INSTITUTE OF CONTROL AND COMPUTATION ENGINEERING
FACULTY OF ELECTRONICS AND INFORMATION TECHNOLOGY
WARSAW UNIVERSITY OF TECHNOLOGY

MASTER OF SCIENCE THESIS

STAR-TRACKER PROGRAM FOR CUBESAT SATELLITES

Szymon MICHALSKI

Supervisor:
prof. dr hab. inż. Ryszard Romaniuk

Warszawa 2016

Abstract

Last years directed space industry towards small satellites. Many countries which did not have possibility to enter this branch of industry before now create their own solutions. The goal of this work is to create fully functioning star-tracker software eligible to be used in future satellites as Polish sulition of determination of satellite attitude towards Earth. Work contains also description of individual parts and variants of solutions connected with star-tracker.

Streszczenie

Ostatnie lata ukierunkowały przemysł kosmiczny na małe satelity. Wiele krajów, które wcześniej nie miały możliwości wejścia w tą gałąź przemysłu, teraz tworzą własne rozwiązania. Celem niniejszej pracy dyplomowej jest stworzenie w pełni działającego oprogramowania star-trackera nadającego się do wykorzystania w przyszłych satelitach jako polskie rozwiązanie problemu określania orientacji satelity względem ziemi. Praca zawiera też opis poszczególnych części i wariantów rozwiazania problemów związanych ze star-trackerem.

Contents

Acronyms

ADCS Attitude Control and Determination System.

GPS Global Positioning System.

LEO Low Earth Orbit.

LIS Lost-In-Space.

List of Symbols

ϕ Euler angle, roll.

θ Euler angle, pitch.

ψ Euler angle, yaw.

$\boldsymbol{R}(\cdot)$ Rotation matrix using Euler angles.

$\boldsymbol{q}$ Unit quaternion.

q_0 Scalar part of unit quaternion.

$\boldsymbol{q}_{vec}$ Vector part of unit quaternion.

$\boldsymbol{Q}$ Quaternion matrix.

v General Euler angle.

$\boldsymbol{n}$ Unit vector.

$\boldsymbol{I}$ Identity matrix.

$\boldsymbol{S}(\cdot)$ Skew symmetric matrix.

$\boldsymbol{R}_n^b$ Rotation matrix representing a rotation from n to b.

$\boldsymbol{M}$ Least squares estimate of rotation matrix.

$\boldsymbol{r}$ Known directional unit vector in the NED frame.

$\boldsymbol{b}$ Known directional unit vector in the BODY frame.

Jenssen et al. [1] Valenti et al. [2] Delabie [3] Jalabert et al. [4] Felikson et al. [5] Knutson [6] Rose [7] Mortari and Romoli [8]

1 Introduction

Stars were used for navigation already ages ago. Together with technological development, spread of sailing and seafaring people started to use more advanced tools helping to more precisely estimate position of ships on sea. In XVIII century new tool was created, names sextant. This device helped to quite precisely calculate angle between the line ship-horizon and ship-star, what let to estimate position.

Nowadays, hundreds of years later, thanks to other technologies like for example Global Positioning System (GPS) stars are not necessary anymore for traveling. As it is commonly known, GPS technology is based on satellites, and today those satellites need stars for determinating their attitude towards Earth.

With the miniaturization of electronics and battery new type of satellites was invented: CubeSats microsatellite. They have now greater sensoric and processing power possibilities, previously found only in larger satellites. However CubSats are still behind in terms of attitude determination.

1.1 Motivation

The goal of this work is to make fully operational star-tracker software, that could be used on Cubesat satellites. Such program could be used on space missions and could start Polish state-of-the-art technology in growing space technology sector. There is already existing prototype of on-board computer for such CubeSat satellite, which together with this work should create full star-tracker device: hardware + software.

1.2 Outline of thesis

This thesis consists of several chapters. Here they are shortly summarized:

Chapter 1 serves as introduction to this thesis and describest the motivation and goal of this work. It also describes the background of the topic.

Chapter 2 describes all the important foundations for the fully understanding given work.

Chapter 3 is the main part of this thesis. It describes how the star-tracker program works and goes through detailed comparison of different approaches.

Chapter 4 describes the created prototype of star-tracker in Python language.

Chapter 5 talks about the implementation of star-tracker on the existing prototype of on-board computer.

Chapter 6 describes how the finished program is performing.

Chapter 7 contains conclusons about this work and created star-tracker program.

1.3 Cubesat

In recent years there has been the development of microsatellites called CubeSats. CubeSat is a standard created in 1999 at the California Polytechnic State UniversityHeidt et al. [9], which is used for low-cost microsatellites. CubeSats are measured in units. Most are 1-U, 2-U and 3-U. CubeSat 1-U has a size of 10 cm on each edge and the maximum weight of 1.333 kg, while the CubeSat 2-U is 20x10x10 cm and can weigh up to 2.666 kg.

The advantage of the standard is primarily reduction of the price of elevation of such satellites into orbit. Their popularity is evidenced by the fact that the percentage of satellites weighing less than 10 kg has increased to about 60% of all satellites in orbit, and only CubeSats make up about half of the small satellites that are fired every year.

TODO Zdjecie Cubesata CubeSat missions, goals, examples, what can they be used for?

1.4 Means of attitude determination

Attitude of spaceships must be usually stabilised and controlled for various reasons. It is necessary for satellite antenna to be pointing towards Earth for the proper communication, to intelligently control the heat by using the effects of cooling and heating of the shadows and sunlight, as well as to navigate: maneuvers must be performed in the right directions.

Attitude is determined between two coordinate systems (where one is reference system) and defines by what angles the coordinate system connected with the researched object has to be shifted in order to cover the reference system. Devices such as planes and satellites have so called Attitude Determination and Control System (ADCS), which controls attitude of object relative to an intertial reference frame or another entity (the celestial shere, certain areas, the nearby objects, etc.).

Currently, attitude determination of CubeSats is limited mainly to the sun sensors, magnetometers and measurements of inertia. The following table describes the accuracy of different sensors. Unquestionably winner here is the startracker, which, due to its quality and uniqueness of the constellations is ideal for navigation.

1.4.1 Inertial Meaurement Unit

1.4.2 Sun Sensors

1.4.3 Star Sensors

1.4.4 Horizon Sensors

1.4.5 Magnetometer

1.4.6 GPS - maybe

1.4.7 Conclusons

Sun sensors can provide very accurate measurements, but can only operate in sunlight. For low-Earth orbit (LEO), even 30% of the orbit can be done in the darkness. Magnetometers are small and can give accurate measurements of the proper calibration. Their drawback is the limited knowledge of the magnetic field and electromagnetic interference due to highly integrated construction of CubeSats. Microelectromechanical gyroscopes are small enough to fit into a CubeSats. However, they suffer from sudden movements, and could not maintain the correct measurement of the 15-minute period of the eclipse orbit LEO. To be truly competitive and reliable platform, CubeSats must provide the correct determination of attitude. The best way to meet this goal is by using star-trackers.

