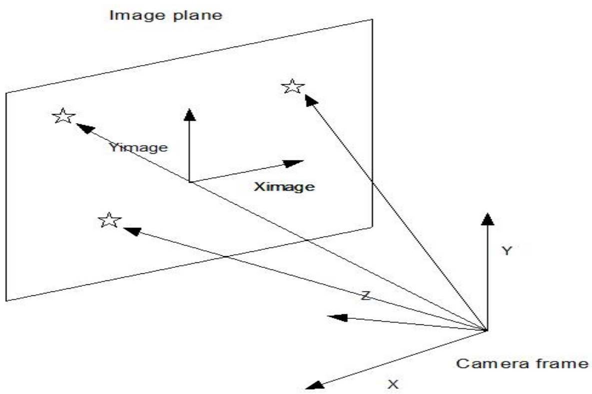


Figure 2.3: Camera Frame

a matrix transformation is required to change the coordinates of the image plane into the body frame. In this thesis the camera frame and the body frame are assumed to be at the same point since it makes the computation simpler. Once the camera is build and assembled on the satellite this matrix transformation can be augmented to obtain the coordinates in the satellite (body) frame.

2.4 Rotation Matrix and Quaternion

2.4.1 Rotation matrix

The three axis attitude transformation is based on the direction cosine matrix. Any attitude transformation in space is actually converted to this essential form [5]. In the figure, the axes **1**,**2** and **3** are unit vectors defining an orthogonal right handed triad. This triad is chosen as the reference inertial frame. Next an orthogonal triad is attached to the centre of mass of a moving body defined by the unit vectors **u**, **v** and **w**. The rotation matrix [**A**] is defined as following,

$$[\mathbf{A}] = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ \mathbf{w}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} \tag{2.1}$$

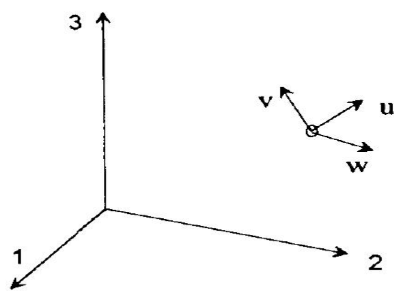


Figure 2.4: Definition of the orientation of the spacecraft axes **u**, **v** **w** in the reference frame **1**,**2**,**3**, Image [5]

In this $\mathbf{u_1}$, $\mathbf{u_2}$ and $\mathbf{u_3}$ are the components of vector **u** along the axes **1**,**2**,**3** of the reference orthogonal system, $\mathbf{u}$=$[\mathbf{u_1},\mathbf{u_2},\mathbf{u_3}]^{\mathrm{T}}$. In a similar way **v** and **w** also have components $\mathbf{v_1}$, $\mathbf{v_2}$ and $\mathbf{v_3}$ and $\mathbf{w_1}$, $\mathbf{w_2}$ and $\mathbf{w_3}$ along the same reference axes, $\mathbf{v}$=$[\mathbf{v_1},\mathbf{v_2},\mathbf{v_3}]^{\mathrm{T}}$ and, $\mathbf{w}$=$[\mathbf{w_1},\mathbf{w_2},\mathbf{w_3}]^{\mathrm{T}}$. The direction cosine matrix is also called attitude matrix and have an important property of mapping vectors from the reference frame to the body frame. Suppose that a vector, $\mathbf{a}$=$[\mathbf{a_1},\mathbf{a_2},\mathbf{a_3}]^{\mathrm{T}}$. The component of vector **a** in the body frame is given by,

$$[\mathbf{A}]a = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \mathbf{u}_3 \\ \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ \mathbf{w}_1 & \mathbf{v}_2 & \mathbf{v}_3 \end{bmatrix} \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_1 \\ \mathbf{a}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{a}_\mathrm{u} \\ \mathbf{a}_\mathrm{v} \\ \mathbf{a}_\mathrm{w} \end{bmatrix} = \mathbf{a_B} \tag{2.2}$$

where $\mathbf{a_B}$ is the vector **a** mapped into the body frame. Since **u** is a unit vector it follows that the scalar product, **u.a** is the component of the vector **a** along the unit vector of the body triad $\mathbf{a_v}$ and $\mathbf{a_w}$

2.4.2 Quaternion

The quaternion's basic definition is a consequence of the properties of the direction cosine matrix [**A**]. It is shown by linear algebra that a proper real orthogonal 3 × 3 matrix has at least one eigen vector with eigen value of unity. This means that since one of the eigen values, $\lambda_i (i = 1, 2, 3)$ is unity , the eigen vector is unchanged by the matrix,$[\mathbf{A}]\mathbf{e_i}$=1$\mathbf{e_i}$

The eigen vector, $\mathbf{e_i}$ has the same components along the reference frame axis. The existence of such a vector is the analytical demonstration of Euler's theorem [5]. It can be demonstrated that any attitude transformation in space by consecutive rotations about the three orthogonal frames can be attained by a single rotation about the eigen vector with unit eigen value. The quaternions is defined as a vector in which,

$$\mathbf{q} = q_4 + \mathbf{iq_1} + \mathbf{jq_2} + \mathbf{kq_3} = q_4 + \mathbf{q} \tag{2.3}$$

where the unit vectors **i**,**j**,**k** satisfy the following conditions,

$$\mathbf{i^2} = \mathbf{j^2} = \mathbf{k^2} = -1, \tag{2.4}$$

$$\mathbf{ij} = -\mathbf{ji} = \mathbf{k}, \tag{2.5}$$

$$\mathbf{jk} = -\mathbf{kj} = \mathbf{i}, \tag{2.6}$$

$$\mathbf{ki} = -\mathbf{ik} = \mathbf{j}. \tag{2.7}$$

In this definition q_4 is a scalar and **q** is a vector. The elements of quaternion sometimes called the Euler symmetric parameters can be expressed in terms of the principal eigen vector $\mathbf{e}$. The parameters are calculated by,

$$\mathbf{q_1} = \mathbf{e_1}\sin(\alpha/2), \tag{2.8}$$

$$\mathbf{q_2} = \mathbf{e_2}\sin(\alpha/2), \tag{2.9}$$

$$\mathbf{q_3} = \mathbf{e_3}\sin(\alpha/2), \tag{2.10}$$

$$\mathrm{q}_4 = \cos(\alpha/2) \tag{2.11}$$

Clearly,

$$\mathbf{q_1}^2 + \mathbf{q_2}^2 + \mathbf{q_3}^2 + q_4{}^2 = 1, |\mathbf{q}| = 1 \tag{2.12}$$

Using the above equations the direction matrix can be expressed in terms of the quaternion,

$$[\mathbf{A(q)}] = (\mathrm{q}_4{}^2 - \mathbf{q_2}^2)1 + 2\mathbf{q}\mathbf{q}^\mathbf{T} - 2\mathrm{q}_4[\mathbf{Q}], \tag{2.13}$$

where,

$$[\mathbf{Q}] = \begin{bmatrix} 0 & -\mathbf{q_3} & \mathbf{q_2} \\ \mathbf{q_3} & 0 & -\mathbf{q_1} \\ -\mathbf{q_2} & \mathbf{q_1} & 0 \end{bmatrix} \tag{2.14}$$

and

$$[\mathbf{A(q)}] = \begin{bmatrix} \mathbf{q_1}^2 - \mathbf{q_2}^2 - \mathbf{q_3}^2 + \mathrm{q}_4{}^2 & 2(\mathbf{q_1}\mathbf{q_2} + \mathbf{q_3}\mathrm{q}_4) & 2(\mathbf{q_1}\mathbf{q_3} - \mathbf{q_2}\mathbf{q_3}) \\ 2(\mathbf{q_1}\mathbf{q_2} - \mathbf{q_3}\mathrm{q}_4) & -\mathbf{q_1}^2 + \mathbf{q_2}^2 - \mathbf{q_3}^2 + \mathrm{q}_4{}^2 & 2(\mathbf{q_2}\mathbf{q_3} + \mathbf{q_1}\mathrm{q}_4) \\ 2(\mathbf{q_1}\mathbf{q_3} + \mathbf{q_2}\mathrm{q}_4) & 2(\mathbf{q_2}\mathbf{q_3} - \mathbf{q_1}\mathrm{q}_4) & -\mathbf{q_1}^2 - \mathbf{q_2}^2 + \mathbf{q_3}^2 + \mathrm{q}_4{}^2 \end{bmatrix} \tag{2.15}$$

Equation (2.10) allows us to express the quaternion in terms of direction cosine matrix and vice versa. A total of four such sets can be defined for the

Parameter	Specification	Justification
Magnitude (M)	6	4-7
Focal length (f)	20mm	
Acquisition time for matching stars	0.25s	
Accuracy (star position and magnitude)	0.02 all 3 axis	
Update rate for tracking stars	4Hz	
Minimum number of stars	3	for triad construction

Table 2.1: Requirements for a star sensor

quaternion. For the first set q_4 is found by summing the diagonal elements a_{11},a_{22} and a_{33}. To find $\mathbf{q_1}$, $\mathbf{q_2}$ and $\mathbf{q_3}$ take the sums a_{234}-a_{32}, a_{31}-a_{13}, and a_{12}-a_{21}. The solution is,

$$q_4 = .5\sqrt{(1 + a_{11} + a_{22} + a_{33}),} \tag{2.16}$$

$$\mathbf{q_1} = .25(a_{23} - a_{32})/q_4, \tag{2.17}$$

$$\mathbf{q_2} = .25(a_{31} - a_{13})/q_4, \tag{2.18}$$

$$\mathbf{q_3} = .25(a_{12} - a_{21})/q_4 \tag{2.19}$$

In this work the output solution is given in terms of quaternion, which can be interchanged to rotation matrix and vice versa.

2.5 Requirements for a star sensor

Before the software development is done the requirements has to be clearly defined. The table above shows the requirements of a star sensor. Out of the requirements mentioned FoV, magnitude and minimum number of stars are the important factors for the software development. The focal length specification is dependent on the optics developed. Rest of the specifications depend on the final hardware on which the software is implemented. So extreme care has to be taken in finding good hardware which support the software operations developed in the algorithm.

2.5.1 Field of View (FoV)

The field of view is defined as the angular extent to which the sky is observable at any specific given direction. This is an important factor since it determines the number of stars that is observable by the camera in a single image. The number of star required for pattern matching depends upon the algorithm developed and is defined in the forth coming chapters. The more number of stars that can be observed in a image the more chance of the star pattern recognition and hence the speed of the algorithm also increases. The speed of algorithm is important in case where quick attitude determination is required. Hence field of view is important to both hardware and software of camera. A thorough analysis was performed with the help of Matlab simulation to find out field of view that is required by the camera.

2.5.2 Simulations for field of view (Monte Carlo Analysis)

The determination of field of view is a complex process so that it was done with the help of a computer simulation. The simulation finds out how many stars are present in an image for a certain magnitude and angle in a particular direction of the sky and checks if it can guarantee enough stars required for pattern matching. This is done with help of extracting the Right ascension (RA), Declination (Dec) and magnitude values of each stars from the catalog. The catalog used was the hipparcos catalog which has stars up to magnitude of 11. They are truncated with a star magnitude of 6 and the rest of stars brighter than 6 which numbers up to 5029 are used for simulations. An initial estimate of magnitude showed that stars brighter than 6 would be enough to guarantee the number of required stars in any directions. The simulation was done for a maximum field of view of 22 degrees and magnitude from 1 to 6. Normally stars brighter than magnitude six are used in small and low cost star sensor because the cost and effort increases as the stars to be detected are more fainter. The spherical co-ordinates which are retrieved from catalog as RA and Dec are converted into unit vector in cartesian coordinates and then recreated on imaginary celestial sphere. The imaginary celestial sphere is of unit radius and the camera frame is assumed to be in centre of the earth and can be pointed to any random pointing direction of the sky. The distribution of stars on the unit celestial sphere is shown below in 3D Plot.

