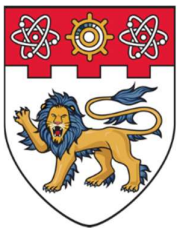


NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

A RELIABLE AND FAST LOST-IN-SPACE MODE STAR TRACKER

MEHTA DEVAL SAMIRBHAI

SCHOOL OF ELECTRICAL AND ELECTRONIC ENGINEERING

2019

A RELIABLE AND FAST LOST-IN-SPACE MODE STAR TRACKER

MEHTA DEVAL SAMIRBHAI

2019

A RELIABLE AND FAST LOST-IN-SPACE MODE STAR TRACKER

MEHTA DEVAL SAMIRBHAI

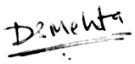
School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfillment of the requirement for the degree of
Doctor of Philosophy

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.



31st Jan 2019

. . . . . .

Date

. . . . . .

Mehta Deval Samirbhai

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

31st Jan 2019

. . . . . .

Date

. . . . . .

Assoc. Prof. Chen Shoushun

Authorship Attribution Statement

This thesis contains material from three papers published in the following peer-reviewed journals where I was the first author.

Chapter 4 and Chapter 5 of this thesis comprise of the algorithms, simulation and real-time testing results as published in the following journals:

- D.S. Mehta, S. Chen, K.S. Low, "A robust star identification algorithm with star shortlisting", *Advances in Space Research*, vol. 61(10), pp. 2647-2660, 2018.
- D.S. Mehta, S. Chen, K.S. Low, "A hamming distance and spearman correlation based star identification algorithm", *IEEE Transactions on Aerospace and Electronic Systems*, 2018.
- D.S. Mehta, S. Chen, K.S. Low, "A rotation-invariant additive vector sequence based star pattern recognition", *IEEE Transactions on Aerospace and Electronic Systems*, 2018.

The contributions of the co-authors are as follows:

- Assoc. Prof. Chen Shoushun provided the initial project direction for the star tracker development.
- I conducted the initial implementation and experiments to test the performance of the existing star tracking algorithms.
- I co-developed the proposed algorithms with Assoc. Prof. Chen Shoushun, along with the expert opinion and guidance of Prof. Low Kay Soon.

- The implementation of all the proposed algorithms, and the reported testing and performance (both simulations and real-time) for accuracy and time complexity were performed by me.
- The real-time implementation of the algorithm and configuring the real-time testing setup for the star tracker prototype was co-designed by Mr. Zhang Yexin and me.
- I prepared the manuscript drafts incorporating the research inputs and opinion of Assoc. Prof. Chen Shoushun.

31st Jan 2019

. . . . . .

Date

D.Mehta

. . . . . .

Mehta Deval Samirbhai

Acknowledgment

As I write this, I can see the journey of four years passing in front of my eyes. There have been many people involved both directly and indirectly in the progression of this thesis and shaping my research career in general.

Firstly, I am incredibly fortunate to have a supervisor like Prof. Chen Shoushun, who believed in me and gave the freedom to explore, execute, and experiment diversified ideas. At the same, he provided the necessary guidance with his unique insight and perspective, whenever I took up a wrong direction. During times of trouble, he has been incredibly patient and supportive. His way of analyzing a problem, research insight, approaching a solution has trained me expertly. He has been more than a supervisor during my research journey. I would also like to extend my thankfulness to the entire research group working on different projects. The weekly meeting discussions were immensely helpful to have a different perspective on new ideas. Specifically, I am grateful to Yexin, without whom the hardware setup of the star tracker would not be possible. The debugging sessions with him will be remembered.

I would also like to thank Prof. Low Kay Soon, the ex-director of the *Satellite Research Center* (SaRC) for his suggestions, ideas, and comments throughout the research journey. I will also take this opportunity to thank the entire research staff at SaRC, who helped me with any doubts I had regarding the satellite systems. I would also like to thank the current director of SaRC - Wee Seng for helping me with the funding issues and resources needed for this research project. Thanks are also due to the administration staff - Ms. Janet, and Ms. Pamela for their help in creating a comfortable workplace throughout the four years.

Special thanks are due to Chin Shi Tong (former staff at SaRC) for her significant help in the initial period of this research project. I am also thankful to my fellow peers of SaRC - Kashyap, and Khoa. All three of us started our respective research journey around the same time at the SaRC, and the time spent at SaRC will be memorable because of you two.

Administrative work in a big school is usually slow and unproductive. However, EEE GPO makes the above statement completely false. Thank you, Christina, during the period of Qualifying Examination. Jeniffer, you have always been speedy at processing the conference applications. Thanks to the superb staff at EEE GPO for making all the things handle fast and efficiently.

Friends outside research help to compensate for a daunting research life during Ph.D. Special thanks are due to my few and close friends - Sumanta, Sushant, Soumya, Zishan, and Shafquat. Thanks are also due to my undergrad friends from NIT Trichy. Though you guys were not here, the connection was always there.

On a personal note, I would like to thank three essential people in my life - my partner Emma, my mother Bina, and my father, Samir. Thank you for dealing with my mood swings, all those tantrums, and distress during the unhappy times. Thank you, Emma, for keeping me inspired throughout this journey. You have made me learned a lot and changed me for the better. It would not have been possible to complete this journey without the three of you.

Abstract

Developing an autonomous and reliable attitude determination system is one of the most crucial requirements for a satellite mission in space. Depending upon the satellite missions, the demand of the attitude (orientation) accuracy varies. Satellites meant for astronomical observations require the pointing precision in terms of arc seconds (1 arc second equals 1/3600 of a degree). Thus, it is necessary to develop a highly reliable and autonomous attitude determination sensors for such satellite missions. The central theme of this thesis is to develop such an attitude determination sensor - a star tracker.

A star tracker has become one of the most widely adopted attitude sensors in the previous two decades as it can provide the accuracy of orientation in terms arc seconds compared to the other attitude sensors such as the earth sensor, magnetometer, and sun sensor. It operates in two modes - Lost-in-space (LIS) mode and tracking mode. In LIS mode, it has to establish the initial attitude without any prior information. When the satellite is initially placed into the orbit or when it loses the attitude information due to some problem, the star tracker operates in the LIS mode to establish the attitude information. In the tracking mode, the star tracker obtains the attitude information from the LIS mode and then merely updates the attitude based on the subsequent images of the stars captured. Thus, LIS mode is critical to the reliable operation of the star tracker.

In the LIS mode of the star tracker, firstly star positions are determined in the image captured. This process is commonly known as the star centroiding process. Later, features are extracted from the star image. Finally, the stars are identified in the image captured by comparing the features extracted from the image with the

ones stored in the star pattern database (prepared offline). The attitude can be easily calculated by the existing QUEST or TRIAD methods once the stars are correctly identified in the image captured. Thus, the process of reliable attitude calculation depends upon identifying the stars correctly in the image captured. This process of identifying the stars in the image is popularly known as star identification or star pattern recognition in the research community. The algorithms and techniques developed for the same are known as the star identification algorithm or star pattern recognition technique.

A star identification algorithm is expected to provide a high identification accuracy, robustness, and low time complexity. It is quite challenging to achieve high robustness in the scenario of patch mismatch, magnitude uncertainty, positional deviation, and false stars present in the image captured. Moreover, there is a trade-off between achieving high robustness and low time complexity for identification. Hence developing a star identification algorithm or a star pattern recognition technique which can achieve both - high robustness and low time complexity has been a challenge for the researchers in the past two decades.

In the research work pursued in this thesis, firstly we analyze the problems faced - patch mismatch, magnitude uncertainty, positional deviation, and false stars in achieving a reliable star identification. We provide a quantitative analysis with a deep understanding of the gravity of the problems mentioned above. Having analyzed these problems, we develop a framework for achieving high robustness and low time complexity for the process of star identification. We propose *three novel approaches* for solving the problem of star identification. For initial testing of the developed approaches, we produce simulated star images (containing the problems faced) which closely resemble the real star images. Later, we implement the proposed methods on hardware and develop a LIS mode star tracker prototype. We configure a state-of-the-art star tracker hardware-in-loop testing system for testing any star tracker prototype. Finally, we test the performance of the LIS mode star tracker prototype on the real images captured by a star tracker SST-20S currently mounted on a satellite VELOX-CI.