Larson and Wertz [10] Lima [11]

1.5 On-board computer

This section will describe the on-board computer which was done as part of other thesis.

Sensor	Typical Performance Range
Inertial Measurement Unit (Gyros and Accelerometers)	Gyro drift rate = 0.003 deg/hr to 1 deg/hr, accel. Linearity = 1 to 5 x $10^{-6} g/g^2$ over range of 20 to 60 g
Sun Sensors	Accuracy = 0.005 deg to 3 deg
Star Sensors (Scanners and Mappers)	Attitude accuraty = 1 arc sec to 1 arc min 0.0003 deg to 0.01 deg
Horizon Sensors -Scanner/Pipper -Fixed Head (Static)	Attitude accuracuracy: 0.1 deg to 1 deg (LEO) $<$ 0.1 deg to 0.25 deg
Magnetometer	Attitude accuracy = 0.5 deg to 3 deg

Table 1: Sensor Accuracy Ranges. Adapted from Larson and Wertz [10]

Sensor	**Accuracy**	**Characteristics and Applicability**
Magnetometers	1.0°(5000km alt) 5.0°(200 km alt)	Attitude measured relative to Earth's local magnetic field. Magnetic field uncertainties and variability dominate accuracy. Usable only below ≈6,000 km.
Earth sensors	0.05°(GEO) 0.1°(LEO)	Horizon uncertainties dominate accuracy. Highly accurate units use scanning.
Sun sensors	0.01°	Typical field of view +-30°
Star sensors	2 arc-sec	Typical field of view +-6°
Gyroscopes	0.001 deg/hr	Normal use involves periodically resetting reference.
Directional antennas	0.01°to 0.5°	Typically 1 of the antenna beamwidth

Table 2: Sensor Accuracy Ranges. Adapted from Hall [12]

2 Preliminaries

2.1 Coordinate frames

2.1.1 ECI frame

The Earth Centered Inertial frame has its x-axis pointing towards the vernal equinox, and its z-axis pointing along the rotation axis of the Earth at some initial time. The y-axis completes a right handed orthogonal coordinate system. The frame's origin is at the center of the Earth. Larson and Wertz [10]

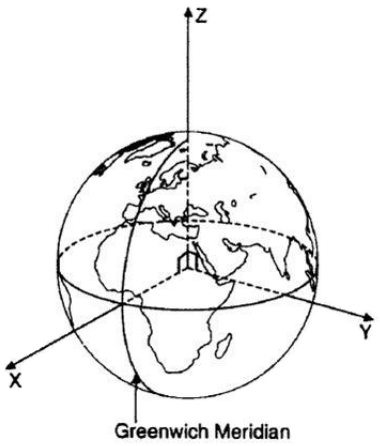

Figure 1: ECI frame, Image Larson and Wertz [10]

2.1.2 ECEF frame

This frame also has its origin at the center of the Earth, but the Earth Centered Earth Fixed frame has its x-axis pointing towards the point where the intersection between the longitude and latitude have zero value. It can also be described as the intersection between the Greenwich meridian and

the Equator. The frame's z-axis is pointing along the Earth's rotation axis. The y-axis completes the right handed orthogonal system. The ECEF frame is not an inertial frame, it rotates relative to the ECI frame along the Earth rotation.

2.1.3 NED frame

The North East Down frame has its z-axis pointing downwards, perpendicular to the tangent plane of the Earth's reference ellipsoid. The ellipsoid is mathematically defined and fitted for approximation of the Earth. The x-axis points towards true north and the y-axis points East. The NED frame is an inertial frame.

2.1.4 BODY frame

This frame is attached to the satellite, and is moving and rotating with it. The origin coincides with the origin of the NED frame. The axes coincide with the principle axes of inertia; the x-axis is pointing forwards, the y-axis is pointing to the right side and the z-axis is pointing downwards through the camera side of the satellite.

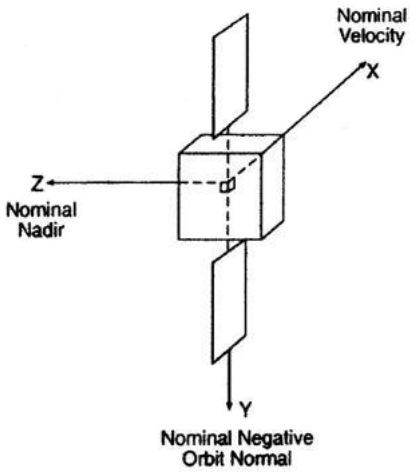

Figure 2: BODY frame, Image Larson and Wertz [10]

2.2 Space environment

2.3 Attitude representations

Several representations for describing attitude are available, the most common being Euler angles. More complicated attitude representations are quaternions. Quaternions are used for all the estimation methods presented in this thesis. They are singular-free, and are therefore well suited for attitude determination.

2.3.1 Euler angles

Euler angles were first described by Leonhard Euler in 1776, and are used to represent the orientation of a body Euler [13]. Three parameters are required for a full understanding of the orientation between two frames, one angle for the rotation around each of the axes. The angles are called roll, pitch and yaw and are usually written as phi, ro and psi. The Euler angles are often used for the definition of rotation matrices about the x, y and z-axis. In R 3 , the coordinate system rotations in a counter-clockwise direction looking towards the origin are given from:

$$\boldsymbol{R}_x(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \tag{1}$$

$$\boldsymbol{R}_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \tag{2}$$

$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

2.3.2 Quaternions

Hamilton [14] Cayley [15] Courant and Hilbert [16] Mebius [17] Barile [18] Shoemake [19] Horn [20]

Quaternions were first described by Sir William Rowan Hamilton in 1843 Hamilton [14]. His intension was to find an extension of vector algebra, and in 1845 Arthur Cayley published an article where he used multiplication of quaternions to describe rotations Cayley [15]. Three of the four elements of a quaternion give the coordinates for the axis of rotation, while the fourth is described by the angle of rotation Courant and Hilbert [16]. A quaternion can be written as a four-dimensional vector:

$$\boldsymbol{q} \coloneqq \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{4}$$

The real part of the quaternion behaves like a scalar in the three-dimensional vector space. Using a rotation angle vi, the real part can be written as:

$$q_0 = \cos(v/2) \tag{5}$$

$$\boldsymbol{n} = \frac{\boldsymbol{n}}{||\boldsymbol{n}||} \tag{6}$$

The imaginary part uses a unit vector given from n = n/——n—— (wzor) where the norm of the vector, ——n——, is defined as the square root of each of the squared elements of n added together. Throughout the thesis, vectors and matrices will be written in bold print. The imaginary part can be written as a vector:

$$\boldsymbol{q}_{vec} \coloneqq \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix} = [\boldsymbol{n} \sin(v/2)] \tag{7}$$

There are several ways to write quaternions. Sometimes it is convenient to think of a quaternion as the sum of a scalar and a vector written as:

$$\boldsymbol{q} \coloneqq q_0 + \boldsymbol{q}_v ec = q_0 + q_1 i + q_2 j + q_3 k \tag{8}$$

Complex numbers can be represented as matrices, and so can quaternions. A quaternion describes a point in 4D space, and can be represented by a 4 × 4 matrix by using a left-isoclinic rotation as proved in Mebius [17], and used in Barile [18] and Shoemake [19]:

$$\boldsymbol{Q} = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix} \tag{9}$$

The transpose of the matrix is the same as the conjugate of the quaternion:

$$\boldsymbol{q}^* \coloneqq q_0 - \boldsymbol{q}_{vec} = q_0 + q_1 i + q_2 j + q_3 k \tag{10}$$

Two quaternions are conjugate if they are orthogonal with respect to their inner product Horn [20]. The inverse of a quaternion q is defined as q -1 = 1/q . In this report, the mathematics are based on unit quaternions, which satisfies the constraint:

$$\boldsymbol{q}^T \boldsymbol{q} = 1 \tag{11}$$

The length of a unit quaternion is 1, which leads its inverse to be its conjugate.