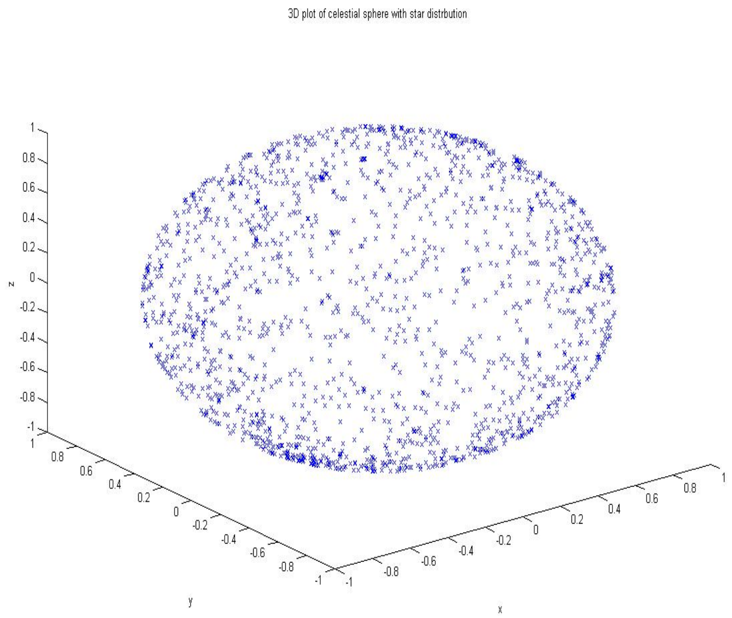


Figure 2.5: 3D Plot of celestial sphere with star distribution

Once the camera is fixed in the centre of the sphere it can be pointed to any direction by generating a random unit vector in any direction. The unit vector represents a specific direction in the celestial sphere and this unit vector is along the optical axis of the camera. The star's unit vector which lies in the field of view of the camera image plane can be calculated by the formula given below. The formula calculates the angle between the star unit vector and the optical axis of the camera. If the angle between those vectors are less than half of the field of view then star is considered to be lying inside the camera frame and rest of the stars are not counted. The simulation gives the number of stars lying in a circular field of view around the optical axis. This can be approximated to the rectangular field of view for calculation purposes. The unit vector of each star is obtained by converting the spherical coordinates into the Cartesian coordinate system.

To find out whether the stars are lying in field of view or not of the camera the following formula is used,

$$\mathbf{a} \cdot \mathbf{b} = |\mathbf{a}||\mathbf{b}|\cos(\theta) \tag{2.20}$$

Since both of the vectors are with unit magnitude the formula reduces to the following,

$$\cos(\theta) = \mathbf{a} \cdot \mathbf{b} \tag{2.21}$$

and,

$$\theta = \mathrm{acos}(\mathbf{a} \cdot \mathbf{b}) \tag{2.22}$$

2.5.3 Simulation results

Monte Carlo simulations [6] are performed to find out the appropriate field of view for the camera. The simulations are done for 10,000 times in random direction and with a field of view of 8 degree, 10 degree, 17 degree and 22 degree and was conducted for a maximum magnitude cut-off value of 6. Even though the minimum number of stars required in one image is three (star triad construction) a threshold value of five is kept for redundancy reasons so that in case the stars that are not detected or are missed due to errors can be rectified. A margin of two stars is a good option for the star pattern matching algorithm.

2.5.4 Magnitude

The magnitude is an important characteristic of the star which depends upon the amount of light emitted and the distance which it has to travel to reach the sensor. The magnitude of a star is given by the formula, M=2.5 log(I/Io), I-effective radiance, Io-Effective standard radiance. With naked eyes and clear sky, stars up to a magnitude of 6 can be seen from earth. In this thesis

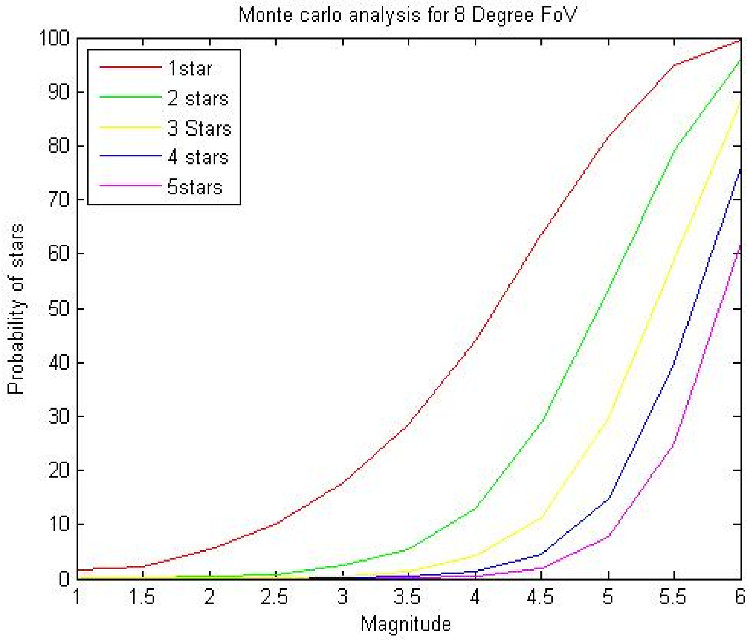


Figure 2.6: 8 Degree FoV

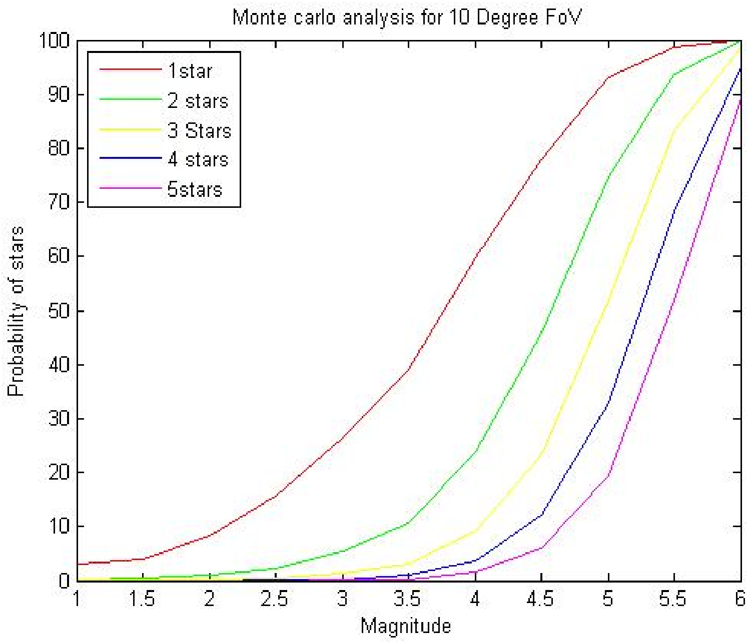


Figure 2.7: 10 Degree FoV

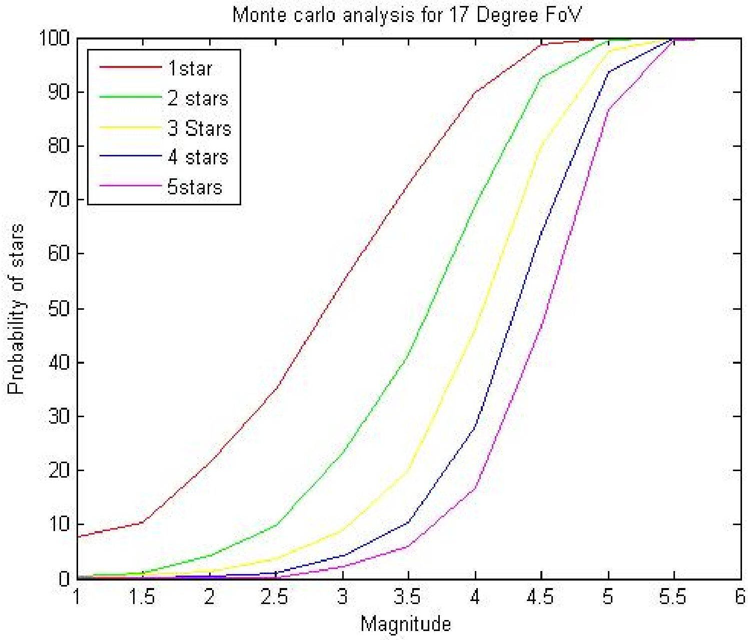


Figure 2.8: 17 Degree FoV

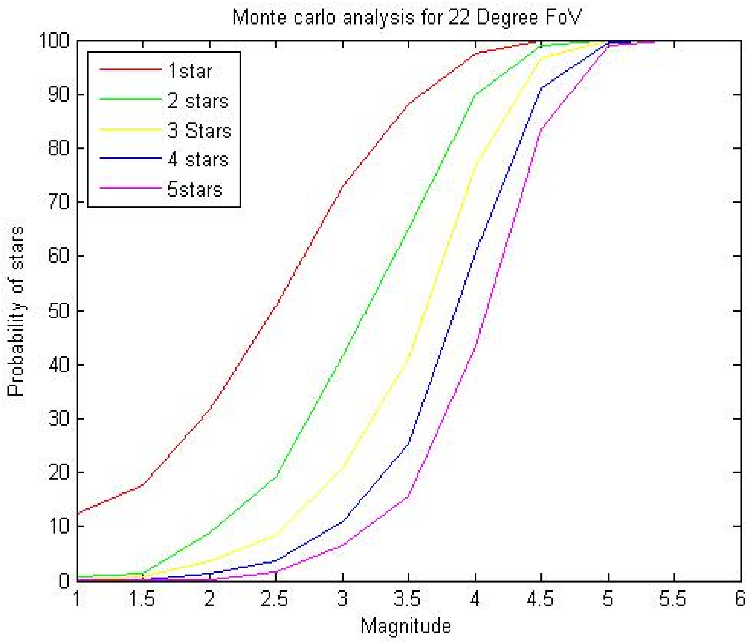


Figure 2.9: 22 Degree FoV

work the catalog used for the star pattern identification is the Hipparcos catalog which contains stars up to a magnitude of 11 and the numbers of stars present are 114936. Clearly it is not possible to process the whole number of stars in the algorithm since it requires huge amount of memory, processing and hence power. There should be a clear trade-off between the magnitude, number of stars and hence the field of view. If stars of magnitude more than 8 or 9 are to be detected more costly sophisticated image sensors are required. So the stars should be truncated at a certain magnitude value. The figure below gives an idea about how the number of stars in catalog increases exponentially as the magnitude increases.

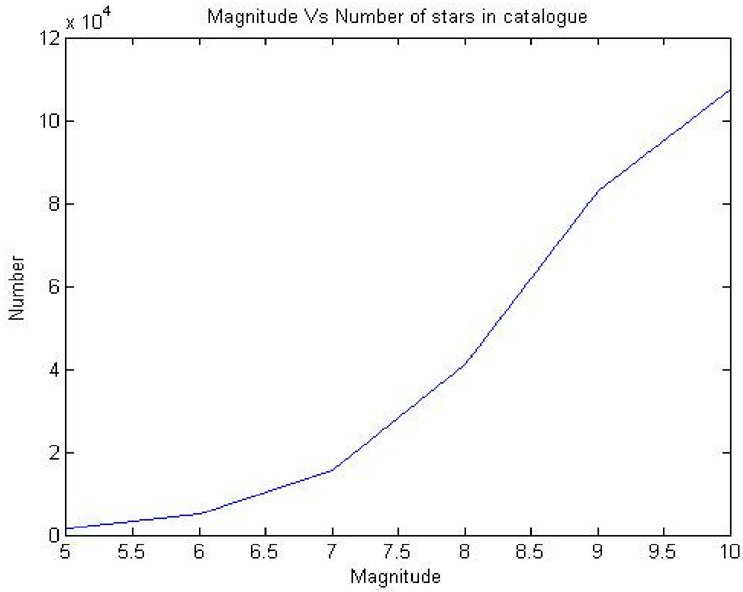


Figure 2.10: Magnitude Vs Number of stars in catalog

From the Monte Carlo simulations and the above figure it became clear that a magnitude of six can guarantee a minimum number of five stars in any direction with a field of view 22 degree. So the catalog was truncated at magnitude of six which produced 5029 stars in total. This truncated catalog is used as the base catalog

Chapter 3

Star detection and pattern Recognition methods

The star pattern matching technique is the initial process for the star sensor attitude determination. The stars are fixed in the inertial reference frame forever and maintains a fixed pattern which remains the same. So they can be used to identify star pattern. This is the way how humans have been identifying constellation for the past centuries. The same technique can be used for the star sensors also. The pattern can be detected from the image and cross checked with the pattern build in catalog. So building up of the catalog depends upon the selection of the proper star pattern method. Each of the methods has got its own advantages and disadvantages which are discussed in the following sections.

3.1 Star detection

In most cases the raw image captured by the camera is corrupted with noise and motion blur [18]. From this corrupted noise image star like features are to be found out so as to construct the triad. The stars detected might be true stars or false stars (which are due to other bright sources). Even if a false star is detected a triad can be constructed, but will not be found in the catalog. If a true star is missed it completely destroys the process of triad construction. The detection of stars highly depend upon the thresholding applied on the image.