The high accuracy of orientation provided by the star trackers has made them suc-

cessful to be adopted as attitude determination sensors for satellite missions. Making them reliable is a challenge, which we have addressed in the research work pursued in this thesis. Star trackers can also be adopted as attitude sensors for the emerging *nano and pico satellites*. In this thesis, we have identified a few research gaps faced in the process of star pattern recognition. We believe that the framework, techniques, and algorithms developed for LIS mode star tracker in this thesis will assist the research community to achieve better performance for attitude determination of satellites.

Contents

Acknowledgment **i**

Abstract **iii**

Contents **vi**

List of Figures **ix**

List of Tables **xiii**

List of Abbreviations **xiv**

List of Symbols and Notations **xvi**

1 Introduction **1**

1.1 Background and Motivation 1

1.2 Thesis Outline and Main Contributions 4

2 Star Tracker **7**

2.1 Star Tracker System 7

2.2 Star Catalog 8

2.3 Star Pattern Database (SPD) Generation 13

2.4 Basic Operation of a Star Tracker 16

2.5 Modes of Operation of Star Tracker 23

2.6 Summary 24

3 Research Problem and Literature Review **25**
3.1 Challenges Associated with Star Pattern Recognition 25
3.2 State-of-the-art Star Pattern Recognition Techniques 32
3.3 Performance Evaluation of Star Pattern Recognition Techniques . . . 52
3.4 Summary 54

4 Proposed Approaches **55**
4.1 Proposed Generalized Framework for Star Identification 55
4.2 Worst-case Patch Mismatch based Shortlisting and Running Sequential Angular based Star Pattern Recognition 57
4.2.1 Feature Extraction, Look-Up-Table and Star Pattern Database Construction 59
4.2.2 Shortlisting and Star pattern recognition 66
4.3 Image to Signal Approach 73
4.3.1 Feature Extraction, and Signal, Look-Up-Table and Star Pattern Database Construction 73
4.3.2 Shortlisting and Star Identification 79
4.3.3 Selecting 32 samples for Shortlisting and 512 Samples for Recognition 84
4.4 Rotation Invariant Vector Frame 86
4.4.1 Construction of Rotation Invariant 2D Vectors 87
4.4.2 Shortlisting Technique based on the Rotation Invariant Vector Frame 91
4.4.3 Star Identification based on Rotation Invariant Vector Frame . 95
4.5 Failure Analysis 111
4.6 Summary 113

5 Simulations and Real-time Testing **114**
5.1 Simulation, Testing and Benchmarking 114
5.1.1 Simulation Setup 114
5.1.2 Testing and Benchmarking 115

5.2 Real-time Testing 130
5.2.1 Testing Setup 131
5.2.2 Real-time Testing Results 133
5.3 Summary 138

6 Conclusion and Future Work **139**
6.1 Conclusion 139
6.2 Future Work 142

A Pseudo code of the Proposed Approaches **145**

Author's Publications **150**

Bibliography **152**

List of Figures

1.1 Satellite orientation in space 2

2.1 Block diagram of an autonomous star tracker 8
2.2 A commercial star tracker module 8
2.3 A part of the Yale bright star catalog 9
2.4 Earth Central Inertial (ECI) frame 10
2.5 RA and DEC angles in Earth Central Inertial (ECI) frame 10
2.6 Apparent magnitude scale 11
2.7 Star distribution 12
2.8 Visualization of stars 13
2.9 Projection of a star on the image plane 14
2.10 Basic operation modules of a star tracker 16
2.11 A real star image 17
2.12 Approximated Gaussian distribution of star 18
2.13 Calculation of a star centroid 19
2.14 Modes of operation of a star tracker 23

3.1 An identified real image 26
3.2 Study on real images for nearest stars 27
3.3 Study on real images for false stars 29
3.4 Study on real images for patch mismatch 30
3.5 Deviation in star positions because of satellite movement 31
3.6 Study performed to quantify the positional deviation in real star images 32

3.7 Feature extraction for Liebe algorithm 33
3.8 A part of the SPD for the Liebe algorithm 34
3.9 Feature extraction for Pyramid algorithm 36
3.10 Grid algorithm 37
3.11 SPD for Grid algorithm 37
3.12 Fast planar triangle algorithm 39
3.13 Geometric voting algorithm 41
3.14 Feature extraction for STOD algorithm 42
3.15 Tree SPD of STOD algorithm 43
3.16 Grid based back propagation neural network 44
3.17 Neural network self-organizing-map 46
3.18 Multilayer SOM neural network 46
3.19 Star pattern model based on Hidden Markov model 48
3.20 Selecting the pole star - Multi-plies algorithm 49
3.21 Accepted pole star - Multi-poles algorithm 50

4.1 Generalized framework for the developed approaches 56
4.2 Proposed algorithm flow for the first approach 58
4.3 Feature extraction for the first approach 59
4.4 Worst case patch mismatch simulation 61
4.5 Process of worst case patch mismatch simulation 63
4.6 Look-Up-Table and Star Pattern Database for the first approach . . . 66
4.7 Shortlisting technique for the first approach 67
4.8 Initial match and running sequential angular match 69
4.9 First approach in the scenario of magnitude uncertainty 70
4.10 Final algorithm flow- first approach 72
4.11 Proposed algorithm flow for the second approach 73
4.12 Feature extraction for the second approach 74
4.13 Signal, LUT and SPD construction from the extracted features 76
4.14 Look-Up-Table and Star Pattern Database for second approach . . . 79
4.15 An example of the image captured for explaining the second approach 80

4.16 Signal construction from the image and comparison with SPD 81
4.17 Calculation of hamming distance between I_{32} and LUT_{A32} 82
4.18 Shortlisting and star identification process - second approach 82
4.19 Comparison of performance of the second shortlisting approach for a different number of samples (N_{LUT}) 85
4.20 Comparison of performance of the second star identification approach for a different number of samples (N_{SPD}) 86
4.21 Construction of rotation-invariant vector frame for two dimensional (2D) vectors 88
4.22 Construction of star shortlisting technique database based on rotation-invariant vector frame 91
4.23 Propsoed shortlisting technique based on rotation-invariant vector frame 93
4.24 Star pattern database and shortlisting of star IDs based on rotation-invariant vector frame 94
4.25 Integration of the additive property with the rotation-invariant vector frame for the scenario of a missing star 96
4.26 Integration of the additive property with the rotation-invariant vector frame for the scenario of a false star 99
4.27 Importance of starting point for constructing the rotation-invariant additive vector sequence 102
4.28 Analysis of nearest stars for selecting the starting point 104
4.29 Generation of star pattern database for rotation-invariant additive vector sequence 106
4.30 Star pattern database for rotation-invariant additive vector sequence . 107
4.31 Star pattern recognition technique based on rotation-invariant additive vector sequence 108
4.32 Flowchart for the star pattern recognition technique based on rotation-invariant additive vector sequence 110

5.1 Performance of star pattern recognition techniques in the scenario of position deviation (1) 120

5.2 Performance of star pattern recognition techniques in the scenario of position deviation (2) 121
5.3 Performance of star pattern recognition techniques in the scenario of magnitude uncertainty (1) 123
5.4 Performance of star pattern recognition techniques in the scenario of magnitude uncertainty (2) 124
5.5 Performance of star pattern recognition techniques in the scenario of false stars (1) 126
5.6 Performance of star pattern recognition techniques in the scenario of false stars (2) 127
5.7 Performance of the proposed first star shortlisting technique 128
5.8 Performance of the proposed third star shortlisting technique 130
5.9 Schematic of hardware-in-loop test for the developed Lost-in-space mode star tracker 131
5.10 Developed Lost-in-space mode star sensor 133
5.11 Developed Lost-in-space mode star sensor 134
5.12 An identified real image by the first proposed approach 135
5.13 An identified real image by the second proposed approach 136
5.14 An identified real image by the third proposed approach 137

List of Tables

1.1 Potential orientation accuracy of different attitude sensors. 3

3.1 Specifications of SST-20S star tracker. 26
3.2 SPD for star $S1$ for the pyramid technique. 35

5.1 Benchmarking of star pattern recognition techniques - Ideal case. . . 117

List of Abbreviations

2D, 3D	Two-, Three- Dimensional
ADCS	Attitude Determination & Control System
BP	Back Propagation
COM	Center of Mass
CMOS	Complementary Metal Oxide Semiconductor
CNN	Convolutional Neural Network
DEC	Declination
ECI	Earth Central Inertial
FOV	Field of View
FPGA	Field Programmable Gate Array
GMV	Geometric Voting
GPS	Global Positioning System
HIP	Hipparcos
HMM	Hidden Markov Model
ID	Identification
JAXA	Japan Aerospace Exploration Agency
LCD	Liquid Crystal Display
LEO	Low Earth Orbit
LIS	Lost-in-space
LUT	Look-Up-Table
NASA	National Aeronautics and Space Administration
OPTG	Optimized Grid Algorithm
PR	Pattern Radius