2.3.3 Advantages of quaternions

The unit quaternion notation is compact, and round off errors are easier to handle than for matrix representation. The nearest orthonormal matrix to one that is not quite orthonormal, is difficult to find. Multiplying unit quaternions may similarly lead to quaternions that are no longer of unit length, but these can easily be normalized to make sure they correspond to valid rotations. The computational cost of normalizing a quaternion is much less than for normalizing a matrix. Quaternions are safe from a phenomenon called gimbal lock. When the pitch angle in a pitch/roll/yaw-system is rotated 90°up or down, and the yaw and roll correspond to the same motion, a degree of freedom of rotation can be lost. In a gimbal-based aerospace inertial navigation system, this could have disastrous results if the aircraft is in a steep dive or ascent Shoemake [19]. The quaternion elements vary continuously over the unit sphere in R^4 , (denoted by S^3) as the orientation changes, avoiding this problem. Due to the possible spin in the CubeSat, combined

with the singularity problem for Euler angles, an attitude estimation method based on unit quaternions is preferred.

3 Star-tracker program

Startracker program is designed to determine the attitude of the satellite with the image of the stars made by the satellite camera. Before the start of the mission star catalog is generated, based on the star catalogues obtained from the observatories on Earth and uploaded to the computer onboard the satellite.

After the start, the satellite is released from rocket's tank in space and has no idea where it is. Then the startracker on the satellite enters into Lost-In-Space mode (LIS) and analyzes the current image of the stars of the camera, and then searches the corresponding result of stars in the on-board star catalogue database. If it finds an entry in the database, the satellite goes into tracking mode. This means each next calculation of the orientation of the satellite attitude is going to be based on a comparison of the current pictures of stars with the preceding ones. If it cannot find the result in the database, this action is repeated from time to time, until a match is found in the database and the program will go into tracking mode.

Of course LIS does not happen only at the beginning of the satellite's flight, but can also result from many causes, eg. a satellite which is for a long time in the darkness may discharge its battery, and during the next entry into sunlit zone will turn on and look again its attitude. Another case is when satellite becomes lost, although once found the orientation and follow her. This happens, because the successive results are based on preceding ones and even the smallest mistake will grow until the satellite will not be able to correctly calculate their attitude. It may also happen that the satellite will rotate around its axis so fast that the program would not keep up with the processing the image and calculations. In this case, the corresponding other satellite's systems should take an action reducing his spin, but this is not part of this work. It is showed in simple conceptual diagram - Figure 3.

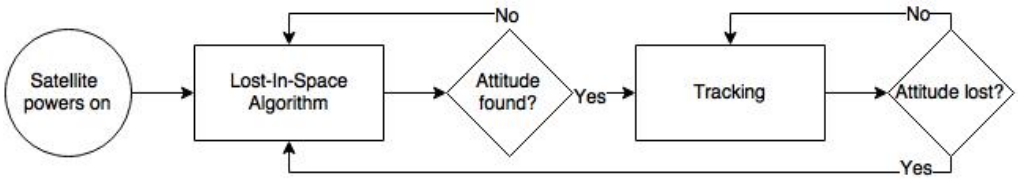

Figure 3: Startracker conceptual algorithm diagram

Each part - LIS and tracking - consists of smaller algorithms. Lost-In-

Space at the beginning of reading an image from the camera, thresholds the image. Thresholding is basicaly throwing away parts of the image which do not exceed the threshold brightness, making just bright enough stars taken into account. Next it calculates centroid (center of mass) of selected stars, identifies these stars using one of the possible methods (here Planar Triangle, described later) and searches the directory on-board technology (k-vector). If the database does not contain corresponding stars, the algorithm returns to the input state and starts analyzing the next image. If it manages to find the corresponding stars, the program determines the attitude of the satellite (by what angle is satellite shifted comparing to the referenced object - Earth or previuos image) and enters tracking mode. Tracking mode is in large part similar to the LIS. The algorithm also first analyzes the photo, selects and calculates centroids stars, identifies them, but does not search for stars in the directory board. In this algorithm are compared two images, one following after the other, and on this basis the current attitude is calculated.

not availableJu and Junkins [21]

Generally star-tracker is divided into three main partsMcBryde and Lightsey [22]:

- recogiting stars on the image and converting the data into list of star vectors by calculating star centroids;
- identyfing which star vector represents which real star in catalogue. This is done by comparing star vectors from the image with data in star catalogue, which is generated before space mission;
- estimating the attitude by calculating the displacement between two frames.

3.1 Centroid - start recognition

Samaan et al. [23]

Liebe [24]

The whole algorithm star-tracker is based on a very precise calculations. For this reason, the calculation based on the position of the pixels themselves

can give incorrect results. It is necessary to calculate the position of the stars with an accuracy exceeding pixels. This is why the calculation of the star centroids is necessary.

The first step is the determination of stars' position in the plane of the image. If focused star pictures are recorded, the image of each star will fall only on one or two pixels, and most likely the saturate these pixels, resulting in a pixel level accuracy. Many startrackers are doing deliberately blurred images, in order to spread the photons on a larger number of pixels, which allows the algorithm for calculating the centroid in subpixel accuracy.

After the registration of such image, the centroid of the star is found similarly to the centroid weightpoints array, with a few differences. Firstly, instead of weight light intensity is used. Secondly, the intensity of light is usually normalized by the pixels around the star in order to filter out glare or noise. The resulting outcome is a series of two-dimensional coordinates on the photo plane with the starting point at the center of the image. This allows the system to the coordinates of the stars can be easily converted into unit vectors in the next step.

The algorithm requires specification of the light intensity threshold (to select the brightest stars) I_{thresh} and the size of the Region of Interest (ROI) a_{ROI} in pixels. These values are adjusted to manipulate the performance of the algorithm. For example, the higher the value of I_{thresh} is more resistant to noise, but can miss some real stars in the picture. Similarly, a large value a_{ROI} means a more exact value of centroid, but the algorith can see one star there, wherea are actually two within a short distance of each other. The centroiding part is about trade-off between noise resistance and star recognition, and also between recognizing few close stars as one and recognizing one big star as a few. Please note that a_{ROI} must be a positive odd number for the proper functioning of the algorithm.