3.1.1 Thresholding

In the process of thresholding, any pixel whose brightness is more than a specific value is accepted as a star. The threshold value is chosen empirically and scaled proportionally to the exposure time for real-time operations. This method is very simple since it doesn't need a memory buffer in processing. But the major disadvantage is it limited ability to distinguish between stars and noise. If a low enough threshold is chosen to bring out the faintest stars (low thresholding), then many noise artefacts are falsely detected. If the threshold is raised to reject the noise (high thresholding), then many faint stars are missed. In case of very bright stars nearby pixels are also detected as stars due to the light spreading from optic lens. But this can be eliminated in centroiding and setting a minimum distance between two stars in the image. The images shown below explain the effect of different levels

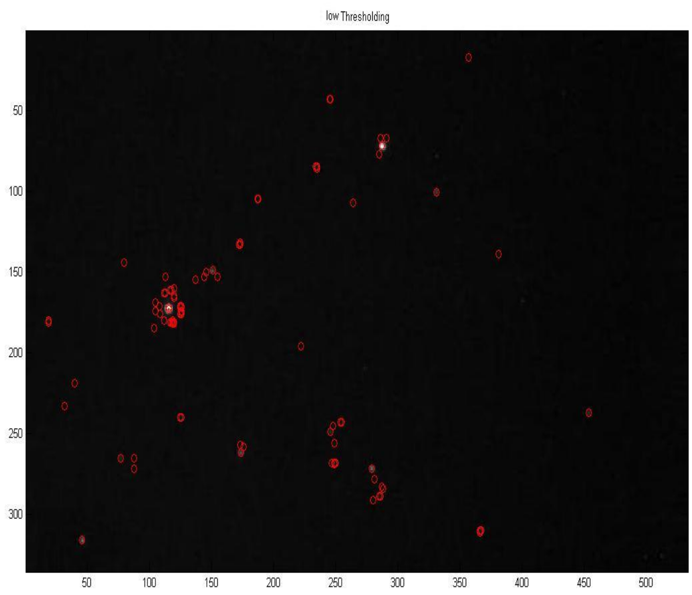


Figure 3.1: Stars detected for low thresholding

Figure 3.2: Stars detected for high thresholding

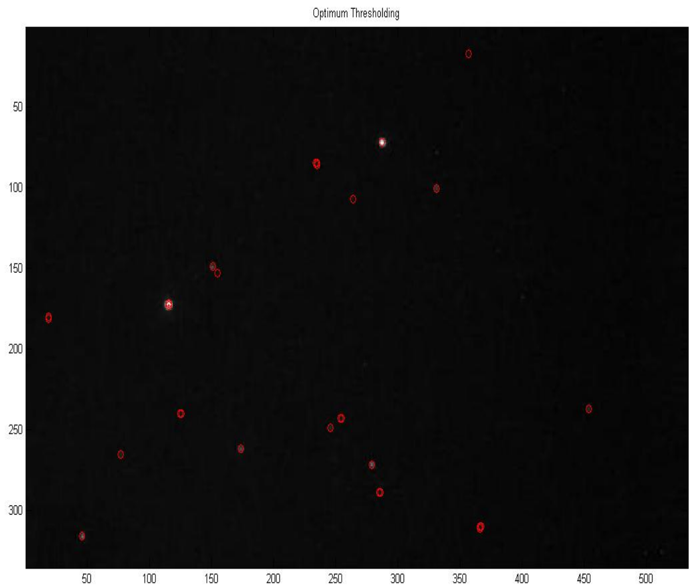


Figure 3.3: Stars detected for optimum thresholding

of thresholding applied to the same image.

3.2 Different methods of recognition

The stars which are identified in the image can be the same for different algorithms, but patterns can be built in different ways according to the software programmer. The selection of the method is dependent on the number of stars that are detected in image and the size of catalog that will be build up. The method developed should be such that it reduces the ambiguity among the patterns build up i.e. there should be less similarity among the data that are required to find out the star pattern. In all the three methods a planar triangle which consists of three stars are used in different ways for recognition.

3.2.1 Area and Polar moment method

The triad consists of three stars which form a triangle. For every star in the catalog a triad is created and the area and moment is also calculated and stored in a Look up Table. While the stars triads are detected in the image, the same features are calculated and compared with the Look Up Table. If a match occurs the star are considered to be detected. The areas and moments are calculated by the formulas given below,

$$\mathrm{Area} = \sqrt{\mathrm{s(s-a)(s-b)(s-c)}} \tag{3.1}$$

$$\mathrm{Moment} = \mathrm{Area(a^2+b^2+c^2)/36} \tag{3.2}$$

$$\mathrm{s} = (\frac{1}{2}\mathrm{(a+b+c)} \tag{3.3}$$

$$\mathrm{a} = |\mathbf{V_1}| - |\mathbf{V_2}| \tag{3.4}$$

$$\mathrm{b} = |\mathbf{V_2}| - |\mathbf{V_3}| \tag{3.5}$$

$$\mathrm{c} = |\mathbf{V_1}| - |\mathbf{V_3}| \tag{3.6}$$

The vectors $\mathbf{V_1}$,$\mathbf{V_2}$, $\mathbf{V_3}$ represent the vectors from the inertial reference

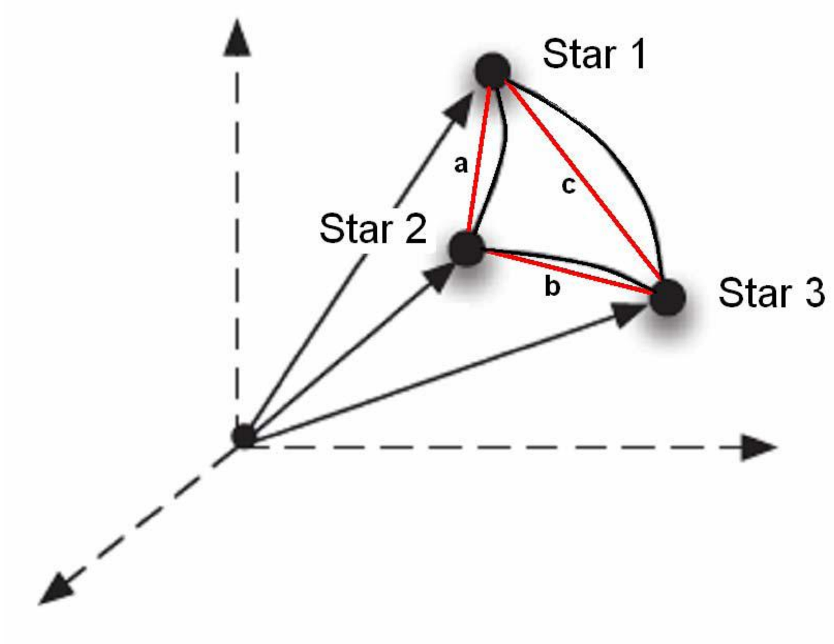


Figure 3.4: Area Polar method [7]

frame to the stars and a, b and c represent the side of the planar triangles. The vectors used while building the catalog is in inertial frame and while detection is in camera frame [7].

3.2.2 Planar moment method

This method uses the angle between the sides of planar triangles to identify the pattern. The angle between each side of the triangle is calculated by the formulas given below [7],

$$\mathbf{a} = \mathbf{V_1} - \mathbf{V_2} \tag{3.7}$$

$$\mathbf{b} = \mathbf{V_2} - \mathbf{V_3} \tag{3.8}$$

$$\mathbf{c} = \mathbf{V_1} - \mathbf{V_3} \tag{3.9}$$

$$\theta_1 = \mathbf{a}.\mathbf{b}/(|\mathbf{a}||\mathbf{b}|) \tag{3.10}$$

$$\theta_2 = \mathbf{b}.\mathbf{c}/(|\mathbf{b}||\mathbf{c}|) \tag{3.11}$$

$$\theta_3 = \mathbf{a}.\mathbf{c}/(|\mathbf{a}||\mathbf{c}|) \tag{3.12}$$

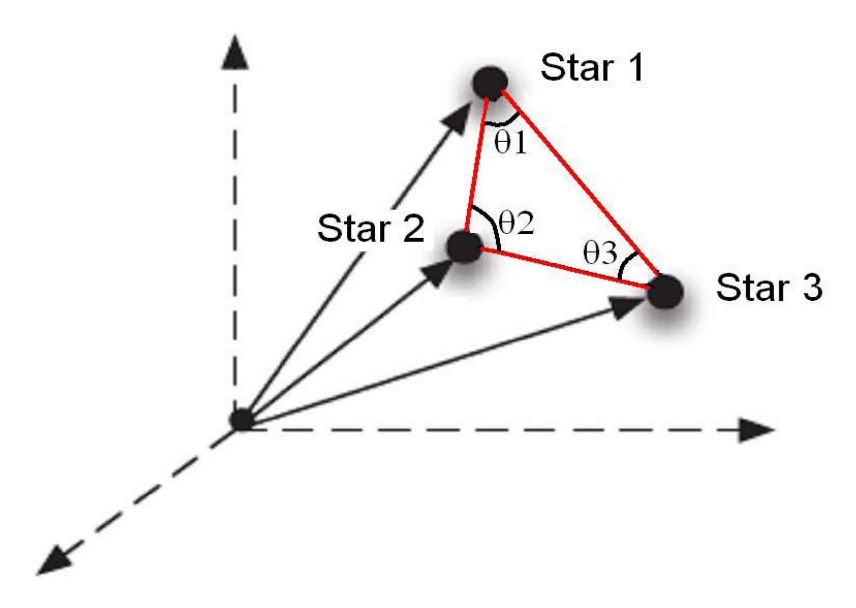


Figure 3.5: Planar moment method [7]

The angles are found in the inertial frame and stored in the look up table. While performing comparison the $\mathbf{V_1}$, $\mathbf{V_2}$ and $\mathbf{V_3}$ is considered as the vectors in the camera frame .

3.2.3 Vector angle method

In this method the angle between the two vectors $\mathbf{V_1}$ and $\mathbf{V_2}$, $\mathbf{V_2}$ and $\mathbf{V_3}$, $\mathbf{V_1}$ and $\mathbf{V_3}$ are calculated for each planar triangle and stored in the look up table [7]. The angles are calculated by the formula given below,

$$\theta = \text{acos}(\mathbf{V_1}).(\mathbf{V_2})/(\mathbf{V_1}||\mathbf{V_2}|) \tag{3.13}$$

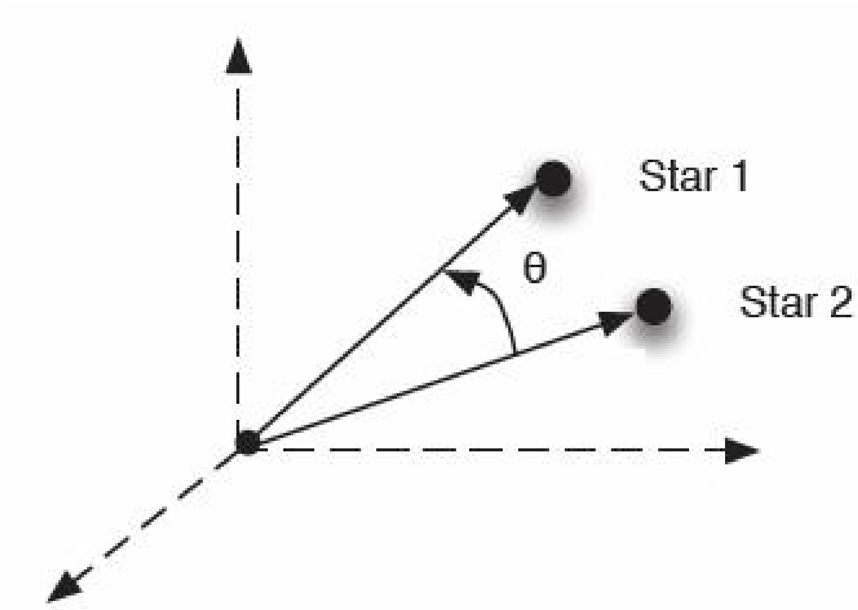


Figure 3.6: Vector angle method [7]

The angles are calculated according to the inertial reference frame and stored in look up table.

In this thesis the Planar moment method is used for triad detection.

3.3 Centroiding

Centroiding is a fundamental process required in star pattern identification. This is done so as to correctly locate the star centre in the star image frame. The image processing for star images are done to digitize the values obtained for CCD and then reduced to a small observable set of values. These are important for the further processing of the image. The algorithm operates on the image and detects the bright spots in the image. But usually the light is spread across an array of the CCD and it is difficult to locate the exact centre of the star. This is mainly due to the spreading and defocusing of the light while travelling though the optics of the camera. A point spreading function can measure the amount of light that has spread across the CCD array. The magnitude of the star and the spreading of the light determines the size of the arrays that must be used in the Centroiding algorithm. Usually it is 3×3 to 15×15. The modern star centroiding algorithm helps to find the star image centre upto a precision on 1/10 of a pixel or even better. The image centroid is thus important since it affects the accuracy of the measurement and hence the whole process.