QUEST	Quaternion Estimator
RA	Right Ascension
RNN	Recurrent Neural Network
ROI	Region of Interest
SAO	Smithsonian Astrophysical Observatory
SOM	Self Organizing Map
SDT	Shortest Distance Transform
SP	Starting Point
SPC	Star Pattern Code
SPD	Star Pattern Database
SPR	Star Pattern Recognition
STOD	Search Tree with Optimized Database
TRIAD	Triaxial Attitude Determination
UART	Universal Asynchronous Receiver Transmitter
YB	Yale Bright

List of Symbols and Notations

α	Right Ascension angle of the star.
δ	Declination angle of the star.
ρ	Pixel size of the image sensor.
σ	Spread of the Gaussian function.
bin_{LUT}	Bin number of the Look-Up-Table.
bin_{SPD}	Bin number of the star pattern database.
d_{max}	Worst-case patch mismatch difference between the image and the SPD.
e	Tolerance selected for matching.
f	Focal length of the lens.
g	Gain function for calculating the quaternion (q).
h	Column size of a sample image.
i	Row position of the pixel in the image.
j	Column position of the pixel in the image.
p_{hd}	Probability of correct star ID shortlisted by hamming distance.
p_{sh}	Probability of correct star ID shortlisted by vector sequence.
q	Quaternion of the attitude calculated.
r_n	Radial distance of the neighboring star from the center star.
r_s	Spearman correlation co-efficient calculated between two signals.
$slab_{LUT}$	Slab of the Look-Up-Table.
w	Row size of a sample image.

x	x coordinate of the star in the image.
x_c	x centroid coordinate of the star in the image.
x^{cam}	x coordinate of the star in the camera frame.
x^{ECI}	x coordinate of the star in the Earth Central Inertial frame.
y	y coordinate of the star in the image.
y_c	y centroid coordinate of the star in the image.
y^{cam}	y coordinate of the star in the camera frame.
y^{ECI}	y coordinate of the star in the Earth Central Inertial frame.
z^{cam}	z coordinate of the star in the camera frame.
z^{ECI}	z coordinate of the star in the Earth Central Inertial frame.
z_n	Number (n) of zeroes in the coded signal.
A_n	Relative angle of two consecutive stars made with the reference star.
C	Attitude conversion matrix.
D_H	Hamming distance between two signals.
I_{32}	32-sample signal formed from the image.
I_{512}	512-sample signal formed from the image.
I	Intensity of star in the image.
I_c	Center of the image.
I_{ij}	Intensity of the pixel *(i,j)* in the image.
J	Cost function for calculating the quaternion (q) attitude.
M_v	Apparent magnitude threshold selected for forming the database.
N_{hd}	Number of shortlisted star IDs by hamming distance.
N_{sh}	Number of shortlisted star IDs by vector sequence.
N_{LUT}	Number of bins selected for forming the LUT signal.
N_{SPD}	Number of bins selected for forming the *SPD* signal.
R_θ	Rotation matrix.
R^I_{sum}	Summation of radial distances (r_n) of the stars in the image.
V^T	Transpose of a vector.
$\widehat{V}$	Normalized vector.

V_b	Boundary vector between two consecutive neighboring stars in the image.
V_{bn}	Rotation-invariant boundary vector between two consecutive neighboring stars in the image.
V^{cam}	Star vectors in the camera frame.
V^{ECI}	Star vectors in the Earth Central Inertial frame.

Chapter 1

Introduction

1.1 Background and Motivation

"**Lewis spins out of control**" [1] - Lewis spacecraft was successfully launched and placed in the orbit by National Aeronautics and Space Administration (NASA) on August 23, 1997. Over the next three days after the launch, a severe malfunction of the attitude determination and control system (ADCS) occurred which led to improper spacecraft attitude, inability to charge the solar array, discharge of the batteries and eventually loss of command and control. The last contact with the Lewis spacecraft was made on August 26, 1997, after which the spacecraft re-entered the atmosphere and was destroyed 33 days later.

The above summary is the disastrous fate that no space agency wishes for their spacecraft or satellite after being successfully launched and placed in the orbit. The most crucial part which led to the failure of the contact with the Lewis spacecraft was the false attitude information provided by the ADCS system of the spacecraft. Apart from Lewis, there have been many cases where satellite missions have explicitly failed due to the malfunction of the ADCS system. [2] provided a comprehensive review of the failures of the satellite missions due to the ADCS system. Hence, developing an accurate and autonomous attitude determination system is one of the most crucial tasks of a satellite mission.

For proper functioning of a satellite, its orientation information needs to be reliable and accurate. Figure 1.1 shows the illustration of a Low Earth Orbit (LEO) satellite in space. Satellite attitude must be updated and calculated continuously during the revolution of the spacecraft around the Earth. Thus, the attitude determination system is one of the most critical subsystems of a satellite, necessary for satellite navigation, guidance, and control. Determining the attitude of a satellite implies estimating the orientation of the satellite by making remote observations of reference bodies available in space.

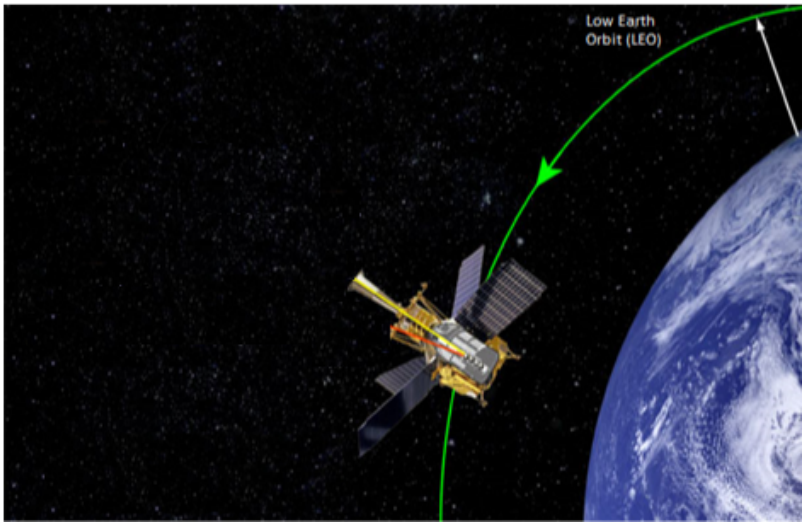


Figure 1.1: Satellite orientation in space. A Low Earth Orbit (LEO) satellite in space.

There are many reference objects in space which can be used for determining the attitude of the spacecraft. Commonly used sensors are the sun sensor, earth (horizon) sensor, and magnetometer [3, 4].The Global Positioning System (GPS) may also be equipped to provide the satellite's position and velocity vector. However, these sensors can only provide the precision in orientation information up to a certain limit as shown in Table 1.1. Star trackers have become widely popular and one of the most extensively used attitude determination sensor in the past two decades, as it can provide the orientation information in terms of arc seconds[1] [5,6]. Some satellite

[1]one arc second=1/3600 degrees.

missions such as communication satellites, astronomical observations require high precision in orientation information [7]. Hence, when a satellite mission requires an attitude of high precision in the range of arc seconds, star trackers (star sensors) are preferred over the other existing attitude determination sensors.

Reference Object	**Potential Orientation**
Stars	1 arcsecond
Sun	1 arcminute
Earth (Horizon)	6 arcminute
Magnetometer	30 arcminute

Table 1.1: Potential orientation accuracy of different attitude sensors.

"**Software error doomed Japanese Hitomi spacecraft**" [8] - Japan's flagship astronomical satellite Hitomi commanded a thruster jet to fire in the wrong direction, which resulted in accelerating, rather than slowing, the spacecraft's rotation. After five weeks of successful launch, at least ten pieces which included both the solar-array paddles broke off the satellite's main body. According to Japan Aerospace Exploration Agency (JAXA), Hitomi's troubles began in two weeks after the launch, with some severe glitches experienced in the star tracker of the satellite.