The idea of how to calculate such centroids is adapted fromMcBryde and Lightsey [22] and described below:

$$x_{start} = x - \frac{a_{ROI} - 1}{2} \tag{12}$$

$$y_{start} = y - \frac{a_{ROI} - 1}{2} \tag{13}$$

$$x_{end} = x_{start} + a_{ROI} \tag{14}$$

$$y_{end} = y_{start} + a_{ROI} \tag{15}$$

$$I_{bottom} = \sum_{i=1}^{x_{end}-1} I(i, y_{start}) \tag{16a}$$

$$I_{top} = \sum_{i=2}^{x_{end}} I(i, y_{end}) \tag{16b}$$

$$I_{left} = \sum_{j=1}^{y_{end}-1} I(x_{start}, j) \tag{16c}$$

$$I_{right} = \sum_{j=2}^{y_{end}} I(x_{start}, j) \tag{16d}$$

$$I_{border} = \frac{I_{top} + I_{bottom} + I_{left} + I_{right}}{4(a_{ROI} - 1)} \tag{16e}$$

$$\tilde{I}(x, y) = I(x, y) - I_{border} \tag{17}$$

$$B = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \tilde{I}(i, j) \tag{18}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{i \times \tilde{I}(i, j)}{B} \tag{19}$$

$$x_{CM} = \sum_{i=x_{start}+1}^{x_{end}-1} \sum_{j=y_{start}+1}^{y_{end}-1} \frac{j \times \tilde{I}(i, j)}{B} \tag{20}$$

$$u = \frac{\begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix}^T}{|| \begin{bmatrix} \mu x_{CM} & \mu y_{CM} & f \end{bmatrix} ||} \tag{21}$$

3.2 Star identification

all Spratling and Mortari [25]

Brightness Independent 4-Star Matching Algorithm for Lost-in-Space 3-Axis Attitude AcquisitionDong et al. [26]

SP-Search: A New Algorithm for Star Pattern Recognition Mortari [27]

Star Identification using Neural networks Miri and Shiri [28] Lindblad et al. [29]

Star pattern recognition using neural networks Li et al. [30]

3.2.1 Angle Matching

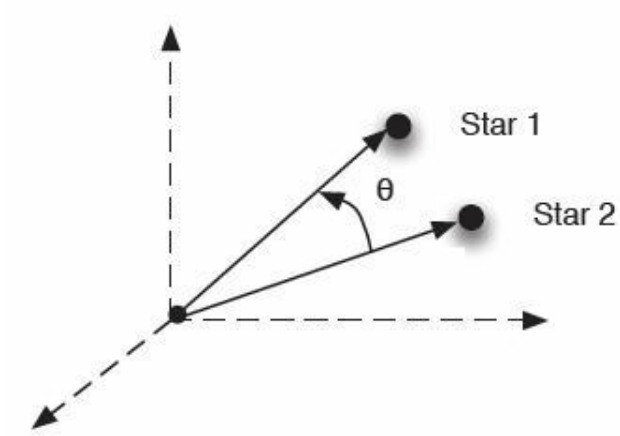

Figure 4: Vector angle method, Image Gottlieb [31]

Gottlieb [31]

$$\theta = \cos^{-1}(\boldsymbol{r}_1 \cdot \boldsymbol{r}_2) \tag{22}$$

$$\boldsymbol{b}_i = A\boldsymbol{r}_i \tag{23}$$

$$\tilde{\boldsymbol{b}}_i = A\boldsymbol{r}_i + \boldsymbol{v}_i, \quad \boldsymbol{v}_i^T A\boldsymbol{r}_i = 0 \tag{24}$$

$$E\{\boldsymbol{v}_i\} = 0 \tag{25a}$$

$$E\left\{\boldsymbol{v}_i\boldsymbol{v}_i^T\right\} = \sigma_i^2[\boldsymbol{I} - (A\boldsymbol{r}_i)(A\boldsymbol{r}_i)^T] \tag{25b}$$

$$\boldsymbol{b}_i^T\boldsymbol{b}_j = \boldsymbol{r}_i^T A^T A\boldsymbol{r}_j = \boldsymbol{r}_i^T\boldsymbol{r}_j \tag{26}$$

$$\tilde{\boldsymbol{b}}_i = A\boldsymbol{r}_i + \boldsymbol{v}_i$$
$$\tilde{\boldsymbol{b}}_j = A\boldsymbol{r}_j + \boldsymbol{v}_j$$

$$z \equiv \tilde{\boldsymbol{b}}_i^T\tilde{\boldsymbol{b}}_j = \boldsymbol{r}_i^T\boldsymbol{r}_j + \boldsymbol{r}_i^T A^T\boldsymbol{v}_J + \boldsymbol{r}_j^T A^T\boldsymbol{v}_i + \boldsymbol{v}_i^T\boldsymbol{v}_j \tag{28}$$

$$E\{z\} = \boldsymbol{r}_i^T\boldsymbol{r}_j \tag{29}$$

$$p \equiv z - E\{z\} = \boldsymbol{r}_i^T A^T \boldsymbol{v}_J + \boldsymbol{r}_j^T A^T \boldsymbol{v}_i + \boldsymbol{v}_i^T \boldsymbol{v}_j \tag{30}$$

$$\sigma_p^2 \equiv E\{p\} = \boldsymbol{r}_1^T A^T R_2 A \boldsymbol{r}_1 + \boldsymbol{r}_2^T A^T R_a A \boldsymbol{r}_2 + Trace(R_1 R_2) = Trace(A \boldsymbol{r}_1 \boldsymbol{r}_1^T R_2) + Trace(A \boldsymbol{r}_2 \boldsymbol{r}_2^T R_1) + Trace(R_1 R_2) \tag{31}$$

3.2.2 Spherical Triangle Matching

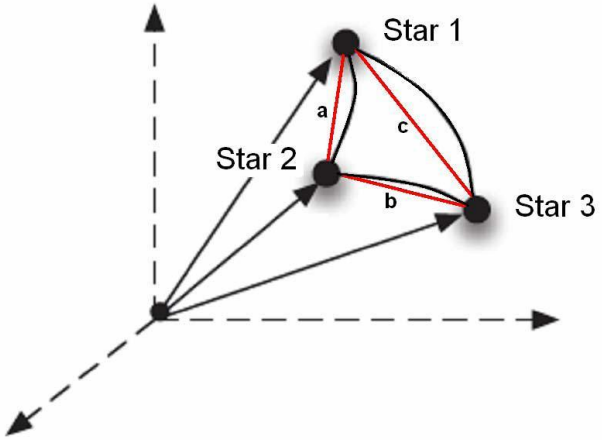

Figure 5: Spherical Triangle Method, Image Cole and Crassidus [32]