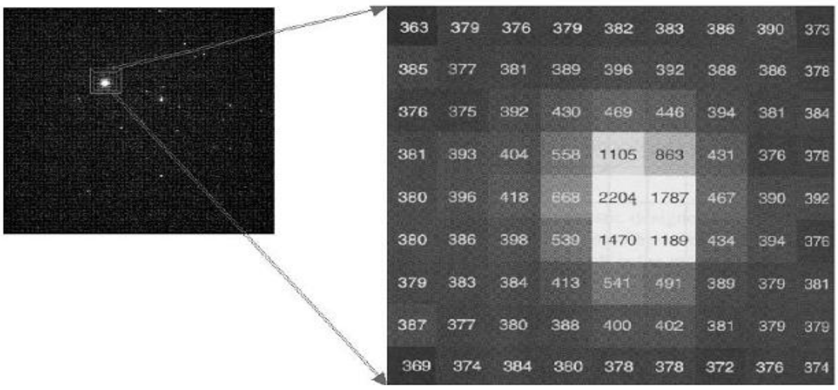


Figure 3.7: Centroiding Image [1]

3.3.1 Centroiding algorithm

The particular centroiding algorithm for this thesis work focuses star light from the nearby 3×3 pixels and finds the exact star centre up to a sub pixel accuracy of 1/10. The star centre is found out by detecting pixels which have a threshold value greater than a certain pre defined specific value. The pixel should also have a value greater than it eight neighbouring pixels so as to ensure that the centroiding is done at the right star centre. The value is defined during calibration of the camera. But in this case since there is no real camera the threshold values are set different for different images. In addition, any location that has 3 or more adjacent pixels of the maximum intensity value is discarded and not considered as a star. This is because the centroiding algorithm may be hard to apply to a very bright object resulting in an inaccurate star location. Once the star is detected the surrounding values of the centre pixels is read and centroid is calculated from the formula given below. The centroiding algorithm outputs centroid offset values (X_c, Y_c) that represent the offset of the calculated star location from the centre of the star pixels of the 3 by 3 array. The centroid X_c and Y_c can be found out by summing up the product of intensities and pixel locations given by,

$$X_c = \frac{\sum_{i=1}^{3}\sum_{i=1}^{3} \mathrm{I(i,j).i}}{\sum_{i=1}^{3}\sum_{i=1}^{3} \mathrm{I(i,j)}} \tag{3.14}$$

$$Y_c = \frac{\sum_{i=1}^{3}\sum_{i=1}^{3} \mathrm{I(i,j).j}}{\sum_{i=1}^{3}\sum_{i=1}^{3} \mathrm{I(i,j)}} \tag{3.15}$$

3.3.2 Centroiding results

The centroiding was applied to the image before detection and the results obtained are shown below. The figure shows the normal star image whose centroid is not calculated. The centre would have been at, X_c=287, Y_c=72 if not centroided. An offset value of, X_c=0.1150, Y_c=0.589 occurred due to centroiding. These values are represented graphically.

3.4 Star Triad

In order to recognize an image from the camera, some characteristics that distinguishes between the stars must be determined. The camera can't cover the whole sky or a big constellation like Orion or Big dipper, due to its limited field of view of 22 degree. So some local features has to be determined for identification[8]. Among these are the,

- Angular distance to the closest neighbouring star
- Spherical angles between the closest stars
- The magnitude

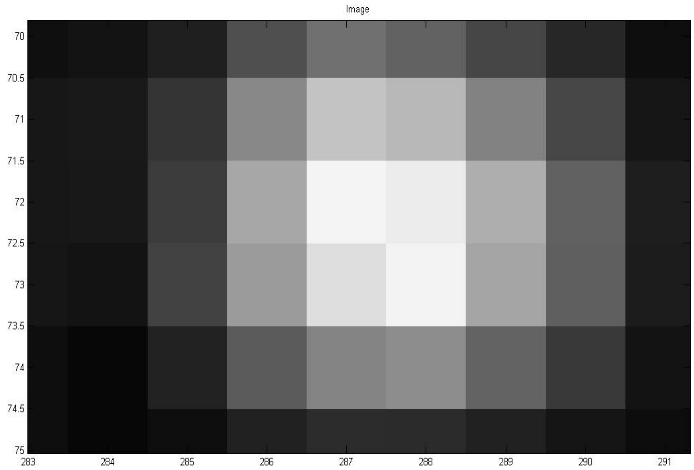


Figure 3.8: Image without centroiding

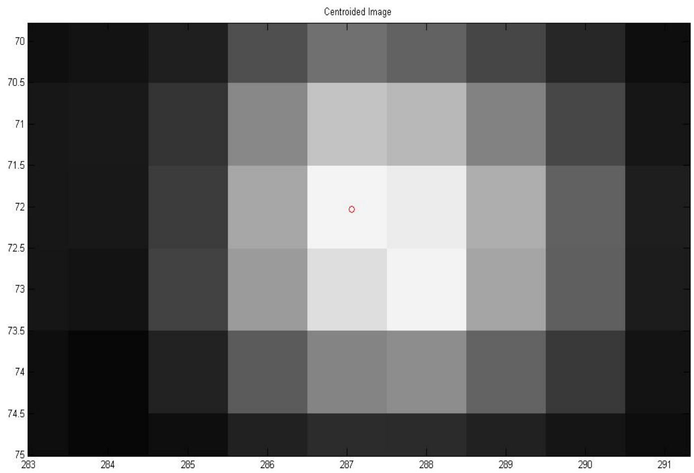


Figure 3.9: Centroided Image

The common feature to characterize a star and its nearest surroundings is the angular distances to the closest neighbours. Figures given below show the distribution of angular distance for the first and second neighbouring stars. Another feature to characterize a star is its magnitude. However brightness as such is not applicable in the present case because noise limits the precision and it increases uncertainty for fainter stars. Thus the distance to the first and second neighbour and the angle between them have been chosen as the characterizing parameters for the database (Figure 6). A sub constellation with these features is named a star triplet and its parameter values are called the Triangular Feature (TF).

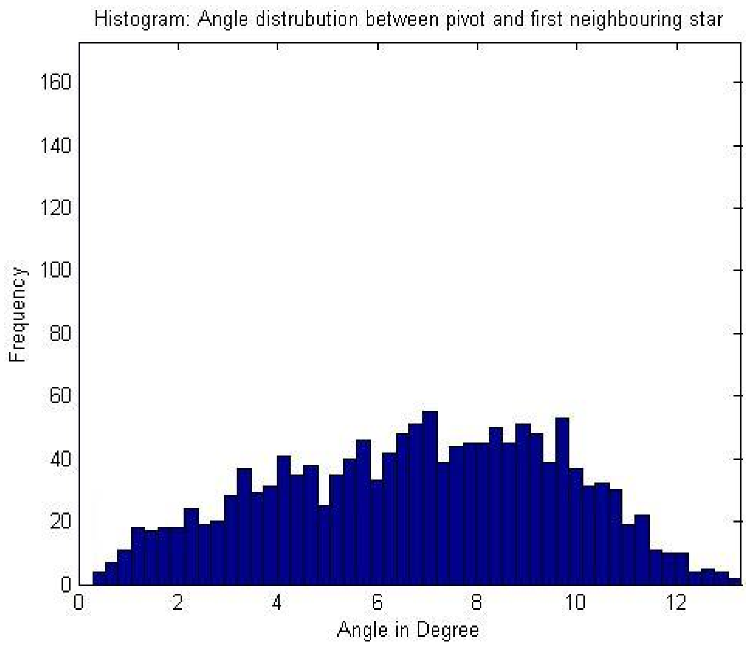


Figure 3.10: Angle distribution between pivot and first star

Normally the triads are selected in a specific order. The first star selected is the brightest star in the image and then the second star is the second brightest star and the third star is the third brightest star.

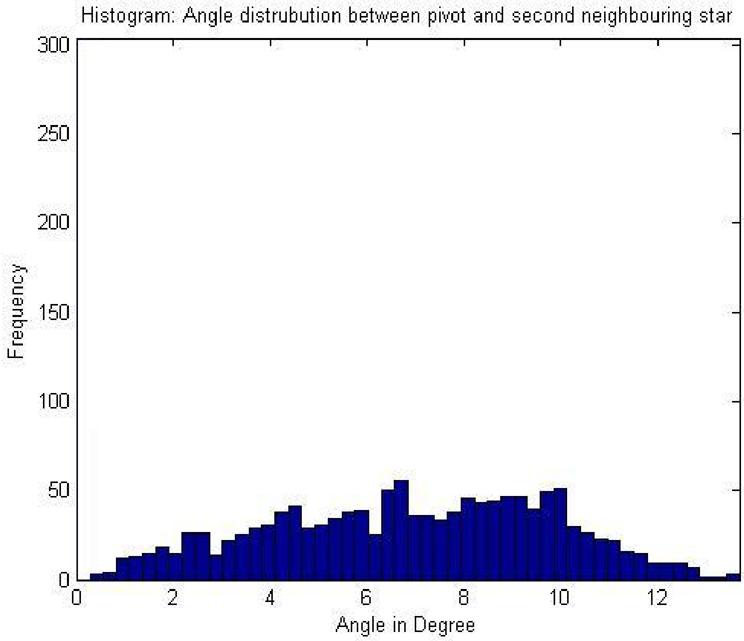


Figure 3.11: Angle distribution between pivot and second star

3.5 Catalogs

A star catalog is the database where the algorithm searches to find a match for the star pattern. The star catalog contains data such as Right Ascension, Magnitude and the Triangular Feature as mentioned before. In this thesis work the two different catalogs are used and they are built from the main hipparcos catalog. The star catalog can be used in different ways to detect the star pattern either by utilizing the stellar coordinates, or stellar magnitudes or both. But in this case the stellar magnitudes were not used in the triad detection since the quality of the imager is not as the same as an expensive star tracker and in course of time there are high chance that the camera optics might degrade and the image detected will not be of good quality. The magnitude were used to detect the stars in the image, but not during triad creation or searching or identifying process.

The Hipparcos catalog [9] is the primary product of the European Space Agency's astrometry mission, Hipparcos. The satellite, which operated for four years, returned high quality scientific data from November 1989 to March

1993. The Hipparcos catalog lists a little more than 118,000 stars with 1 to 3 milli arc-sec level astrometry. The Hipparcos catalog is complete to magnitude 11.

3.5.1 Hipparcos catalog

A short description of the details in the Hipparcos catalog are given below [10]

- StarID: The database primary key from a larger master database of stars.
- HD: The star's ID in the Henry Draper catalog, if known.
- HR: The star's ID in the Harvard Revised catalog, which is the same as its number in the Yale Bright Star Catalog.
- Gliese: The star's ID in the third edition of the Gliese Catalog of Nearby Stars.
- BayerFlamsteed: The Bayer Flamsteed designation, from the Fifth Edition of the Yale Bright Star Catalog. This is a combination of the two designations.
- RA, Dec: The star's right ascension and declination, for epoch 2000.0. Stars present only in the Gliese Catalog, which uses 1950.0 coordinates, have had these coordinates precessed to 2000.
- Proper Name: A common name for the star, such as Barnard's Star or Sirius.
- Distance: The star's distance in parsecs, the most common unit in astrometry. To convert parsecs to light years, multiply by 3.262. A value of 10000000 indicates missing or dubious (e.g., negative) parallax data in Hipparcos.
- Mag: The star's apparent visual magnitude.

- AbsMag: The star's absolute visual magnitude (its apparent magnitude from a distance of 10 parsecs).

- Spectrum: The star's spectral type, if known.

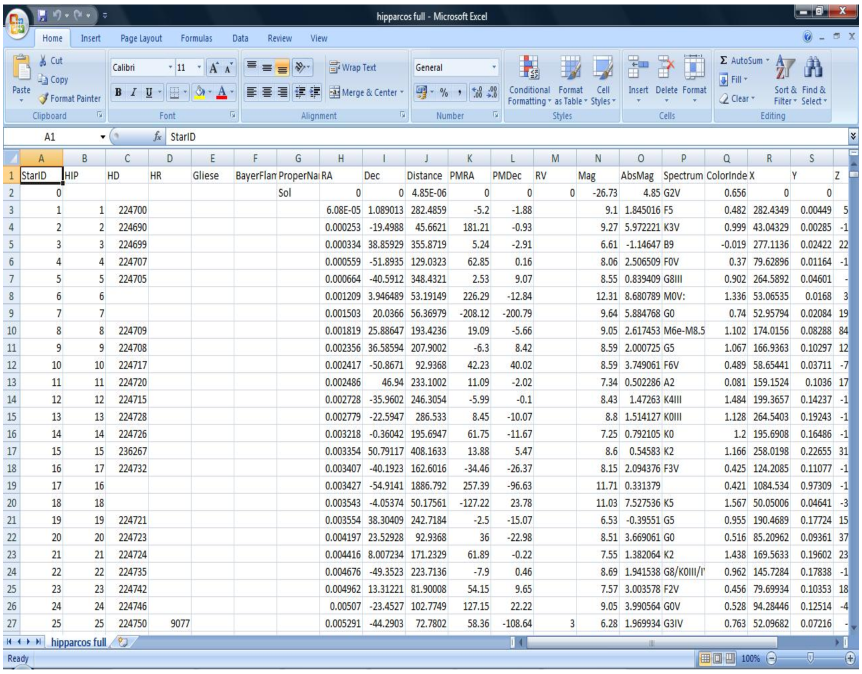
- ColorIndex: The star's color index (blue magnitude - visual magnitude), where known.