The Hitomi's satellite mission failure highlights a significant element in star tracker based ADCS system - the star tracker developed for providing the attitude of the satellite must be extremely reliable. Though a star tracker can give the attitude information in terms of arc seconds, the reliability of such attitude information is of prime importance. Thus, developing a reliable star tracker is a must for satellite missions which require high precision of attitude information. Hence the research work in this thesis aims to address the problems in developing a reliable star tracker.

The objectives of this thesis can be defined as below:

- To identify the major problems faced by a star tracker by analyzing the real star images captured from a currently mounted star tracker SST-20S on VELOX-CI satellite of Satellite Research Center, Nanyang Technological University, Singapore [9].

- To propose novel and diverse approaches to solve the identified problems, which

will eventually increase the reliability of the star tracker. The development of these approaches will lead to making of an accurate and fast star tracking system.

- To justify the proposed approaches by testing them on simulated images.
- To develop a prototype star tracker device based on the proposed approaches.
- To configure and develop a real-time testing setup for the star tracker prototype.
- To perform real-time testing of the prototype star tracker device developed.

The problems faced by a star tracker have been identified and presented for a long time. Our first aim is to provide an in-depth and quantitative analysis of these problems by analyzing the real star images. This analysis will enable us to point to the most significant challenge and hence develop the solutions accordingly.

Many diverse approaches have been developed in the past two decades for solving the problem associated with the star tracker. However, in these existing approaches, there is usually a trade-off between recognition reliability and time complexity of identification. We aim to develop a novel framework of techniques which addresses both the issues of recognition reliability and time complexity of a star tracker. We will also show the benchmarking of our proposed approaches with the state-of-the-art techniques on simulated star images.

Finally, we have developed a star tracker prototype based on the proposed approach. We have also configured a real-time testing setup for the prototype developed. We end the research work pursued in this thesis by performing the real-time testing of the developed star tracker prototype.

1.2 Thesis Outline and Main Contributions

In line with the objectives of this thesis, the main contributions of this research work are as follows:

- A quantitative and detailed analysis of the major problems faced by the Lost-in-space mode star tracker is performed, which will be helpful to the research community in identifying which specific problem is more important and up to what extent in different scenarios.
- A novel framework is presented for reliable and fast star identification.
- Three different novel approaches to star pattern recognition are developed based on the novel framework proposed.
- A Lost-in-space mode star tracker prototype is developed based on the proposed approaches and tested on the real star images.

This thesis is organized into six chapters for an easy and enhanced readability. The outline of this thesis is as follows:

Chapter 1: Introduction This chapter discusses the motivation behind the research goal pursued in this thesis. Along with the research goal and motivation, we also describe the precise objectives of this thesis and provide an outline briefly highlighting the significant contributions of this research work.

Chapter 2: Star Tracker In this chapter, we introduce a basic star tracker and its two operating modes. We explain in detail the functioning of a star tracker followed by identifying the critical mode of operation - Lost-in-space (LIS) mode. Later, we identify the problems faced by an LIS mode star tracker. Finally, the chapter is concluded by analyzing real star images captured from the star tracker SST-20S.

Chapter 3: Research Problem and Literature Review This chapter deals with the literature review performed for solving the problems faced by an LIS mode star tracker. State-of-the-art techniques developed in the past two decades in addressing the issues of an LIS mode star tracker are explained in detail in this chapter. Along with the deep understanding of the existing techniques and algorithms, each method is also followed by a critical review and failure analysis in the different scenarios of the problems identified in the Chapter 2.

Chapter 4: Proposed approaches This chapter forms the central theme of this thesis and is concerned with the novelty of the research work of this thesis. This chapter firstly introduces the framework for the proposed approaches. We have developed three novel techniques to solve the problems faced by an LIS mode star tracker. The algorithms developed are in accordance to make the LIS mode star tracker reliable and fast at the same time. Every proposed approach is explained in detail as well as some possible scenarios of potential failure analysis of the developed techniques are also described.

Chapter 5: Simulations & Real-time Testing In this chapter, we first explain the methodology and parameters dealing with the simulation of the star images to perform the initial testing of the proposed approaches. Later, we describe the hardware implementation of the proposed approaches and present the LIS mode star tracker prototype developed. Finally, we configure a real-time testing setup for the star tracker prototype developed and show real-time testing results.

Chapter 6: Conclusion & Future Work This chapter is the final chapter which realizes the goals of thesis and deduces the conclusion of the research work pursued. It lists the unique contributions of this research work to the research community working in the domain of star tracker. Finally, we also suggest a few prospects which can be pursued for extending the research work presented in this thesis.

Chapter 2

Star Tracker

In this chapter, we will demonstrate the basics of a star tracking system. Firstly, we will introduce the star tracker device. Later, we will list and explain some star catalogs along with the generation of the star pattern database (SPD) from the star catalog. After having described the offline process of SPD generation, we will then give the details of each step involved in the basic operation of a star tracker. Finally, we will describe the two modes of operation of a star tracker and identify the critical mode.

2.1 Star Tracker System

A star tracker is an optoelectronic device which determines the orientation of the satellite by comparing the features extracted from the star image captured with those stored in the database in an onboard computer. A basic star tracker unit consists of an image sensor to capture the image of the stars in space, and an onboard computer which stores the star pattern database (SPD). Fig 2.1 shows the basic structure of an autonomous star tracker unit. A commercial star tracker module - Hydra star tracker developed by SODERN [10] is shown in Fig 2.2.

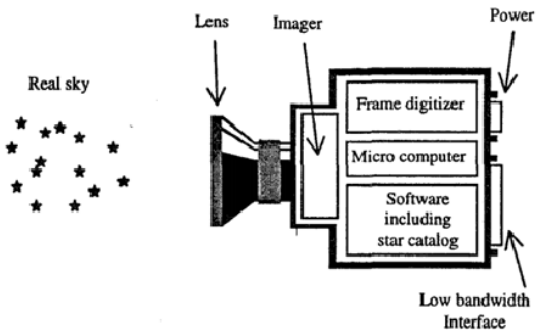


Figure 2.1: Block diagram of an autonomous star tracker (adapted from [11]).

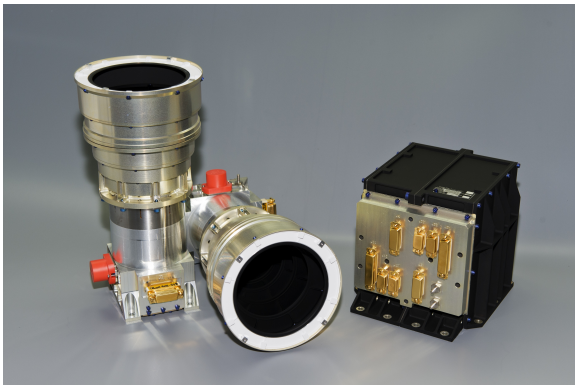


Figure 2.2: A commercial star tracker module from SODERN [10].

2.2 Star Catalog

A star catalog is an astronomical catalog that contains the information about all identified stars present in the space. There have been many different star catalogs which have been developed and maintained by various space agencies. Some of the popular ones are Smithsonian Astrophysical Observatory (SAO) Star Catalog [12], Yale Bright (YB) star catalog [13], and Hipparcos (HIP) catalog [14]. We can adopt any star catalog for forming the SPD. Fig 2.3 shows a part of the YB star catalog. As can be seen from Fig 2.3, a star catalog contains the star number (ID), right ascension (RA) angle, declination (DEC) angle, epoch, apparent magnitude (M_v), and other information concerning the star such as the relative velocity. Out of all the

mentioned information in the star catalog, we select the star ID and its corresponding RA, DEC, and M_v values for forming the SPD for the star tracker. The information mentioned in the star catalog apart from the above three is of not much importance for the preparation of the SPD. Before describing the process of forming the SPD and the basic operation of a star tracker system, we will first show the meaning of the RA, DEC and M_v values for a star ID in the star catalog.