Cole and Crassidus [32]

$$A = 4 \tan^{-1} \sqrt{\tan \frac{s}{2} \tan \frac{s-a}{2} \tan \frac{s-b}{2} \tan \frac{s-c}{2}} \tag{32}$$

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

$$a = \cos^{-1}\left(\frac{b_1 \cdot b_2}{|b_1||b_2|}\right)$$

$$b = \cos^{-1}\left(\frac{b_2 \cdot b_3}{|b_2||b_3|}\right)$$

$$c = \cos^{-1}\left(\frac{b_3 \cdot b_1}{|b_3||b_1|}\right)$$

$$I_p = \sum \theta^2 dA \tag{34}$$

3.2.3 Planar Triangle

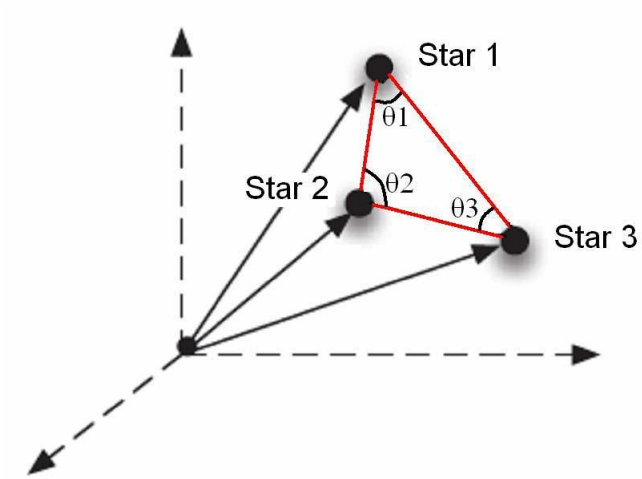

Figure 6: Planar Triangle Method, Image Cole and Crassidis [33]

Cole and Crassidis [33]

$$s = \frac{1}{2}(a+b+c) \tag{35a}$$

$$a = ||\boldsymbol{u_p} - \boldsymbol{u_q}|| \tag{35b}$$

$$b = ||\boldsymbol{u_q} - \boldsymbol{u_r}|| \tag{35c}$$

$$c = ||\boldsymbol{u_p} - \boldsymbol{u_r}|| \tag{35d}$$

$$A = \sqrt{s(s-a)(s-b)(s-c)} \tag{36}$$

$$J = A\frac{(a^2+b^2+c^2)}{36} \tag{37}$$

Derivatives

$$H = \begin{bmatrix} \boldsymbol{h}_1^T & \boldsymbol{h}_2^T & \boldsymbol{h}_3^T \end{bmatrix} \tag{38}$$

$$\boldsymbol{h}_1^T \equiv \frac{\delta A}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_1} + \frac{\delta A}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_1} \tag{39a}$$

$$\boldsymbol{h}_2^T \equiv \frac{\delta A}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_2} + \frac{\delta A}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_2} \tag{39b}$$

$$\boldsymbol{h}_3^T \equiv \frac{\delta A}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_3} + \frac{\delta A}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_3} \tag{39c}$$

$$\frac{\delta A}{\delta a} = \frac{u_1 - u_2 + u_3 + u_4}{4A} \tag{40a}$$
$$\frac{\delta A}{\delta b} = \frac{u_1 + u_2 - u_3 + u_4}{4A} \tag{40b}$$
$$\frac{\delta A}{\delta c} = \frac{u_1 + u_2 + u_3 - u_4}{4A} \tag{40c}$$

$$u_1 = (s-a)(s-b)(s-c) \tag{41a}$$
$$u_2 = s(s-b)(s-c) \tag{41b}$$
$$u_3 = s(s-a)(s-c) \tag{41c}$$
$$u_4 = s(s-a)(s-b) \tag{41d}$$

$$\frac{\delta a}{\delta \boldsymbol{b}_1} = (\boldsymbol{b}_1 - \boldsymbol{b}_2)^T/a, \quad \frac{\delta a}{\delta \boldsymbol{b}_2} = -\frac{\delta a}{\delta \boldsymbol{b}_1} \tag{42a}$$
$$\frac{\delta b}{\delta \boldsymbol{b}_2} = (\boldsymbol{b}_2 - \boldsymbol{b}_3)^T/b, \quad \frac{\delta b}{\delta \boldsymbol{b}_3} = -\frac{\delta b}{\delta \boldsymbol{b}_2} \tag{42b}$$
$$\frac{\delta c}{\delta \boldsymbol{b}_1} = (\boldsymbol{b}_1 - \boldsymbol{b}_3)^T/c, \quad \frac{\delta c}{\delta \boldsymbol{b}_3} = -\frac{\delta c}{\delta \boldsymbol{b}_1} \tag{42c}$$

$$\sigma_A^2 = HRH^T \tag{43}$$

$$R \equiv \begin{bmatrix} R_1 & 0_{3x3} & 0_{3x3} \\ 0_{3x3} & R_2 & 0_{3x3} \\ 0_{3x3} & 0_{3x3} & R_3 \end{bmatrix} \tag{44}$$

Polar Moment

$$\bar{H} = \begin{bmatrix} \bar{\boldsymbol{h}}_1^T & \bar{\boldsymbol{h}}_2^T & \bar{\boldsymbol{h}}_3^T \end{bmatrix} \tag{45}$$

$$\bar{\boldsymbol{h}}_1^T \equiv \frac{\delta J}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_1} + \frac{\delta J}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_1} + \frac{\delta J}{\delta A}\boldsymbol{h}_1^T \tag{46a}$$

$$\bar{\boldsymbol{h}}_2^T \equiv \frac{\delta J}{\delta a}\frac{\delta a}{\delta \boldsymbol{b}_2} + \frac{\delta J}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_2} + \frac{\delta J}{\delta A}\boldsymbol{h}_2^T \tag{46b}$$

$$\bar{\boldsymbol{h}}_3^T \equiv \frac{\delta J}{\delta b}\frac{\delta b}{\delta \boldsymbol{b}_3} + \frac{\delta J}{\delta c}\frac{\delta c}{\delta \boldsymbol{b}_3} + \frac{\delta J}{\delta A}\boldsymbol{h}_3^T \tag{46c}$$

$$\frac{\delta J}{\delta a} = Aa/18, \quad \frac{\delta J}{\delta a} = Ab/18, \quad \frac{\delta J}{\delta a} = Ac/18 \tag{47a}$$

$$\frac{\delta J}{\delta A} = (a^2 + b^2 + c^2)/36 \tag{47b}$$

$$\sigma_J^2 = \bar{H} R \bar{H}^T \tag{48}$$

3.2.4 Pyramid

Mortari et al. [34]

3.2.5 Rate Matching

Samaan et al. [35] to be removed?