- X,Y,Z: The Cartesian coordinates of the star, in a system based on the equatorial coordinates as seen from Earth. +X is in the direction of the vernal equinox (at epoch 2000), +Z towards the north celestial pole, and +Y in the direction of R.A. 6 hours, declination 0 degrees.

- VX,VY,VZ: The Cartesian velocity components of the star, in the same coordinate system described immediately above. They are determined from the proper motion and the radial velocity (when known). The velocity unit is parsecs per year; these are small values (around 10^{-5} to 10^{-6}), but they enormously simplify calculations using parsecs as base units for celestial mapping.

3.5.2 Base catalog

As mentioned in the previous sections a catalog with all the stars of magnitude up to 11 is very huge in number and searching the whole catalog becomes a tedious process for star pattern matching. So the catalog has to be truncated at some point to make it smaller but at the same time it should be able to find star pattern in all directions. The first step was to truncate the stars above a certain value of magnitude. From the Monte Carlo analysis it was clear that for a 22 degree field of view and a star magnitude of 6 would guarantee minimum of five stars in any direction. So the magnitude of the Hipparcos catalog was truncated at a magnitude 6, which left 5029 stars in catalog. Several other irrelevant data were also deleted from the catalog to produce the base catalog. Finally, the base catalog consists of the Star ID, RA and Dec and Mag (Magnitude).



StarID	HIP	HD	HR	Gliese	BayerFlan	ProperNa	RA	Dec	Distance	PMRA	PMDec	RV	Mag	AbsMag	Spectrum	ColorInde	X	Y	Z
0						Sol	0	0	4.85E-06	0	0	0	-26.73	4.85	G2V	0.656	0	0	
1	1	224700					6.08E-05	1.089013	282.4859	-5.2	-1.88		9.1	1.845016	F5	0.482	282.4349	0.00449	5
2	2	224690					0.000253	-19.4988	45.6621	181.21	-0.93		9.27	5.972221	K3V	0.999	43.04329	0.00285	-1
3	3	224699					0.000334	38.85929	355.8719	5.24	-2.91		6.61	-1.14647	B9	-0.019	277.1136	0.02422	22
4	4	224707					0.000559	-51.8935	129.0323	62.85	0.16		8.06	2.506509	F0V	0.37	79.62896	0.01164	-1
5	5	224705					0.000664	-40.5912	348.4321	2.53	9.07		8.55	0.839409	G8III	0.902	264.5892	0.04601	-
6	6						0.001209	3.946489	53.19149	226.29	-12.84		12.31	8.680789	M0V:	1.336	53.06535	0.0168	3
7	7						0.001503	20.0366	56.36979	-208.12	-200.79		9.64	5.884768	G0	0.74	52.95794	0.02084	19
8	8	224709					0.001819	25.88647	193.4236	19.09	-5.66		9.05	2.617453	M6e-M8.5	1.102	174.0156	0.08288	84
9	9	224708					0.002356	36.58594	207.9002	-6.3	8.42		8.59	2.000725	G5	1.067	166.9363	0.10297	12
10	10	224717					0.002417	-50.8671	92.9368	42.23	40.02		8.59	3.749061	F6V	0.489	58.65441	0.03711	-7
11	11	224720					0.002486	46.94	233.1002	11.09	-2.02		7.34	0.502286	A2	0.081	159.1524	0.1036	17
12	12	224715					0.002728	-35.9602	246.3054	-5.99	-0.1		8.43	1.47263	K4III	1.484	199.3657	0.14237	-1
13	13	224728					0.002779	-22.5947	286.533	8.45	-10.07		8.8	1.514127	K0III	1.128	264.5403	0.19243	-1
14	14	224726					0.003218	-0.36042	195.6947	61.75	-11.67		7.25	0.792105	K0	1.2	195.6908	0.16486	-1
15	15	236267					0.003354	50.79117	408.1633	13.88	5.47		8.6	0.54583	K2	1.166	258.0198	0.22655	31
16	17	224732					0.003407	-40.1923	162.6016	-34.46	-26.37		8.15	2.094376	F3V	0.425	124.2085	0.11077	-1
17	16						0.003427	-54.9141	1886.792	257.39	-96.63		11.71	0.331379		0.421	1084.534	0.97309	-1
18	18						0.003543	-4.05374	50.17561	-127.22	23.78		11.03	7.527536	K5	1.567	50.05006	0.04641	-3
19	19	224721					0.003554	38.30409	242.7184	-2.5	-15.07		6.53	-0.39551	G5	0.955	190.4689	0.17724	15
20	20	224723					0.004197	23.52928	92.9368	36	-22.98		8.51	3.669061	G0	0.516	85.20962	0.09361	37
21	21	224724					0.004416	8.007234	171.2329	61.89	-0.22		7.55	1.382064	K2	1.438	169.5633	0.19602	23
22	22	224735					0.004676	-49.3523	223.7136	-7.9	0.46		8.69	1.941538	G8/K0III/I'	0.962	145.7284	0.17838	-1
23	23	224742					0.004962	13.31221	81.90008	54.15	9.65		7.57	3.003578	F2V	0.456	79.69934	0.10353	18
24	24	224746					0.00507	-23.4527	102.7749	127.15	22.22		9.05	3.990564	G0V	0.528	94.28446	0.12514	-4
25	25	224750	9077				0.005291	-44.2903	72.7802	58.36	-108.64	3	6.28	1.969934	G3IV	0.763	52.09682	0.07216	-

Figure 3.12: Hipparcos Catalog Screen shot, [10]

3.5.3 Oriented catalog

The oriented catalog is constructed from the base catalog according to the developed algorithm in the thesis work. Considering a star its two closest neighbours $\mathbf{N}_1$ and $\mathbf{N}_2$ are detected and three vectors between them are constructed as $\mathbf{SN}_1$, $\mathbf{SN}_2$, $\mathbf{N}_1\mathbf{N}_2$. The three angles between each of the vectors are also found. The pivot star is labelled as the intersection of $\mathbf{SN}_1$and $\mathbf{SN}_2$. Once the complete triangular feature is calculated a coefficient is assigned to each triangle. The coefficient is calculated as the norm of the cross product of the vectors $\mathbf{SN}_1$ and $\mathbf{SN}_1$ i.e,

$$\text{Coefficient} = \mathbf{SN}_1 \times \mathbf{SN}_2 \tag{3.16}$$

For every star in the position catalog, the neighbouring triangle and the pivot star are detected, and the associated coefficient calculated. Every line of the oriented triangle catalog is composed of pivot star, first neighbouring

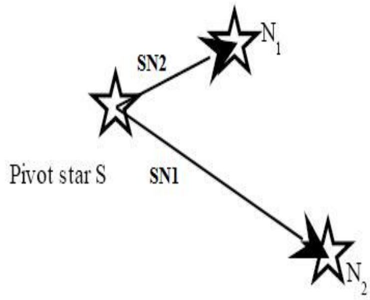


Figure 3.13: Star neighbours, Image [11]

star, second neighbouring star and coefficient. Every triangle is written only once in the catalog. The coefficient assigned to each triangle depends upon the shape of the triad which is a distinct feature for matching the pattern.

The distribution of coefficients is presented in,

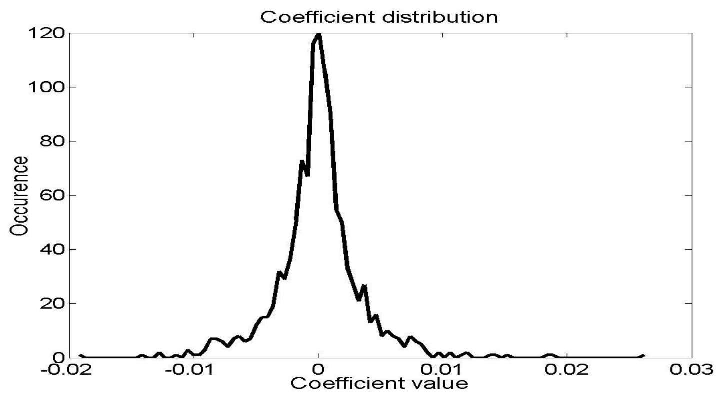


Figure 3.14: Coefficient Distribution, Image [11]

When the stars are detected, oriented triangles are built up and is compared with the coefficients in catalog. Several candidates triangles will appear for a coefficient and a pattern of coherent stars can be deducted if every candidate is cross checked with the three angles in the triangular feature[11].

Chapter 4

Algorithms

One of the main tasks of the thesis work is development of the software algorithm. The software algorithm can be divided into two sections. The first section is responsible for the building up of a catalog and constructing pre-determined triads of all stars in the catalog (Building Catalog). The second section is responsible for the identification of star triads, patterns and angles between them from the images obtained (Image Analysis). The two section are constructed and merged together so that the information gathered in the first section is compared with the second section to get a proper outcome.

4.1 Building catalog

The Hipparcos catalog gives the Right ascension and Declination of the star coordinates with respect to the inertial reference frame. Now the task is to build our own catalog which contains the information about the triads. As an initial step the stars from the base catalog are read which have been already truncated at magnitude 6. The RA and Dec value given in the catalog are converted in to unit vectors and for each star the next two brightest star are found satisfying the condition that both of them lie in between a minimum and maximum angular separation. This ensures that stars lie in field of view. So for each star one triad is constructed. A triad constructed in a oriented

Table 4.1: Look up table in oriented catalog

Star ID	First	Second	Third	1-2	2-3	1-3	Coefficient
1166	11	73	82	87.611	52.741	39.648	0.999130693

Table 4.2: Base catalog

Star ID	RA	Dec	Mag
1166	0.06371	0.35267	4.79000

catalog is shown as example in the table 4.1,

From the above table it is clear that the catalog contains the Star ID then the first, second and third star. They are given temporary numbers based upon the position of stars in catalog which is different from it's unique Star ID. This is done so that in future additional number of stars can be added if the magnitude of stars are truncated at different level. For example, in the above table for the star having a Star ID 1166 in the main catalog is assigned a star number of 11 and the first nearest star is 73 and the second nearest star is 82 which forms the triad.

4.2 Image analysis

This section is responsible for the star detection and triad construction. The algorithm first scans the complete image and reads each pixel value which ranges from 0 to 255. The pixels has a value proportional to the number of photons falling on it which in turn depends upon the brightness. So each pixel of the chip is mapped with a corresponding intensity value in the range of 0 to 255 where '255' represents the maximum intensity. Normally a pixel which has a value greater than the thresholding value is detected as star and the pixel location is transferred to the centroiding algorithm (Section 3.3.1) which exactly detects the centre of the star up to sub pixel accuracy.

After the centroiding process is done the algorithm returns the coordinates of the stars in the image. This coordinates are with respect to the