Line# HR#	RA	DEC	Epoch	RA PM	DEC PM	MAG	Title HD #	Comments: Spec	Cross-ref	R-Velocity
1	00:05:09.90	+45:13:45.00	2000.00	-00.012	-00.018	6.70	3 #	A1Vn	BD+44 4550	RV -01
2	00:05:03.80	-00:30:11.00	2000.00	+00.045	-00.060	6.29	6 #	gG9	BD-01 4525	RV +01
3	00:05:20.10	-05:42:27.00	2000.00	-00.009	+00.089	4.61	28 #	K0III *	BD-06 6357	RV4-00
4	00:05:42.00	+13:23:46.00	2000.00	+00.045	-00.012	5.51	87 #	G5III	BD+12 5063	RV -00
5	00:06:16.00	+58:26:12.00	2000.00	+00.263	+00.030	5.96	123 #	G5V	BD+57 2865	RV7-01
6	00:06:19.00	-49:04:30.00	2000.00	+00.565	-00.038	5.70	142 #	G1IV	CD-4914337	RV0+00
7	00:06:26.50	+64:11:46.00	2000.00	+00.008	+00.000	5.59	144 #	B9III	BD+63 2107	RV -00
8	00:06:36.80	+29:01:17.00	2000.00	+00.380	-00.182	6.13	166 #	K0V	BD+28 4704	RV7-00
9	00:06:50.10	-23:06:27.00	2000.00	+00.100	-00.045	6.18	203 #	A7V	CD-23 4	RV +00
10	00:07:18.20	-17:23:11.00	2000.00	-00.018	+00.036	6.19	256 #	A6Vn	BD-18 6428	RV -00
11	00:07:44.10	-02:32:56.00	2000.00	+00.027	-00.002	6.43	315 #	B8III *	BD-03 2	RV +01
12	00:07:46.80	-22:30:32.00	2000.00	+00.052	-00.044	5.94	319 #	A2Vp:	CD-23 13	RV -01
13	00:08:03.50	-33:31:46.00	2000.00	-00.037	+00.000	5.68	344 #	K1III	CD-34 17	RV +00
14	00:08:12.10	-02:26:52.00	2000.00	+00.009	-00.003	6.07	352 #	K2III *	BD-03 3	RV +00
15	00:08:23.30	+29:05:26.00	2000.00	+00.136	-00.163	2.06	358 #	B8IVp *	BD+28 4	RV2-01
16	00:08:17.40	-08:49:26.00	2000.00	-00.052	-00.033	5.99	360 #	gG8	BD-09 5	RV +02
17	00:08:41.00	+36:37:36.00	2000.00	-00.100	-00.145	6.19	400 #	F8IV	BD+35 8	RV4-01
18	00:08:33.40	-17:34:39.00	2000.00	+00.000	-00.025	6.06	402 #	M0III	BD-18 3	RV -01
19	00:08:52.20	+25:27:46.00	2000.00	+00.114	+00.031	6.23	417 #	K0III	BD+24 3	RV +01
20	00:09:20.20	+79:42:53.00	2000.00	+00.102	-00.027	6.01	431 #	A7IV	BD+78 1	RV2+00
21	00:09:10.70	+59:08:59.00	2000.00	+00.525	-00.181	2.27	432 #	F2III *	BD+58 3	RV2+01
22	00:09:02.40	+18:12:43.00	2000.00	+00.137	-00.024	5.53	448 #	G9III	BD+17 7	RV -02
23	00:09:02.40	-54:00:07.00	2000.00	+00.051	+00.016	6.33	469 #	G4IV	CP-54 19	RV9+00

Figure 2.3: A part of the Yale bright star catalog [13].

A star in the space is usually specified in terms of the angles RA and DEC with respect to the earth central inertial (ECI) frame. As can be seen in Fig 2.4, every star in the star catalog is considered at the circumference of the earth, i.e. fixed distance from the center of the earth. The reason for the above consideration is because, in the star image captured (in the visible spectrum), it is not possible to estimate the distance of the star from the earth. Hence, every star can be characterized by the two angles RA and DEC which are shown in Fig 2.4 and Fig 2.5 (for clarity). The angle made by the projection of the star on the equatorial plane with the vernal equinox axis is labeled as the RA angle. The angle made by the star with the north pole axis is labeled as the DEC angle. Clearly, the RA angle will vary from 0^0 to 360^0; and the DEC angle will vary from -90^0 to +90^0. Hence, the position of a star with respect to the earth central inertial (ECI) frame is defined by the two angles-

right ascension (RA) and declination (DEC) as shown in Fig 2.4 and Fig 2.5.

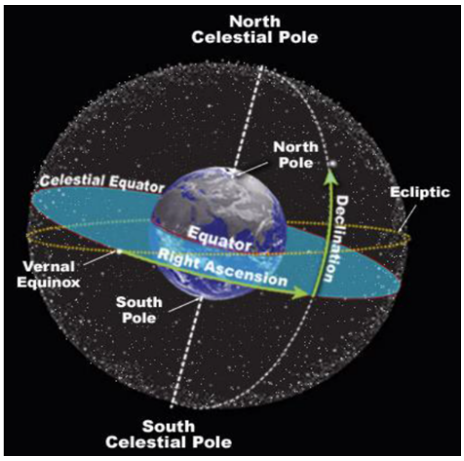


Figure 2.4: Star position in the earth central inertial (ECI) frame.

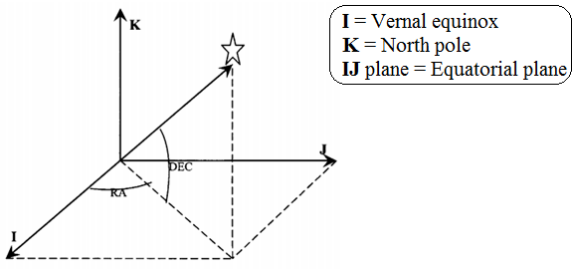


Figure 2.5: Depiction of the right ascension (RA) and declination (DEC) angles with respect to the ECI frame.

The brightness of a celestial body is one of the most fundamental observable quantity. As celestial bodies can have a vast range of intensity, a scaled system to classify brightness is adopted in astronomy. Historically, in astronomical science, the scaled system adopted is the logarithmic measure of the intensity of an object. Hence, the star magnitude is mentioned as the apparent magnitude (M_v), which relates to the standard of its brightness as seen from the Earth. In Fig 2.6, we show some important celestial bodies distributed along the apparent magnitude scale.

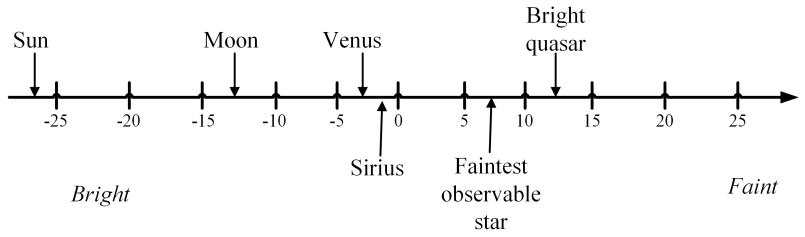


Figure 2.6: Apparent magnitude scale showing some eminent celestial objects.

The left-hand side of the scale corresponds to the brightest object (or) the most negative M_v value, and as we move towards the right, the brightness of the objects decreases and the corresponding M_v value increases. Most of the stars that can be detected by human eyes have M_v value of less than +6.3. Some eminent examples of the apparent magnitude of celestial bodies are: Sun = -27, Moon = -12.6, Sirius star = -1.4, and faintest naked eye star = +6.3. On this apparent magnitude scale, an interval of 1 corresponds to a factor of approximately 2.512 times of the actual brightness. This factor is calculated by the relation between the apparent magnitude and the actual brightness which is given by Eq 2.1 & Eq 2.2:

$$M_{v2} - M_{v1} = -2.512\,(\log_{10} I_2 - \log_{10} I_1) = -2.512 \log_{10}(\frac{I_2}{I_1}) \tag{2.1}$$

$$\frac{I_2}{I_1} = 2.512^{(M_{v1} - M_{v2})} \tag{2.2}$$

where M_{v1} and M_{v2} are the apparent magnitudes of both the stars in the catalog, and I_1 and I_2 are their corresponding brightness in the image.

Hence, while selecting the stars for preparing the SPD, only the stars which have an M_v value below a magnitude threshold are considered. The selection of the magnitude threshold is based upon the intrinsic properties of the image sensor to be adopted for the star tracker. A star distribution map for the stars having an M_v value of less than 6.0 is shown in Fig 2.7. The right ascension (RA) angle varies

from 0^0 to 360^0, and the declination (DEC) angle varies from -90^0 to 90^0 for the star distribution shown in Fig 2.7. The stars are adopted from the SAO catalog, and there are totally 4956 stars which have an M_v value of less than +6.