3.2.6 Voting

Kolomenkin et al. [36]

3.2.7 Grid

Padgett and Kreutz-Delgado [37]

3.3 Star-catalogue and searching for matching stars

3.3.1 Star Catalogue Generation

$$\boldsymbol{u} = \begin{bmatrix} \cos\alpha\cos\delta \\ \sin\alpha\cos\delta \\ \sin\delta \end{bmatrix} \tag{49}$$

$$m_i \leq m_{max} \tag{50}$$

$$m_j \leq m_{max} \tag{51}$$

$$\boldsymbol{u_a^T u_b} \geq \cos\theta_{FOV} \tag{52}$$

3.3.2 Candidate Matching

to be removed?

3.3.3 Result Verification

to be removed?

3.3.4 k-vector

The k-vector database is built a priori for some given working magnitude threshold and for the star tracker maximum angular aperture. Essentially, the k-vector table is a structural database of all cataloged star pairs that could possibly fit in the camera FOV over the whole sky. The star pairs are ordered with increasing interstar angle. The data stored are the k index, the cosine of the interstar angle, and the master catalog indices I[k] and J[k] of the kth star pair. The k-vector access logic is invoked in real time for a minimal set of star pairs in elementary measured star polygons (three for a triangle, six for a four-star pyramid, etc.); the fact that the vertices between adjacent measured star pairs share a common cataloged star is the key observation leading to logic for efficiently identifying the stars by simply comparing the

k-vector accessed catalog indices from the several sets of candidate star pairs (which must contain the common measured pivot star, if it is in the catalog). Mortari and Rogers [38]

Mortari [39]

Mortari and Neta [40]

Trzeba dodać pogrubienia vectorów

$$z(x) = mx + q \tag{53}$$

$$m = \frac{y_{max} - y_{min} + \delta\epsilon}{n - 1} \tag{54}$$

$$q = y_{min} - m - \delta\epsilon \tag{55}$$

$$\epsilon \approx 22.2 \times 10^{-16} \tag{56}$$

$$\delta\epsilon = (n - 1)\epsilon \tag{57}$$

$$k(i) = j \quad where \quad s(j) \leq z(i) < s(j + 1) \tag{58}$$

or

$$k(i) = j \quad \text{where j is the greatest index such} \quad s(j) \leq y(I(i)) \quad \text{is satisfied.} \tag{59}$$

$$j_b = \left\lfloor \frac{y_a - q}{m} \right\rfloor \quad and \quad j_t = \left\lceil \frac{y_b - q}{m} \right\rceil \tag{60}$$

$$k_{start} = k(j_b) + 1 \quad and \quad k_{end} = k(j_t) \tag{61}$$

3.4 Attitude Determination

Jenssen et al. [1]

AIM (Attitude estimation using Image Matching)Delabie [3]

all Hall [12] Markley and Mortari [41]

3.4.1 q-method

$$\boldsymbol{s}_b = \boldsymbol{R}^{bi}\boldsymbol{s}_i \quad \boldsymbol{m}_b = \boldsymbol{R}^{bi}\boldsymbol{m}_i \tag{62}$$

$$\begin{aligned} J &= \frac{1}{2}\sum w_k(\boldsymbol{v}_{kb} - \boldsymbol{R}^{bi}\boldsymbol{v}_{ki})^T(\boldsymbol{v}_{kb} - \boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \\ &= \frac{1}{2}\sum w_k(\boldsymbol{v}_{kb}^T\boldsymbol{v}_{kb} + \boldsymbol{v}_{ki}^T\boldsymbol{v}_{ki} + 2\boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \end{aligned} \tag{63}$$

$$J = \sum w_k(1 - \boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki}) \tag{64}$$

$$g(\boldsymbol{R}) = \sum w_k\boldsymbol{v}_{kb}^T\boldsymbol{R}^{bi}\boldsymbol{v}_{ki} \tag{65}$$

$$\boldsymbol{R} = (q_4^2 - \boldsymbol{q}^T\boldsymbol{q})1 + 2\boldsymbol{q}\boldsymbol{q}^T - 2q_4\boldsymbol{q}^{\boldsymbol{x}} \tag{66}$$

$$\bar{\boldsymbol{q}}^T\bar{\boldsymbol{q}} = 1 \tag{67}$$

$$g(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T\boldsymbol{K}\bar{\boldsymbol{q}} \tag{68}$$

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{S} - \sigma\boldsymbol{I} & \boldsymbol{Z} \\ \boldsymbol{Z}^T & \sigma \end{bmatrix} \tag{69}$$

$$\boldsymbol{B} = \sum_{k=1}^{N} w_k(\boldsymbol{v}_{kb}\boldsymbol{v}_{ki}^T) \tag{70}$$

$$\boldsymbol{S} = \boldsymbol{B} + \boldsymbol{B}^T \tag{71}$$

$$\boldsymbol{Z} = \begin{bmatrix} B_{23} - B_{32} & B_{32} - B_{13} & B_{12} - B_{21} \end{bmatrix}^T \tag{72}$$

$$\sigma = tr[\boldsymbol{B}] \tag{73}$$

$$g'(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T \boldsymbol{K} \bar{\boldsymbol{q}} - \lambda \bar{\boldsymbol{q}}^T \bar{\boldsymbol{q}} \tag{74}$$

$$\boldsymbol{K}\bar{\boldsymbol{q}} = \lambda \bar{\boldsymbol{q}} \tag{75}$$

$$g(\bar{\boldsymbol{q}}) = \bar{\boldsymbol{q}}^T \boldsymbol{K} \bar{\boldsymbol{q}} = \bar{\boldsymbol{q}}^T \lambda \bar{\boldsymbol{q}} = \lambda \bar{\boldsymbol{q}}^T \bar{\boldsymbol{q}} = \lambda \tag{76}$$

3.4.2 Wahba's problem

The developed extended QUEST method described in this report builds upon the principles of Wahba's problem. The problem was first stated by Grace Wahba in 1965 Wahba [42]. Given two sets of vector observations, a rotation matrix $\boldsymbol{M}$ can be found which minimizes the orientation error. This is an optimization problem, where the cost function is:

$$\sum_{j}^{n} ||\boldsymbol{r}_j - \boldsymbol{M}\boldsymbol{b}_j|| \tag{77}$$

For satellite attitude determination, the vectors $\boldsymbol{r}_j$ for $j \in \{1, n\}$ are the reference sensor data given in the NED frame. The vectors $\boldsymbol{b}_j$ for $j \in \{1, n\}$ are the measured sensor data in the BODY frame. $\boldsymbol{M}$ is the least squares estimate of the rotation matrix which carries the known frame of reference into the satellite fixed frame of reference. The QUEST method uses this problem in order to minimize the attitude estimation error.

3.4.3 QUEST

improvement to quest implementation Cheng Yang and Shuster Malcolm D. [43]

kallman filtering Shuster [44]

$$J(\boldsymbol{q}) = \frac{1}{2}\sum_{j=1}^{n}\frac{1}{\sigma_j^2}(\boldsymbol{b}_j - \boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j)^T(\boldsymbol{b}_j - \boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j) = \frac{1}{2}\sum_{j=1}^{n}\frac{1}{\sigma_j^2}(\boldsymbol{b}_j^T\boldsymbol{b}_j - 2\boldsymbol{b}_j^T\boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j + \boldsymbol{r}_j^T\boldsymbol{r}_j) \tag{78}$$

$$J(\boldsymbol{q}) = \sum_{j=1}^{n}\frac{1}{\sigma_j^2}(1 - \boldsymbol{b}_j^T\boldsymbol{R}_b^i(\boldsymbol{q})\boldsymbol{r}_j) \tag{79}$$

3.4.4 TRIAD

a must

3.4.5 The Fast Optimal Attitude Matrix

to be removed?