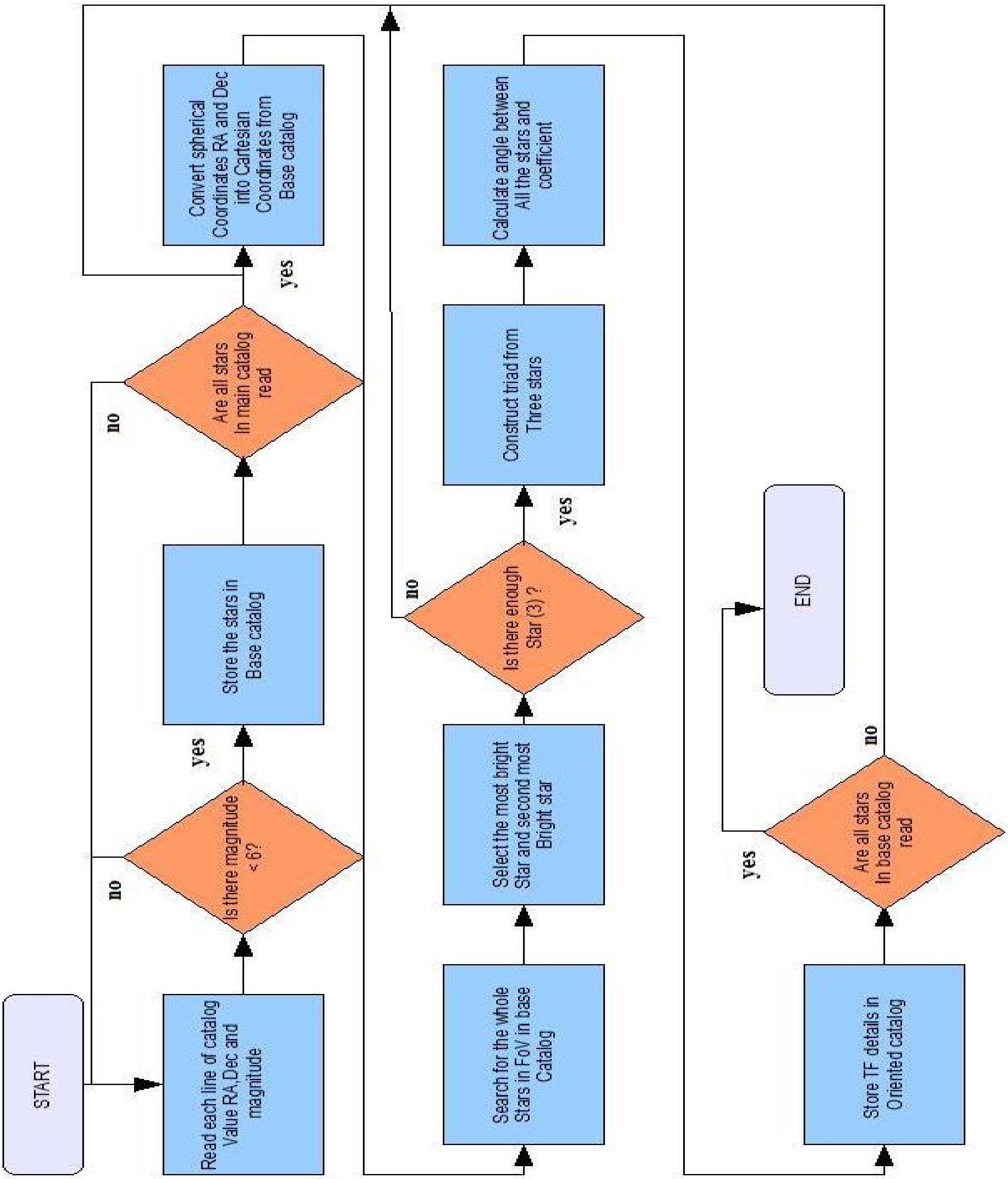


Figure 4.1: Flow chart for Building catalog

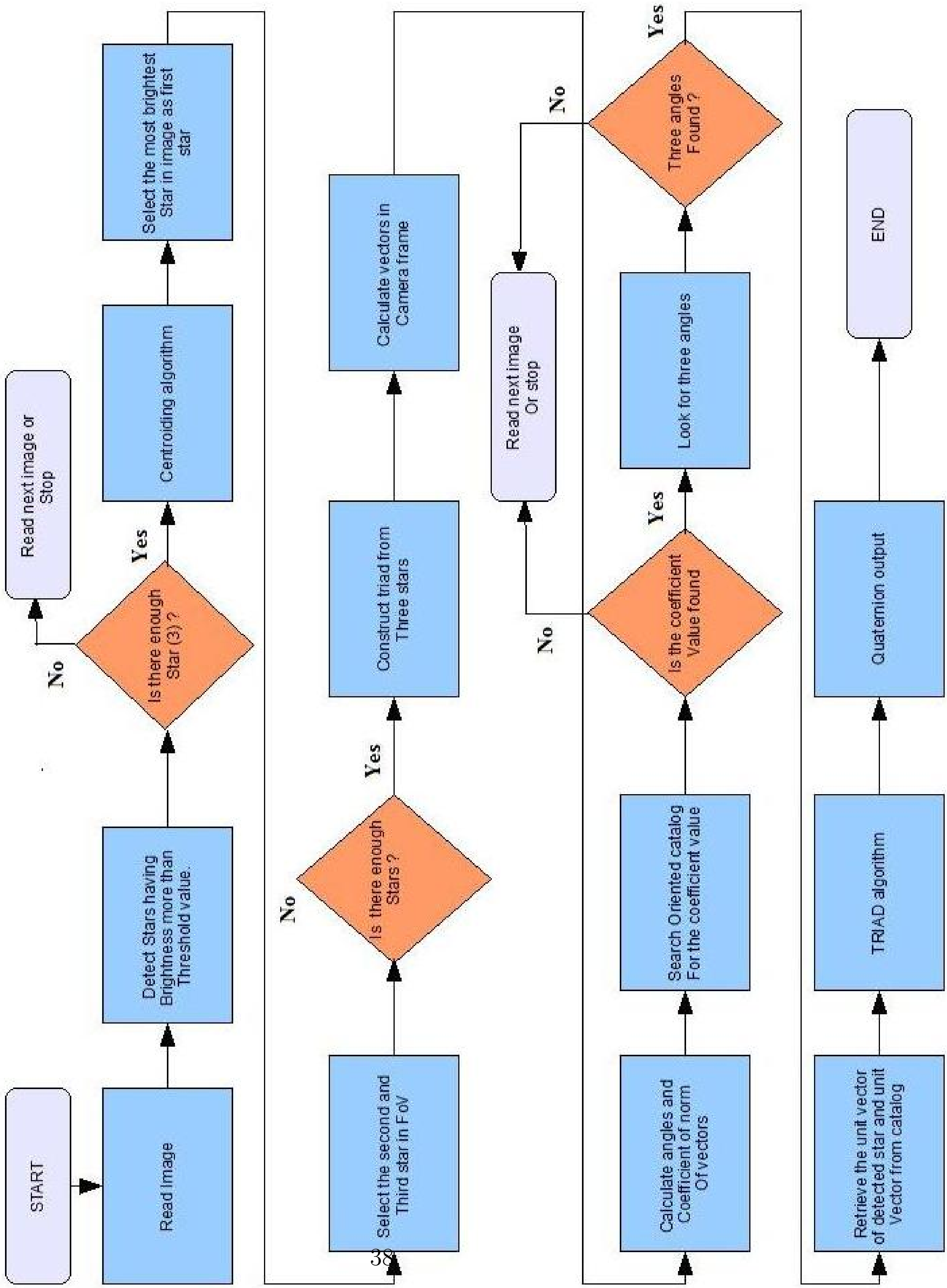


Figure 4.2: Flow chart for Image Analysis

origin at top left corner of the image. This has to be shifted to the coordinates at the frame in centre, since the camera frame is in the middle of the image. But the z-axis coordinate remain the same and is assumed to be unit value. So the coordinates are multiplied with a rotation matrix which shifts the coordinates from the origin to the centre and another shift is also required to represent the coordinates in the real camera frame which is clear from the fig (2.3). The rotation matrix is represented by,

$$[\mathbf{R}] = \begin{bmatrix} 1 & 0 & 0 & \text{-columns/2} \\ 0 & \text{cosd}(180) & -\text{sind}(180) & \text{rows/2} \\ 0 & \text{sind}(180) & \text{cosd}(180) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.1}$$

The coordinates obtained after shifting process represent the real coordinates in the camera frame and now the triad detection process has to be initiated. Since the possible star has been already detected by thresholding now the task lies in finding out the proper star for the triad creation. For triad creation three stars are to be found out which meet the specific criteria. An intuitive approach is used in such a way that the algorithm calculates the star intensity of all the stars in the image and stores it in an array.

Out of this the brightest star is selected and assigned as the pivot star. The two other stars are chosen as the second and third brightest star which lies with the minimum and maximum area that is specified by a two rings.

- Star1- First brightest star.
- Star2- Second brightest star in the ring.
- Star3- Third brightest star in the ring.

The ring gives the minimum and maximum angular distances that are measured in radians. The minimum angle separation is done so that the bright star which is spread across many nearby pixels is not misinterpreted as a two different stars but as a single star. The maximum angular distance ensures that the brightest star is picked up only from the field of view.

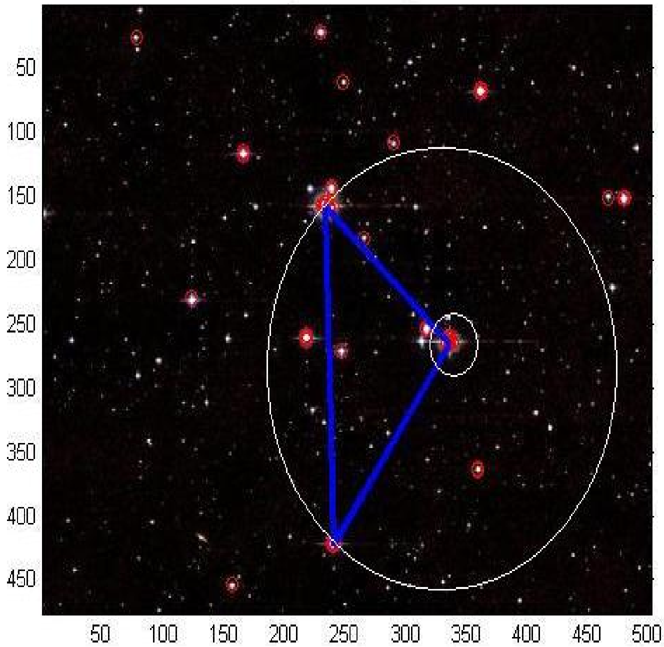


Figure 4.3: Minimum and Maximum angular distance

The vectors of each star are constructed by taking x-coordinates and y-coordinates from the pixel location and the z- coordinate is assumed to be unit distance. The three star vectors are then normalized. The normalization of the vectors doesn't affect the angle between them or the shape of the triad which are the important factors for the detection.

Chapter 5

Attitude determination

The attitude of a three-dimensional body is most conveniently defined with a set of axis fixed to the body. This set of axes is generally a triad of orthogonal coordinates; and is normally called a body coordinate fame. The attitude of a body is thought of a coordinate transformation that transforms a defined set of reference coordinates into the body coordinates of the spacecraft. So the task of the attitude determination algorithms is to find out a an appropriate transformation[5].

In the previous section we have seen how the stars are been identified and detected. After a successful match between the stars in the image and the stars in catalog is occurred, we will be having a set of observed position in camera frame and a set of observed position in the inertial frame. A three dimensional rotation has three degrees of freedom. Each observation has two degrees of freedom, therefore if only one star is identified, the problem is underdetermined and with two or more stars it is over determined, and noise ensures that there will be no exact solution. In practice it is only possible to get an initial fix with three or more stars.

The attitude determination problem can be defined as finding out the rotation matrix or quaternion (described in section 2.5) which transforms the coordinates in the inertial frame to the camera frame namely, $\mathbf{A}= \mathbf{R}_{\mathrm{B}}{}^{\mathrm{I}}$ where B and I represent body frame and inertial frame respectively. Thus obtaining the rotation matrix between camera frame and inertial frame helps in finding out the rotation matrix between the body frame of satellite and the inertial frame. The rotation matrix between the camera frame and the satellite body frame is fixed since the camera is in fixed position w.r.t satellite body. There are several methods by which the attitude determination can be done from two vector observation namely QUEST algorithm [12] or TRIAD algorithm or Wahba problem [13]. Both of these methods are briefly discussed in following sections and appropriate algorithm is selected based on the number of vectors observed and the sensor on which the algorithm is applied

5.1 Vector from Star Image

The star sensor measures the coordinates of the star-direction unit vector in the sensors three dimensional orthogonal frame [5]. The star vector components in the sensor's axis frame can be derived from the image. Let us assume that the sensor axis frame X_s, Y_s, Z_s is as shown in the figure (5.1) and the star vector is to be expressed in the same frame. The vector can be measured in the terms of the angles ϕ and λ, where λ is the elevation of the star image above the X_s- Y_s plane ϕ is the angle between (a) the projection of the line of sight of the star on the X_s- Y_s and (b) the Y_s axis. The components of star sensor vector in the sensor frame is given by [4],

$$\mathbf{S} = \begin{bmatrix} -\sin(\phi)\cos(\lambda) \\ \cos(\phi)\cos(\lambda) \\ -\sin(\lambda) \end{bmatrix} \tag{5.1}$$

From the figure(5.1) it can be easily observable that,

$$\tan(\phi) = u/f \tag{5.2}$$

$$\tan(\lambda) = (v/f)\cos(\theta)\cos(\phi) \tag{5.3}$$

where u and v are the two coordinates of the star image on the focal plane (i.e. the image on the CCD plane) and f is the focal length. Applying this equations the unit vector for the whole stars in the image can be found out.

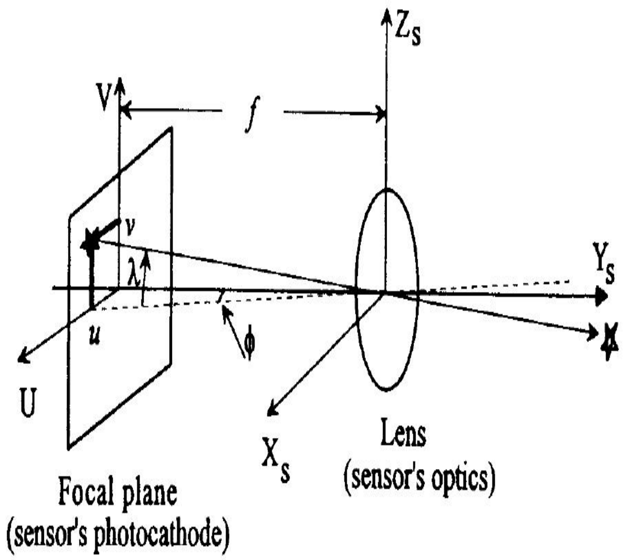


Figure 5.1: Star unit vector from camera frame , Image [5]

5.2 Triad Algorithm

The triad algorithm which determines the attitude based on vector observation was defined by Chang and Shuster [14]. Given two nonparallel reference unit vectors $\mathbf{V}_1$ and $\mathbf{V}_2$ and the corresponding observation unit vectors $\mathbf{W}_1$ and $\mathbf{W}_2$ we wish to find an orthogonal matrix $\mathbf{A}$ which satisfies the condition given in Eq 5.4 and 5.5,

$$\mathbf{A}\mathbf{V}_1 = \mathbf{W}_1 \tag{5.4}$$

$$\mathbf{A}\mathbf{V}_2 = \mathbf{W}_2 \tag{5.5}$$

where $\mathbf{V}_1$ and $\mathbf{V}_2$ are vectors in inertial reference frame and $\mathbf{W}_1$ and $\mathbf{W}_2$ are vectors in camera frame.