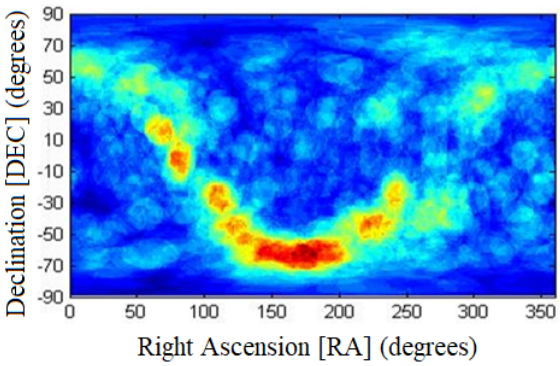


Figure 2.7: Distribution of the stars having an M_v value of less than 6.0 from SAO catalog.

As can be seen from Fig 2.7, the density of star distribution varies widely when focused in a certain region of the map. For example, the star density is quite high when the RA angle is between 150^{00} to 200^0, and DEC angle is between -70^0 to -50^0. This means that if the star tracker captures the image with the satellite oriented in the above region, the image captured will contain many stars. On the other hand, if one looks at the RA angle between 0^0 to 50^0, and DEC angle between -70^0 to -50^0, the star density is quite low. It implies that if the satellite is oriented in the above region, the image captured by the star tracker will contain less number of stars. Hence some part of the SPD will contain a higher number of stars to be considered for feature extraction whereas some other part of the SPD will contain fewer stars. In the next section, we will explain the process of generation of the star pattern database (SPD).

2.3 Star Pattern Database (SPD) Generation

The process of preparing the star pattern database is carried out offline. In Fig 2.8, we show the visualization of the stars distributed on the surface of the Earth, which is also known as the sky map. The sky map is divided into patches of the same size. The size and the shape of the patch depend upon the field of view (FOV) of the image sensor adapted for the star tracker. Thus, the FOV of a patch of the SPD must be the same as the FOV of the image sensor.

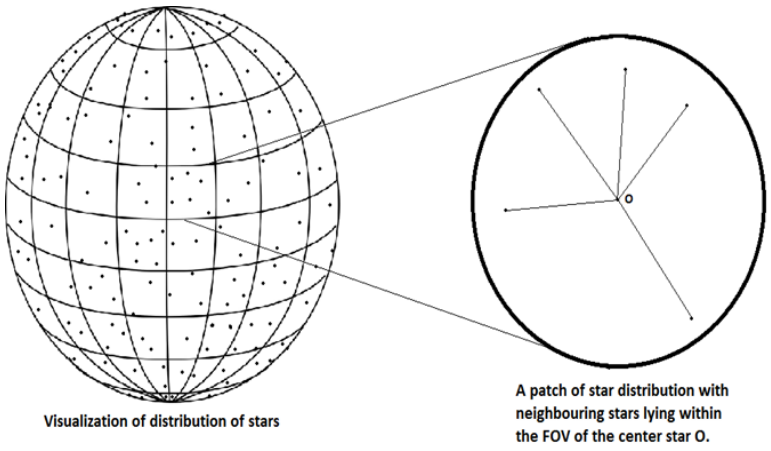


Figure 2.8: Visualization of stars on the surface of Earth.

It should be noted that the parameters of a star (RA and DEC angles) are in a three-dimensional (3D) ECI frame. However, the image captured by the star tracker is 2D in a camera frame. Hence, for comparison and matching, we first need to convert the star co-ordinates into the camera frame and later represent them on the image plane (2D). This conversion will depend upon the image sensor properties and is calculated accordingly. The projection of a star in 3D space onto the image plane is shown in Fig 2.9. The image sensor is assumed to be of the pin-hole characteristic, and the star projection on the image plane (X-Y) is carried out. In Fig 2.9, f is the focal length of the lens, X-Y is the image plane, (x,y) are the co-ordinates of the observed star on the image plane, and $\vec{r}$ is the light source of the observed star. The

focal length (f) of the camera is given by Eq 2.3.

$$f = \left(\frac{h}{2}\tan\left(\frac{FOV}{2}\right)\right) \times \rho \tag{2.3}$$

where h is the height of the image (focal plane), FOV is the field of view of the camera, and ρ is the pixel size of the image sensor. The first step is to convert the 3D co-ordinates of the star in the ECI frame to the camera frame. This conversion is given by a 3 x 3 attitude conversion matrix (C) [15] given by Eq 2.4.

$$\mathbf{C} = \begin{pmatrix} \frac{\sin\alpha}{\cos\delta} & -\frac{\cos\alpha}{\cos\delta} & 0 \\ \cos\alpha\tan\delta & \tan\alpha\sin\delta & -1 \\ \cos\alpha\cos\delta & \cos\delta\sin\alpha & \sin\delta \end{pmatrix} \tag{2.4}$$

where α and δ are the Right Ascension (RA) and Declination (DEC) angles of a star respectively.

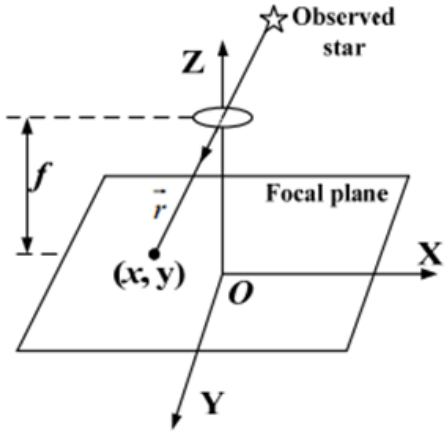


Figure 2.9: Projection of a star on the image plane.

After having calculated the attitude conversion matrix (C), we can represent the star co-ordinates in the camera frame as given by Eq 2.5.

$$V^{cam} = [x^{cam}; y^{cam}; z^{cam}] = C \cdot V^{ECI} \tag{2.5}$$

where V^{cam} are the star co-ordinates in the camera frame, and V^{ECI} are the star co-ordinates in the ECI frame.

Let (h,v) be the vertical and horizontal dimensions of the image sensor respectively. Once the star co-ordinates are calculated in the camera frame, Eq 2.6 gives the location of the x co-ordinate of the star on the image plane [15].

$$x = \frac{f}{\rho}\frac{x^{cam}}{z^{cam}} + \frac{v}{2} \tag{2.6}$$

Similarly the y co-ordinate on the image plane is given by Eq 2.7.

$$y = \frac{f}{\rho}\frac{y^{cam}}{z^{cam}} + \frac{h}{2} \tag{2.7}$$

After having located the star position on the image plane, the features are extracted for preparing the SPD. A patch of the sky map on the image plane can be viewed and zoomed as shown in Fig 2.8. In the example in Fig 2.8, we have selected a conical FOV. A star is considered at the center of the patch, and all the neighboring stars which lie within the FOV of the patch are considered for forming the star pattern for the center star. The star pattern is then prepared for the center star by extracting the features to be considered for comparison and pattern recognition. As mentioned earlier, only the stars which have a M_v value of less than the magnitude threshold selected are considered for forming the SPD. Usually, the magnitude threshold selected lies between +5.0 to +6.5. Depending upon the features to be extracted, each star above this magnitude threshold is appended in the SPD with its features extracted from the neighboring stars within the FOV. Once the SPD is prepared, it is stored in an onboard computer and later it is accessed for comparison of the features with that from the image acquired by the image sensor. Having ex-

plained the offline process of generation of the SPD, we will now describe the online process of star tracker in the next section.

2.4 Basic Operation of a Star Tracker

In Fig 2.10, we show the basic operation modules of a star tracker. We will now explain each module individually.

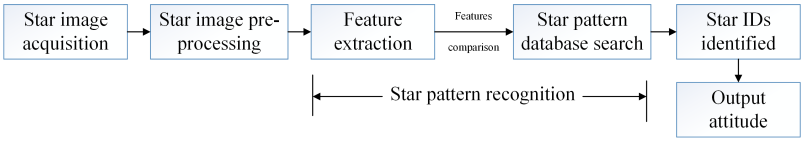


Figure 2.10: Basic operation modules of a star tracker.

1. **Star image acquisition**: The image sensor captures the image of the stars within a defined FOV. The parameters of the image captured depend upon the parameters of the image sensor. Usually, the image captured is of the size 512 x 512 pixels or 1024 x 1024 pixels. The bits per pixel are 8 or 10. The shape of the FOV is usually conical and the size of the FOV ranges from 8^0 x 8^0 to 15^0 x 15^0. The pixel size ranges from 5μm to 15μm, and the focal length lies between 50-100 cm. The exposure time of the image sensor selected is usually in *ms* corresponding to the exposure time of an ordinary CMOS image sensor. A real star image captured by the star tracker SST-20S [9] is shown in Fig 2.11.