3.4.6 DCM (Direction Cosine Matrix)

Juang et al. [45] and

McBryde and Lightsey [22]

$$\boldsymbol{B} = \sum_{\boldsymbol{i=1}}^{\boldsymbol{n}}\boldsymbol{b_i}\boldsymbol{r_i^T} \tag{80}$$

$$\boldsymbol{B} = \boldsymbol{USV^T} \tag{81}$$

$$\boldsymbol{U}_+ = \boldsymbol{U} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{U} \end{bmatrix} \tag{82}$$

$$\boldsymbol{V}_+ = \boldsymbol{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & det\boldsymbol{V} \end{bmatrix} \tag{83}$$

$$\boldsymbol{A} = \boldsymbol{U}_+ \boldsymbol{V}_+^{\boldsymbol{T}} \tag{84}$$

4 Prototype

For now the following parts are finished in Python:

1. Centroiding
2. Planar Triangle Recognition with variations (nearly - without equation 44)
3. Pyramid alg ?
4. k-vector
5. QUEST (not started yet)

Testing

Kruijff et al. [46]

5 Complete program

6 Testing of star-tracker

Kim et al. [47]

References

[1] K. L. Jenssen, K. H. Yabar, and J. T. Gravdahl, "A comparison of attitude determination methods: theory and experiments," in *proceedings of the 62nd International Astronautical Congress, Cape Town, South Africa*, 2011, pp. 3–7.

[2] R. G. Valenti, I. Dryanovski, and J. Xiao, "Keeping a Good Attitude: A Quaternion-Based Orientation Filter for IMUs and MARGs," *Sensors*, vol. 15, no. 8, pp. 19 302–19 330, 2015.

[3] T. Delabie, "A highly efficient attitude estimation algorithm for star trackers based on optimal image matching," in *AIAA Guidance, Navigation and Control Conference, Minneapolis, Minnesota*, 2012.

[4] E. Jalabert, E. Fabacher, N. Guy, S. Lizy-Destrez, W. Rappin, and G. Rivier, "Optimization of star research algorithm for ESMO star tracker," 2011.

[5] D. Felikson, J. Hahmall, M. F. Vess, and M. Ekinci, "On-Orbit Solar Dynamics Observatory (SDO) Star Tracker Warm Pixel Analysis," in *AIAA Guidance, Navigation and Control Conference, Portland, Oregon*, vol. 6728, 2011.

[6] M. W. Knutson, "Fast star tracker centroid algorithm for high performance CubeSat with air bearing validation," Ph.D. dissertation, Massachusetts Institute of Technology, 2012.

[7] A. Rose, "STAR integrated tracker," *arXiv preprint nucl-ex/0307015*, 2003.

[8] D. Mortari and A. Romoli, "StarNav III: a three fields of view star tracker," in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 1. IEEE, 2002, pp. 1–57.

[9] H. Heidt, J. Puig-Suari, A. Moore, S. Nakasuka, and R. Twiggs, "CubeSat: A new generation of picosatellite for education and industry low-cost space experimentation," 2000.

[10] W. J. Larson and J. R. Wertz, "Space mission analysis and design," Microcosm, Inc., Torrance, CA (US), Tech. Rep., 1992.

[11] S. M. R. C. P. Lima, "Comparison of small satellite attitude determination methods," 2000.

[12] C. D. Hall, "Spacecraft attitude dynamics and control," *Lecture Notes posted on Handouts page [online]*, vol. 12, no. 2003, 2003. [Online]. Available: http://www.dept.aoe.vt.edu/ cdhall/courses/aoe4140/attde.pdf

[13] L. Euler, "Formulae generales pro translatione quacunque corporum rigidorum," *Novi Acad. Sci. Petrop*, vol. 20, pp. 189–207, 1775.

[14] W. R. Hamilton, "LXXVIII. On quaternions; or on a new system of imaginaries in Algebra: To the editors of the Philosophical Magazine and Journal," 1844.

[15] A. Cayley, "XIII. On certain results relating to quaternions: To the editors of the Philosophical Magazine and Journal," 1845.

[16] R. Courant and D. Hilbert, "Methods of mathematical physics, Volume I," 1953.

[17] J. E. Mebius, "A matrix-based proof of the quaternion representation theorem for four-dimensional rotations," *arXiv preprint math/0501249*, 2005.

[18] M. Barile, "Conjugate elements," 1997. [Online]. Available: http://mathworld.wolfram.com/ConjugateElements.html

[19] K. Shoemake, "Animating rotation with quaternion curves," in *ACM SIGGRAPH computer graphics*, vol. 19, no. 3. ACM, 1985, pp. 245–254.

[20] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *JOSA A*, vol. 4, no. 4, pp. 629–642, 1987.

[21] G. Ju and J. L. Junkins, "Overview of star tracker technology and its trends in research and development," *Advances in the Astronautical Sciences*, vol. 115, pp. 461–477, 2003.

[22] C. R. McBryde and E. G. Lightsey, "A star tracker design for CubeSats," in *Aerospace Conference, 2012 IEEE*, March 2012, pp. 1–14.

[23] M. A. Samaan, D. Mortari, T. Pollock, and J. L. Junkins, "Predictive centroiding for single and multiple FOVs star trackers," *Advances in the Astronautical Sciences*, vol. 112, pp. 59–71, 2002.

[24] C. C. Liebe, "Accuracy performance of star trackers-a tutorial," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 587–599, 2002.

[25] B. B. Spratling and D. Mortari, "A survey on star identification algorithms," *Algorithms*, vol. 2, no. 1, pp. 93–107, 2009.

[26] Y. Dong, F. Xing, and Z. You, "Brightness independent 4-star matching algorithm for lost-in-space 3-axis attitude acquisition," *Tsinghua Science & Technology*, vol. 11, no. 5, pp. 543–548, 2006.

[27] D. Mortari, "SP-search: A new algorithm for star pattern recognition," *Advances in the Astronautical Sciences*, vol. 102, no. Pt II, pp. 1165–1174, 1999.

[28] S. S. Miri and M. E. Shiri, "Star identification using Delaunay triangulation and distributed neural networks," *International Journal of Modeling and Optimization*, vol. 2, no. 3, p. 234, 2012.

[29] T. Lindblad, C. S. Lindsey, Å. Eide, Ö. Solberg, and A. Bolseth, "Star Identification using Neural Networks," 1997.

[30] C. Li, K. Li, L. Zhang, S. Jin, and J. Zu, "Star pattern recognition method based on neural network," *Chinese Science Bulletin*, vol. 48, no. 18, pp. 1927–1930, 2003.