Because the matrix $\mathbf{A}$ is over determined by the above equations, we begin by constructing two triads of manifestly orthonormal reference and observation vectors,

$$\mathbf{r}_1 = \mathbf{V}_1 \qquad \mathbf{r}_2 = \frac{\mathbf{V}_1 \times \mathbf{V}_2}{|\mathbf{V}_1 \times \mathbf{V}_2|} \tag{5.6}$$

$$\mathbf{r}_3 = \mathbf{V}_1 \times \frac{(\mathbf{V}_1 \times \mathbf{V}_2)}{|\mathbf{V}_1 \times \mathbf{V}_2|} \tag{5.7}$$

$$\mathbf{s}_1 = \mathbf{W}_1 \mathbf{s}_2 = \frac{\mathbf{W}_1 \times \mathbf{W}_2}{|\mathbf{W}_1 \times \mathbf{W}_2|} \tag{5.8}$$

$$\mathbf{s}_1 = \mathbf{W}_1 \times \frac{(\mathbf{W}_1 \times \mathbf{W}_2)}{|\mathbf{W}_1 \times \mathbf{W}_2|} \tag{5.9}$$

There exists a unique orthogonal matrix A which satisfies,

$$\mathbf{A}\mathbf{r}_1 = \mathbf{s}_1 \quad i = (1, 2, 3) \tag{5.10}$$

which is given by,

$$\mathbf{A} = \sum_{i=1}^{3} \mathbf{s}_1 \mathbf{r}_{\mathrm{i}}{}^{\mathrm{T}} \tag{5.11}$$

where T denotes the matrix transpose (is interpreted as a 3 x 1 matrix and as a 1 x 3 matrix). In other notation Eq. (5.12) is identical to,

$$\mathbf{A} = \mathbf{M}_{\mathrm{obs}} \mathbf{M}_{\mathrm{ref}}{}^{\mathrm{T}} \tag{5.12}$$

with,

$$\mathbf{M}_{\mathrm{obs}} = [\mathbf{r}_1 \vdots \mathbf{r}_2 \vdots \mathbf{r}_3] \quad \mathbf{M}_{\mathrm{ref}} = [\mathbf{s}_1 \vdots \mathbf{s}_2 \vdots \mathbf{s}_3] \tag{5.13}$$

The right members of Eqs. (5.14) are 3x3 matrices labelled according to their column vectors. Equation (5.12) or equivalently Eq.(5.13) defines the TRIAD solution. A necessary and sufficient condition that the attitude matrix given by Eq. (5.12) also satisfy Eqs. (5.4 and 5.5) is,

$$\mathbf{V}_1 \cdot \mathbf{V}_2 = \mathbf{W}_1 \cdot \mathbf{W}_2 \tag{5.14}$$

The TRIAD solution is not symmetric in indices 1 and 2. Clearly, because part of the information contained in the second vector is discarded. The TRIAD solution will be more accurate when ($\mathbf{W}_{\mathrm{I}}$,$\mathbf{V}_{\mathrm{I}}$) is chosen to be the observation reference vector pair of greater accuracy.

5.3 Quest ALGORTIHM

In triad algorithm the aim was to find a matrix **A** which transfers the coordinate from one frame to another frame, but in QUEST algorithm it is to find an orthogonal matrix $\mathbf{A}_{\mathrm{opt}}$ that minimizes the loss function[14]. When the loss function (eq 5.17) is at its minimum the matrix **A** satisfies both the equation 5.15 and 5.16,

$$\mathbf{A}\mathbf{V}_1 = \mathbf{W}_1 \tag{5.15}$$

$$\mathbf{A}\mathbf{V}_2 = \mathbf{W}_2 \tag{5.16}$$

The loss function is given by,

$$\mathrm{L(A)} = \frac{1}{2}\sum_{i=1}^{n} a_i|\hat{W}_i - A\hat{V}_i|^2 \tag{5.17}$$

where,

$$a_i = 1, 2 \ldots n$$

are a set of nonnegative weights which can be scaled depending on the observations. The loss function may be scaled without affecting the determination of $\mathbf{A}_{\mathrm{opt}}$, it is possible to set,

$$\sum_{i=1}^{n} \mathrm{a_i} = 1 \tag{5.18}$$

The gain function g(A) is defined by,

$$g(A) = 1 - L(A) = \sum_{i=1}^{n} a_i \mathbf{W}_i{}^T \mathbf{A} \mathbf{V}_i \tag{5.19}$$

The loss function L(A) will be at a minimum when the gain function g(A) is at a maximum. So when the matrix A_opt is found out the gain function will be at maximum. Interpreting the individual terms of Eq. (5.20) as 1 x 1 matrices, it follows from a well-known theorem on the trace that,

$$g(A) = \sum_{i=1}^{n} a_i \mathrm{tr}[\mathbf{W}_i{}^T \mathbf{A} \mathbf{V}_i] = tr[\mathbf{A}\mathbf{B}^T] \tag{5.20}$$

where tr denotes the trace operation and **B**, the attitude profile matrix, is given by,

$$\mathbf{B} = \sum_{i=1}^{n} a_i \mathbf{V}_i{}^T \tag{5.21}$$

The maximization of g(A) is complicated by the fact that the nine elements of **A** are subject to six constraints. Therefore, it is convenient to express **A** in terms of its related quaternion.

The quaternion **q** representing a rotation is given by,

$$\mathbf{q} = \begin{Bmatrix} \mathbf{Q} \\ \mathrm{q} \end{Bmatrix} = \begin{Bmatrix} \mathbf{X}\sin(\theta/2) \\ \cos(\theta/2) \end{Bmatrix} \tag{5.22}$$

where **X** is the axis of rotation and θ is the angle of rotation about X. The quaternion satisfies a single constraint, which is,

$$\mathbf{q}^T\mathbf{q} = |\mathbf{Q}|^2 + q^2 = 1 \tag{5.23}$$

The attitude matrix **A** is related to the quaternion by,

$$\mathbf{A}(\mathrm{q}) = (\mathrm{q}^2 - \mathbf{Q} \cdot \mathbf{Q})\mathbf{I} + 2\mathbf{Q}\mathbf{Q}^{\mathrm{T}} + 2\mathrm{q}\mathbf{Q} \tag{5.24}$$

where **I** is the identity matrix and **Q** is the antisymmetric matrix given by,

$$\mathbf{Q} = \begin{bmatrix} 0 & \mathbf{Q}_3 & -\mathbf{Q}_2 \\ -\mathbf{Q}_3 & 0 & \mathbf{Q}_1 \\ \mathbf{Q}_2 & -\mathbf{Q}_1 & 0 \end{bmatrix} \tag{5.25}$$

Substituting Eq.(5.25) into Eq.(5.21) the gain function may be rewritten as,

$$\mathrm{g}(\mathrm{q}) = (\mathrm{q}^2 - Q \cdot Q) tr \mathrm{B}^{\mathrm{T}} + 2tr[\mathrm{Q}\mathrm{Q}^{\mathrm{T}}\mathrm{B}^{\mathrm{T}}] + 2qtr[\bar{Q}B^{T}] \tag{5.26}$$

Introducing the quantities,

$$\sigma = tr\mathbf{B} = \sum_{i=1}^{n} a_i \mathbf{W}_{\mathrm{i}} \mathbf{V}_{\mathrm{i}} \tag{5.27}$$

$$\mathbf{S} = \mathbf{B} + \mathbf{B}^{\mathrm{T}} = \sum_{i=1}^{n} \mathrm{a}_{\mathrm{i}} (\mathbf{W}_{\mathrm{i}} {\mathbf{V}_{\mathrm{i}}}^{\mathrm{T}} + (\mathbf{V}_{\mathrm{i}} {\mathbf{W}_{\mathrm{i}}}^{\mathrm{T}}) \tag{5.28}$$

$$\mathbf{Z} = \sum_{i=1}^{n} \mathrm{a}_{\mathrm{i}} (\mathbf{W}_{\mathrm{i}} \times \mathbf{V}_{\mathrm{i}}) \tag{5.29}$$

leads to the bilinear form,

$$\mathrm{g(q)} = \mathrm{q^T}\mathbf{K}\mathrm{q} \tag{5.30}$$

where the 4 x 4 matrix **K** is given by,

$$\mathbf{Q} = \begin{bmatrix} \mathbf{S} - \sigma\mathbf{I} & \mathbf{Z} \\ \mathbf{Z}^{\mathrm{T}} & \sigma \end{bmatrix} \tag{5.31}$$

Equation (5.30) may be written alternatively as,

$$\mathbf{Z} = \mathbf{B} - \mathbf{B}^{\mathrm{T}} \tag{5.32}$$

The problem of determining the optimal attitude has been reduced to finding the quaternion that maximizes the bilinear form of Eq. (5.31). The constraint of Eq. (5.24) can be taken into account by the method of Lagrange multipliers. A new gain function g' (q) is defined as,

$$\mathrm{g(q)} = \mathrm{q^T}\mathbf{K}\mathrm{q} - \lambda \mathrm{g^T q} \tag{5.33}$$

which is maximized without constraint. X is then chosen to satisfy the constraint. It may be verified by straightforward differentiation that g' (q) attains a stationary value provided,

$$\mathbf{Kq} = \lambda\mathbf{q} \tag{5.34}$$

Thus $\mathbf{q_{opt}}$ must be an eigenvector of **K**. Equation (5.35) is independent of the normalization of **q** and therefore Eq. (5.24) does not determine X. However X must be an eigen value of **K** and for each eigenvector of **K**,

$$g(q) = q^{T}\mathbf{K}q = \lambda q^{T}q = \lambda \tag{5.35}$$

Thus g(q) will be maximized if $\mathbf{q_{opt}}$ is chosen to be the eigenvector of $\mathbf{K}$ belonging to the largest eigenvalue of $\mathbf{K}$. More concisely,

$$\mathbf{K}\mathbf{q}_{\mathrm{opt}} = \lambda_{\mathrm{max}}\mathbf{q}_{\mathrm{opt}} \tag{5.36}$$

which is the desired result.

5.4 Comparison of TRIAD and Quest Algorithm

Before implementing the algorithm a proper comparison is necessary to find out the suitable one for specific mission and the accuracy needed. A simple connection between TRIAD and Quest algorithm can be found out by examining the loss function given in Eq (5.21). The loss function reduces to the following for a case where two vectors are observed,

$$L(A) = \frac{1}{2}a_1|\mathbf{W}_1 - \mathbf{A}\mathbf{V}_1|^2 + \frac{1}{2}a_2|\mathbf{W}_2 - \mathbf{A}\mathbf{V}_2|^2 \tag{5.37}$$

It will now be proved that the orthogonal matrix that minimizes L (A) passes in the limit a_2/a_1 tends to 0 into the TRIAD attitude matrix. It may be noted first that as a_2/a_1 becomes increasingly smaller the constraint,

$$\mathbf{A}\mathbf{V}_1 = \mathbf{W}_1 \tag{5.38}$$

is enforced with increasingly greater severity. Therefore it is sufficient to show that the attitude matrix that minimizes the loss function of Eq. (5.38) subject to the constraint of Eq. (5.39) is the TRIAD attitude matrix. In terms of the TRIAD vectors of Sec(5.1) , the constraint becomes,

$$\mathbf{Ar}_1 = \mathbf{s}_1 \tag{5.39}$$

and the constrained loss function takes the form,

$$L(A) = \frac{1}{2}a_2|\mathbf{W}_1 \cdot \mathbf{W}_2 - \mathbf{V}_1 \cdot \mathbf{V}_2|\mathbf{s}_1 - |\mathbf{W}_1 \times \mathbf{W}_2|\mathbf{s}_3 + |\mathbf{V}_1 \times \mathbf{V}_2||\mathbf{Ar}_3|^2 \tag{5.40}$$

Because $\mathbf{A}$ is orthogonal, $\mathbf{Ar}_3$ can have no component along $\mathrm{S_j}$. Therefore, the choice,

$$\mathbf{Ar}_3 = \mathbf{s}_3 \tag{5.41}$$

minimizes the loss function subject to the constraint. Because A must also be unimodular, it follows that,

$$\mathbf{Ar}_2 = \mathbf{s}_2 \tag{5.42}$$

Hence, $\mathbf{A}$ is the TRIAD attitude matrix

Two algorithms TRIAD and QUEST has been discussed which determines the three-axis attitude from vector observations. Both of these algorithms are simple and powerful with high computational efficiency and accuracy. Most of the attitude determination software requires extensive trigonometric calculations which uses high computation resources and power. The TRIAD and QUEST algorithms uses only simple matrix operations to determine attitude. In attitude determination systems which require fast computations, TRAID and QUEST is preferred especially in ground support systems where attitude is frequently calculated.