2. **Star image pre-processing**: After star tracker captures the image, it must be pre-processed. The process of pre-processing the image is further divided into the following:

    - Thresholding: Every pixel of the star image captured has the value ranging from 0 (black) to 255 (white) considering we select 8 bits for representing a pixel value. An intensity threshold must be applied to remove any dark noise spikes which may have been generated by the intrinsic process of

Figure 2.11: A real star image.

the image sensor. Usually, the intensity threshold ranges from 0.3 to 0.6 of the maximum intensity value. Apart from the intensity threshold, an area threshold is also applied to the image captured. The area threshold corresponds to the number of pixels a star occupies in the image. Spikes generated from the image sensor and radiation effect from the space may occupy very low number of pixels in the image and hence should not be recognized as stars. Thus, they must be removed from further processing of the image. These spikes can be removed by selecting an area threshold for the stars. Usually, an area threshold of 4-8 pixels is selected. Thus, stars occupying a certain number of pixels having an intensity above the intensity threshold are only identified as the true stars and passed on the further steps of processing.

- Centroiding: The stars captured by the image sensor in the image behave like point sources. For an aberration-free lens system, a Bessel function

is usually approximated. However, due to the aberration in optical lens and image sensor, the energy distribution of the star in the image is not a Bessel function. Instead it is usually assumed to have a two-dimensional Gaussian distribution [16] as depicted in Fig 2.12 and Eq 2.8.

$$I(x,y) = \frac{I_0}{\sqrt{2\pi\sigma_{PSF}^2}} \exp\left[-\frac{(x-x_c)^2 + (y-y_c)^2}{2\sigma_{PSF}^2}\right] \tag{2.8}$$

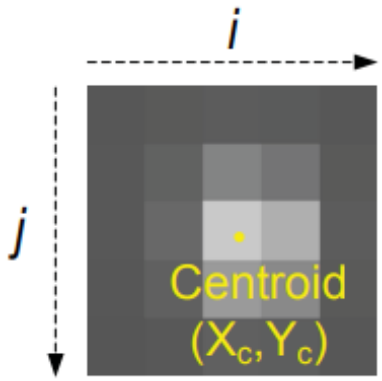


Figure 2.12: Approximated Gaussian distribution of star.

where x, y is the position of the star center, I_0 is the intensity of the star, and σ_{PSF} is the spread of the Gaussian distribution. The shape of the star occupies several connected pixels. The number of pixels for each star can range from as low as 5 to as high as 40. Thus, before extracting the features a star position has to be defined and pointed to a particular position in the x,y coordinate system. To point the location of a star in the image, centroiding operation is implemented over all the pixels that a star occupies in the image. By calculating the centroid of a star, the precision of the star position can achieve a sub-pixel level, and thus the star trackers can provide attitude solution in terms of arc seconds.

Centroid of mass (COM) algorithm [17] is the most common centroiding

algorithm adopted to determine the star locations in the image. After having applied an intensity threshold to the star pixels as described earlier, all the qualified pixels form a region of interest (ROI) for a star. A COM calculation is then employed on the all the pixels which fall under the ROI as shown in Fig 2.13. The COM is calculated by the Eq 2.9 & Eq 2.10.

$$x_c = \sum_{i,j} x_{ij} I_{ij} / \sum_{i,j} I_{ij} \tag{2.9}$$

$$y_c = \sum_{i,j} x_{ij} I_{ij} / \sum_{i,j} I_{ij} \tag{2.10}$$

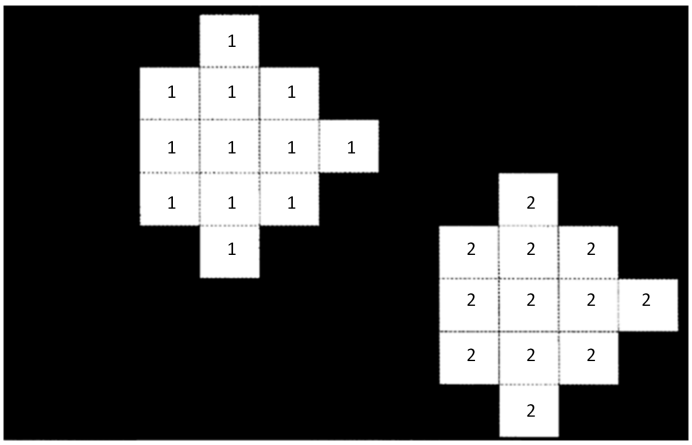


Figure 2.13: Distribution of a star in an image.

where x_c, y_c is the centroid calculated for the star and I_{ij} is the intensity of the pixel (i,j) in the ROI and x_{ij} and y_{ij} are its corresponding position in the x,y frame. This COM calculation is analogous to calculating the center of mass of a body. The calculation of centroid of a star always has some error. This error is due to the sensor noise, the effects of pixelization, and measurement uncertainty. We explain this introduction of error and

its effects on the feature extraction and comparison in detail in Chapter 3 which covers the research problem associated with the star tracker.

3. **Feature extraction and SPD search**: After having determined the centroid of each star in the image, the next step of feature extraction and comparison with the SPD is the most crucial step in the entire operation of star tracker. Features are extracted from the star positions in the image. This feature extraction process from the image is like the one adopted for preparing the SPD. Thus, depending upon the technique chosen for developing the SPD, the same procedure must be utilized for extracting the features from the image. Once the features are obtained from the image, they are searched for finding a match with the ones stored in the SPD. The search speed and accuracy of matching depend upon the technique adopted for feature extraction. Hence the process of feature extraction and comparison with the SPD forms the most crucial part of the star tracker. This step is also known as the star pattern recognition (SPR) or star identification algorithm in the research community. A detailed literature review of the state-of-the-art SPR techniques is provided later in Chapter 3 of this thesis.

4. **Star IDs identified and attitude calculation**: After having matched the features extracted from the image with the SPD, the stars in the image are identified as the ones which correspond to having obtained a maximum amount of match in the above step 3. The process of attitude calculation is the final step which is based on the stars identified in the image. Three to four stars need to be correctly identified in the image for calculating the attitude. The attitude is usually calculated by the existing TRIAD [18] or QUEST [19] methods. These methods take the body vectors (camera frame) of the stars identified in the image and the corresponding 3D vectors of the identified stars stored in the star catalog for calculating the attitude. After having identified three to four stars correctly in the image, the task of attitude calculation can be easily accomplished. We will explain the QUEST [19] method below.

   Suppose, there is a set of N unit vectors v_i (i=1,2...N) in the reference frame.

The corresponding vector in the body frame is represented by b_i (i=1,2...N). Then, the goal of the attitude determination method is to calculate the attitude matrix that converts the vectors from the reference frame to the body frame, i.e. finding the matrix A in Eq 2.11.

$$b_i = Av_i \tag{2.11}$$

The above equation will be over-determined in the case of $N > 2$. Thus, the problem of finding the attitude matrix A transforms into minimizing a cost function J given by Eq 2.12.

$$J = \frac{1}{2}\sum_{i=1}^{N} |b_i - Av_i| \tag{2.12}$$

The above cost function J can be re-written and given by Eq 2.13.

$$J = \sum_{i=1}^{N} \left(1 - b_i^T A v_i\right) \tag{2.13}$$

Thus, minimizing the cost function J transforms into maximizing the gain function g given by Eq 2.14.

$$g = \sum_{i=1}^{N} b_i^T A v_i \tag{2.14}$$

Hence the quaternion can be used to express this gain function given by Eq 2.15.

$$g = q^T K q \tag{2.15}$$

where the matrix K is given by Eq 2.16.

$$K = \begin{bmatrix} S - \sigma I & Z \\ Z^T & \sigma \end{bmatrix} \tag{2.16}$$

where σ is given by Eq 2.17, S is given by Eq 2.19, and Z is given by Eq 2.20.

$$\sigma = \sum_{i=1}^{N} b_i^T v_i \tag{2.17}$$

$$B = \sum_{i=1}^{N} b_i v_i^T \tag{2.18}$$

$$S = B + B^T \tag{2.19}$$

$$z = \sum_{i=1}^{N} b_i \times v_i \tag{2.20}$$

The quaternion attitude of the satellite can then be calculated by Eq 2.21.

$$q = \frac{1}{\sqrt{1 + |y|^2}} \begin{bmatrix} y \\ 1 \end{bmatrix} \tag{2.21}$$

where y is given by Eq 2.22.

$$y = \left[(\lambda_{\max} + \sigma)\, I - S \right]^{-1} Z \tag{2.22}$$

where λ_{max} corresponds to the maximum value of the eigen value of the matrix K, which will maximize the gain function and give an optimal solution.