[31] D. Gottlieb, "Star pattern recognition techniques," *Spacecraft Attitude Determination and Control, The Netherlands*, pp. 257–266, 1978.

[32] C. L. Cole and J. Crassidus, "Fast star pattern recognition using spherical triangles," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit. Providence, Rhode Island: AIAA*, 2004.

[33] C. L. Cole and J. L. Crassidis, "Fast star-pattern recognition using planar triangles," *Journal of guidance, control, and dynamics*, vol. 29, no. 1, pp. 64–71, 2006.

[34] D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junkins, "The pyramid star identification technique," *Navigation*, vol. 51, no. 3, pp. 171–183, 2004.

[35] M. A. Samaan, D. Mortari, and J. L. Junkins, "Recursive mode star identification algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 41, no. 4, pp. 1246–1254, 2005.

[36] M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum, "Geometric voting algorithm for star trackers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 441–456, 2008.

[37] C. Padgett and K. Kreutz-Delgado, "A grid algorithm for autonomous star identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 202–213, 1997.

[38] D. Mortari and J. Rogers, "A k-vector Approach to Sampling, Interpolation, and Approximation," *The Journal of the Astronautical Sciences*, vol. 60, no. 3-4, pp. 686–706, 2013.

[39] D. Mortari, "A fast on-board autonomous attitude determination system based on a new star-ID technique for a wide FOV star tracker," *Advances in the Astronautical Sciences*, vol. 93, pp. 893–904, 1996.

[40] D. Mortari and B. Neta, "K-vector range searching techniques," *Adv. Astronaut. Sci*, vol. 105, pp. 449–464, 2000.

[41] F. L. Markley and D. Mortari, "How to estimate attitude from vector observations," 1999.

[42] G. Wahba, "A least squares estimate of satellite attitude," *SIAM review*, vol. 7, no. 3, pp. 409–409, 1965.

[43] Cheng Yang and Shuster Malcolm D., "Improvement to the Implementation of the QUEST Algorithm," *Journal of Guidance, Control, and Dynamics*, vol. 37, no. 1, pp. 301–305, 2013, doi: 10.2514/1.62549.

[44] M. Shuster, "Kalman filtering of spacecraft attitude and the QUEST model," *Journal of the Astronautical Sciences*, vol. 38, pp. 377–393, 1990.

[45] J.-N. Juang, H.-Y. Kim, and J. L. Junkins, "An efficient and robust singular value method for star pattern recognition and attitude determination," 2003.

[46] M. Kruijff, E. Heide, C. De Boom, and N. Heiden, "Star sensor algorithm application and spin-off," in *54th International Astronautical Congress of the International Astronautical Federation(IAF)*, 2003.

[47] J.-J. Kim, J. Tappe, A. Jordan, and B. Agrawal, *Star Tracker Attitude Estimation for an Indoor Ground-Based Spacecraft Simulator*, ser. Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, aug 2011, doi:10.2514/6.2011-6270.

List of Tables

List of Figures

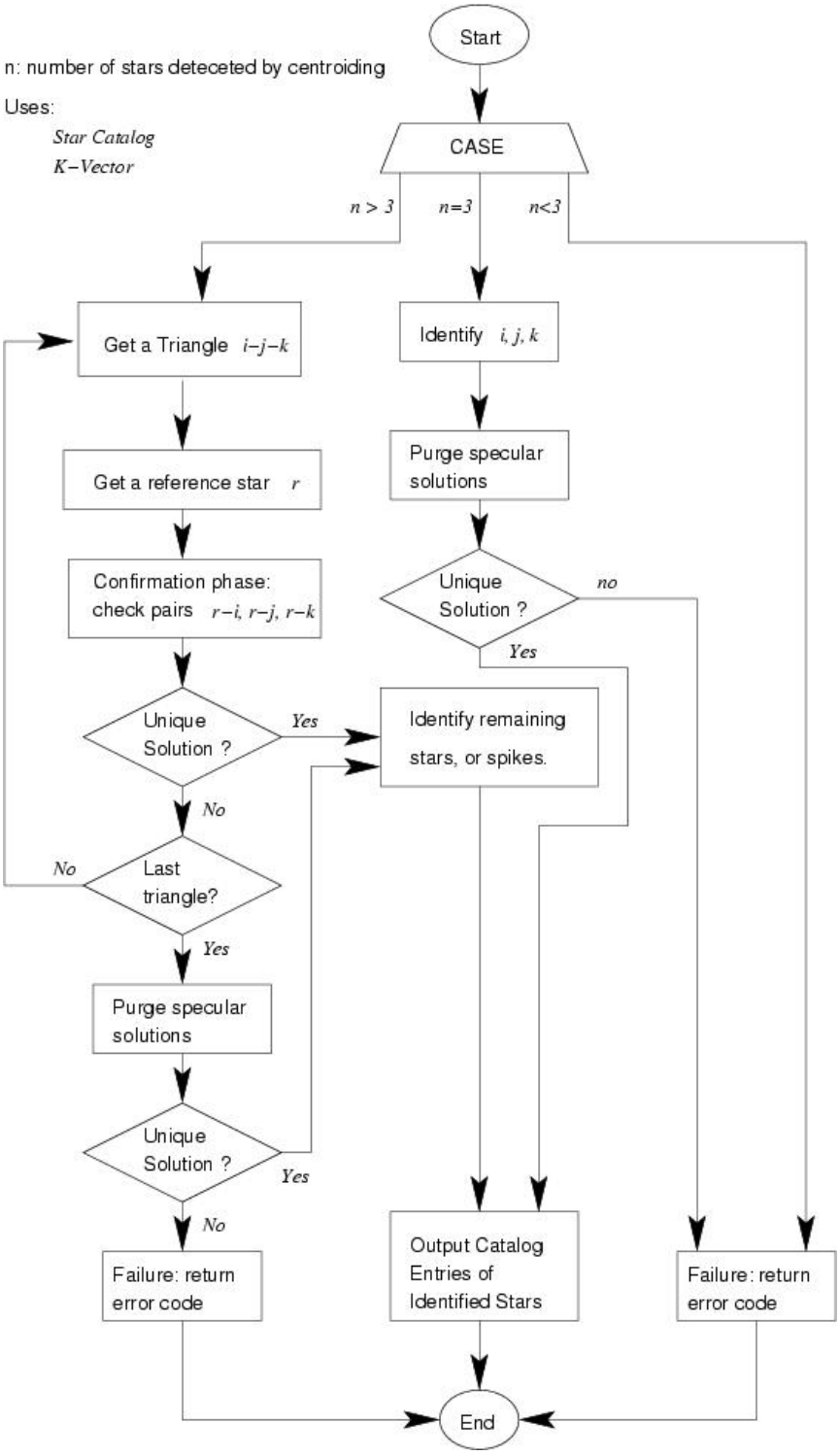

Figure 7: Pyramid Method Flowchart, Image Mortari et al. [34]