The QUEST algorithm requires more computational resources, since it requires more statement in the code for realising the same task. The accuracy of the QUEST algorithm is very high when compared to the TRAID algorithm. So a comparison between these two algorithms must be done before selecting it for a specific mission. In this case QUEST is the best choice considering accuracy but computational efficiency is be very high. For low accuracy missions the opposite is usually the case and the TRIAD algorithm is the best choice. When more than three observations must be employed in a single frame the QUEST algorithm becomes computationally more efficient as well. A possible drawback is the method of sequential rotations which potentially could increase the number of computations by a factor of 4

Even though QUEST algorithm was the best choice for in terms of high accuracy, in this thesis work TRIAD algorithm is selected so as to reduce the computations, but deliver a reasonable accuracy. Once the stars were detected from the oriented catalog two vectors has to be input to the triad algorithm so as to produce an attitude solution. Out of the three stars detected in an image two stars are selected to form pairs and hence three star pairs are formed. This three pairs were input to the same algorithm so that it produces different solution out of which most accurate one is selected.

Chapter 6

Results

The algorithm developed for the thesis was tested for more than twenty different images and were proven to be successful. The images are from various camera and has different field of view, magnitude and focal length. So for each test cases the settings was changed in the algorithm manually. For example, an image from the Gemini constellation where the brightest stars (pollux and castor) are visible is shown and this camera image has a field of view 13 degree. For each test cases the catalog is also newly constructed since the field of view changes for each images.

6.1 Triad detection results

The star triads were detected and the triangular features like angles and coefficient were calculated from the image. The star image shown below is the castor and pollux from the Gemini constellation. The star detected on the right end is the castor and the star in the middle is pollux which is confirmed from the constellation picture available [16]. Further confirmation is done when the catalog number is retrieved during search process which is 37718 in this case. The catalog values and the values calculated are shown in separate tables. An additional margin is given for angles and coefficient so that error in calculation of triangular feature doesn't effect the search process.

Table 6.1: Oriented Catalog values

Star ID	First	Second	Third	1-2	2-3	1-3	Coefficient
37718	529	510	527	146.266	15.005	18.729	0.555334882

Table 6.2: Calculated values from Image

Star ID	1-2	2-3	1-3	Coefficient
37718	146.5395	14.8350	18.6255	0.5514

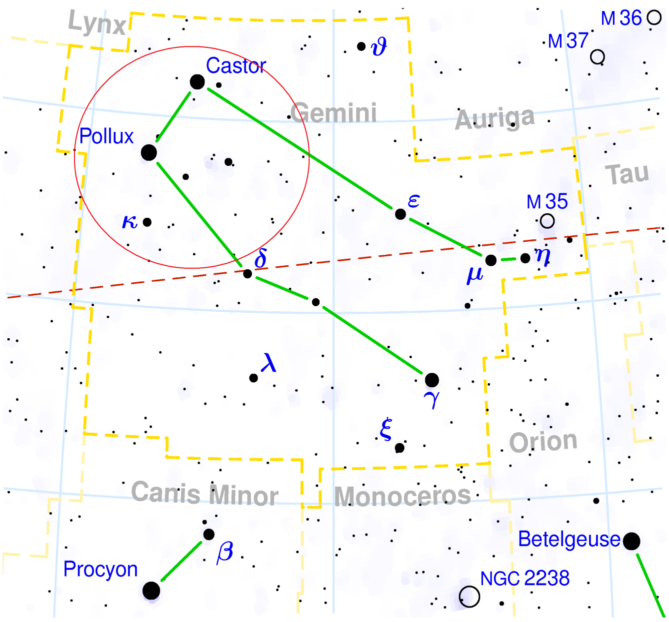


Figure 6.1: Gemini Constellation, Red round indicates the FoV camera points.Image[16]

In the figure (6.1) the red circle shows the area where the camera is pointed and the figure (6.2) shows the stars detected in the same constellation.

6.2 Attitude determination results

The Attitude determination results were obtained from the TRAID algorithm. From one image three stars are detected and any two stars can form a pair which can be passed to the TRIAD algorithm. So below shown are the

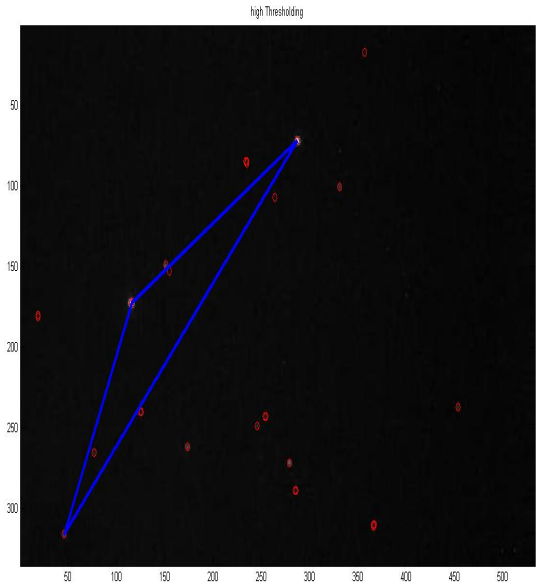


Figure 6.2: Detected triad in a star image

Table 6.3: Results for quaternions

quaternions1	0.9485	-0.1270	0.2167	-0.1929
quaternions2	0.9484	-0.1268	0.2168	-0.1933
quaternions3	0.9109	0.0017	0.4030	-0.0888

three results for three such pair i.e. first and second star, second and third star and first and third star. For all the three star pairs both the unit vector in the sensor frame and unit vector in the camera frame were calculated and used in TRIAD algorithm. The result of the TRIAD algorithm also depends upon the order in which the stars are passed. For the first two pairs the quaternion are nearly equal and doesn't vary much, but for the third pair it varies since the order of the star vector passed to algorithm were changed.

Chapter 7

Conclusion

The objective of this thesis work was to build MATLAB software which would be able to detect star pattern from images and produce an attitude solution. As an initial step several simulations were conducted to find out the appropriate field of view of camera and maximum magnitude of the stars that should be detected. The software doesn't detect all the possible triads in image instead it detects a single triad in the image and searches for it in catalog. This method has got an advantage that it requires only very few memory space since the number of triads stored is less and hence the search time is also less. A considerable amount of reduction in processing time also has been noticed and will be further reduced when performed in embedded code in processors. Since all the possible triads are not detected in one direction additional gyroscopes or some other devices which can calculate attitude change must be employed in the mission with this kind of star sensor. The most challenging part was the star detection algorithm since there was no real camera built. Moreover the images were from different camera which had varying intensities, focal length and pixels. So each time the magnitude settings and focal length had to be changed. But the above results were shown for a standard image with focal length 15-20mm and 640 * 480 pixel dimensions and field of view 13 degree.

Chapter 8

Possible future work

Since the software is a initial prototype there are lots of possibility for future work

- The development of a C/embedded code for the same algorithm to be implemented in the processors/controllers. There are modern tools which can help to generate C code from Matlab. The Embedded MATLAB Function block provides a subset of the MATLAB language that can be used to generate embedded C code using Real-Time Workshop. But usually the code generated in these cases are not optimized and are not properly commented. So a manual development of C code would be the best option but by applying the same logic defined in the Matlab algorithm ,which makes the work easier.

- The triad algorithm is simple and requires less processing power, but the accuracy is less when compared to the QUEST algorithm. So implementation of QUEST algorithm in the attitude determination part can also be another future work.

Appendix A

Software Description

This section contains the description of the whole software files that are used in the development of the thesis work. The code is separately attached in the Compact Disc (CD) due to the very high size. Each section only gives a short description and it is advised to read the commented code to understand the full functioning of the software modules. All the files are in MATLAB programming

A.1 Software modules

A.1.1 main.m

This is the main software module where the image is acquired an processed. This module is responsible for selecting stars, centroiding, calculating vectors in the camera frame. The main module also calls several other functions during its operation. The sub modules called are centroiding.m search.m, and triad.m which are briefly explained in the following sections.

A.1.2 centroiding.m

This is sub module where the stars coordinate are centroided to sub pixel accuracy. Theoretical explanation is given in the section 3.3

A.1.3 search.m

Is used to find out the candidate stars. This section searches the oriented catalog and finds out the stars which have the same triangular feature as the star detected in the image.

A.1.4 triad.m

This module is used to implement the triad algorithm as defined in the section (5.1). The triad algorithm takes input as star vectors from the catalog and from the image and produces an unique attitude solution in terms of quaternion. The output can also be converted into rotation matrix which is commented for time being and can be changed according to future user.

A.1.5 build_oriented_catalog.m

This software module is used to build the oriented catalog. The process is as per described in the section 3.5.3. The input from the base catalog is truncated at magnitude of 6 and the output is stored in oriented catalog in the text (.txt) format

A.1.6 Monte Carlo_analysis.m

This simulation program is used to do a Monte Carlo analysis and find out the appropriate field of view.

A.1.7 celestial_sphere.m

Is used to simulate the whole celestial sphere and to study about the distribution of stars in the sky.

A.2 Files

A.2.1 Hipparcos full.csv

Is the excel file downloaded from internet [10].It contains all data mentioned in the section 3.5.1. This file is truncated from a magnitude 6 for the creation of base catalog and only relevant data are stored in basecatalog.txt

A.2.2 Hipparcoscatalog.txt

Is the text file created from the hipparcos catalog avoiding all the redundant data like velocity, parsec etc so as to increase speed while creation of the base catalog file. Hence it is not mentioned in the main sections.

A.2.3 base catalog.txt

Is the text file in which the star details such as StarID, RA, Dec and Magnitude are stored. The magnitude is truncated at 6

A.2.4 oriented catalog.txt

Is the text file in which the triads and its triangular features are stored.

Bibliography

[1] M K Samaan, *Toward faster and more accurate star sensors using recursive centroiding and star identification*. Phd Thesis, University of Texas, Texas, 2003.

[2] $http//microsat.sm.bmstu.ru/e-library/Components/StarCameras/Star$ $/20Sensor/20KM/201301_files/img0242.jpeg$

[3] M. D. Shuster and S.D. Oht *Three-Axis Attitude Determination from Vector Observations*. J.Guidance and Control vol.4, no.1 AIAA, Computer Sciences Corporation, Silver Spring, 1981.

[4] J. R. Wertz and W. J. Larson, *Space mission analysis and design*. Third Ed. Microcosm, Silver Spring, 1999.

[5] Marcel J.SIDI, *Spacecraft Dynamics and Control*. Cambridge University Press, Cambridge, 1997.

[6] *http//en.wikipedia.org/wiki/Monte_Carlo_method*.

[7] Kenneth Daniel Diaz, *Performance analysis of a fixed point star tracker algorithm for use onboard of a pico satellite*. Master thesis, California Polytechnic State University, San Luis Obispo, August 2006.

[8] Carl Christian Liebe, *Pattern Recognition of Star Constellations for Spacecraft,* ApplicationsIEEE AES Systems Magazine, January 1993

[9] *http//en.wikipedia.org/wiki/Hipparcos_Catalogue*

[10] *http//astronexus.com/node/34*

[11] B. M. G. Lamy au Rousseau, J. Bostel, *New star pattern recognition algorithm for aps star tracker application: Oriented triangles,* IEEE Aerospace Electron. Syst.Mag., pp.27-31, 2005.

[12] M. L. Psiaki. *Extended Quest Attitude Determination Filtering* 2009, May

[13] Malcolm D. Shuster, *The Generalized Wahba Problem* The Journal of the Astronautical Sciences, Vol. 54, No. 2, April-June 2006, pp. 245-259

[14] Chang, Y., M. Shuster, *QUEST and The Anti-QUEST: Good and Evil Attitude Determination Determination,* J.Astronautical Sciences, Vol. 53 No. 3, July-September 2005.

[15] M.Hlond and M.Bzowski, *Star Sensor device simulation program* Science Working Team Meeting,Space Research Centre, Bern, April 2007

[16] *http//en.wikipedia.org/wiki/Gemini constellation*

[17] Ray Zenick,Thomas J. McGuire, *Lightweight, Low-Power Coarse Star Tracker,* 17th Annual AIAA/USU Conference on Small Satellites,Utah.

[18] Henry Hallam *Low-Cost Star Tracker for Spacecraft Attitude Determination,* Master Thesis,Pembroke College, Cambridge,March 2009