2.5 Modes of Operation of Star Tracker

A star tracker operates in two modes of operation as shown in Fig 2.14.

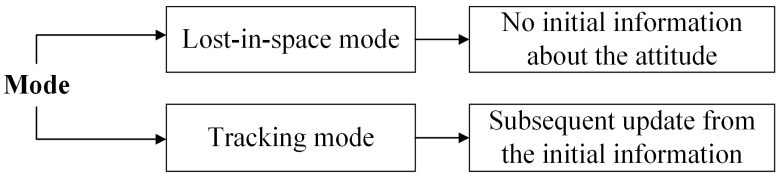


Figure 2.14: Modes of operation of a star tracker.

- **Lost-in-space (LIS) mode**: As the name suggests, the star tracker does not have any initial information about the attitude in this mode and is known as Lost-in-space. When the satellite is recently placed into the orbit or has accidentally lost its orientation information during its trajectory, the star tracker operates in this mode. Thus, the attitude of the satellite must be established or recovered depending upon the scenario. Hence the SPR techniques play a crucial part in determining the orientation of the spacecraft.

- **Tracking mode**: Once the initial attitude information is established or known, the star tracker enters a tracking mode. In the tracking mode, the star tracker keeps on updating the attitude based upon the subsequent images of the stars. It should be noted that the update of the orientation of the satellite depends upon the initial attitude information provided from the LIS mode of the star tracker.

From the above description, it should be clear that establishing the *initial attitude* in the LIS mode of the star tracker is more *challenging* compared to the constant update of the attitude in the tracking mode of the star tracker. As we have described earlier, calculating the attitude depends upon correctly identifying the stars in the image. Hence the SPR technique or the star identification algorithm developed for star pattern recognition forms the most *crucial* part of a star tracker.

The research purpose of this thesis can now be narrowed down to identifying the problems faced in the star pattern recognition, feature extraction, and comparison as this is the most crucial part of a star tracker. Our research goal is then to address the identified problems by developing a reliable and fast star pattern recognition technique. Based on the star pattern recognition technique developed, we aim to finally develop a fast and dependable prototype of Lost-in-space mode star tracker.

2.6 Summary

In this chapter, we explained in detail the basic operation of a star tracker. We first demonstrated the offline process of construction of the star pattern database (SPD) from the star catalog. Later, we described the online steps of the star tracker operation. We pointed out the most crucial online stage of the star tracker system - *feature extraction and comparison with the SPD*. We also described the two different modes of operation of the star tracker and concluded that the LIS mode of the star tracker is more critical and challenging compared to the tracking mode. Hence the next chapter will deal with the challenges faced in the LIS mode of the star tracker and a literature study of the existing star pattern recognition techniques.

Chapter 3

Research Problem and Literature Review

As described in the previous chapter, the LIS mode of the star tracker is more critical compared to the tracking mode of the star tracker. Concerning the LIS mode, the most crucial step is identifying the stars present in the image. This identification of stars in the image is accomplished by feature extraction and comparison with the SPD. Hence the star pattern recognition (SPR) technique or the star identification algorithm forms the most crucial part of the LIS mode star tracker. In this chapter, we will first identify the problems associated with the star pattern recognition by analyzing real star images. Later, we will perform an in-depth review of the existing diverse approaches to star identification. Finally, we will conclude the chapter by listing the essential parameters while evaluating a star pattern recognition technique.

3.1 Challenges Associated with Star Pattern Recognition

We performed a study on 1019 real star images captured by a star tracker SST-20S currently mounted on the satellite VELOX-CI [9]. The specifications of the SST-20S star tracker and the real image are listed in Table 3.1. For explaining the process of

identifying and analyzing the problems associated with star pattern recognition, we will be using an identified real image as shown in Fig 3.1.

Parameter	**Value**
Field of View (FOV)	15^0 x 15^0
SensitivityM_v	6.3
Star catalog	SAO J2000
Resolution (w x h)	1024 x 1024 pixels
Pixel size (ρ)	13μm
Focal length (f)	50 mm
Bits per pixel	8
Position accuracy	40 arcsec

Table 3.1: Specifications of SST-20S star tracker.

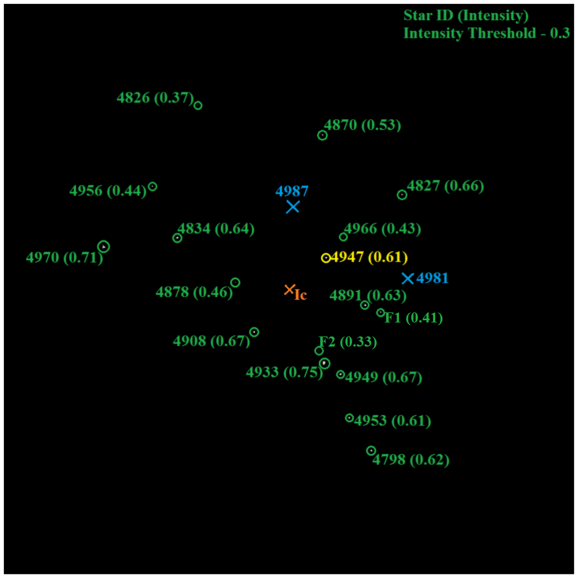


Figure 3.1: A real star image showing star ID and their corresponding intensity.

In Fig 3.1, the center of the image is marked with orange and labeled as I_c. The reference star (star ID: 4947) is chosen as the star nearest to the center of the image. The other stars present in the image are marked in green and labeled by their star ID (intensity) in the image. The intensity of the image varies from 0 (black) to 1.0 (255) being the white intensity. The intensity of each star in Fig 3.1 is the mean of

the intensity of the pixels occupied by that star in the image. As can be seen from Fig 3.1, the lowest intensity is that of star F2 (0.33), and the highest intensity is that of star ID 4933 (0.75). The standard deviation (σ) of all the star intensities is 0.122. The maximum intensity, minimum intensity and standard deviation of the intensities for all the 1019 images are like those for example in Fig 3.1. Thus, the intensity threshold was chosen to vary from 0.3 to 0.6 in the steps of 0.1 for the four different analysis described below:

1. **Number of nearest neighboring stars to the reference star** - We will now explain the calculation to check how many of nearest stars to the reference star is present with different intensity thresholds applied to the image using Fig 3.1.

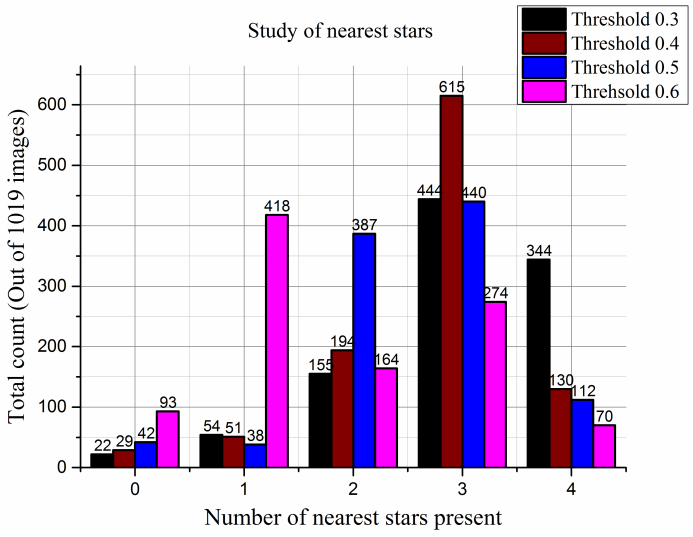


Figure 3.2: Study on real images for nearest stars.

For an intensity threshold of 0.3 in Fig 3.1, the reference star ID 4947 has only two nearest stars present, i.e. the star ID 4966 and star ID 4891. Star IDs 4987 and 4981 (both marked by a blue cross) are present in the SPD of star ID 4947. However, they are missed in the image captured. If we increase the intensity

threshold to 0.5 in Fig 3.1, then star ID 4966 will also cease to be present in the image, which will imply only one nearest star is present for star ID 4947 in the image. A similar analysis was done for all the 1019 images, and the results are shown in Fig 3.2. As can be seen from the analysis in Fig 3.2, even for a low-intensity threshold of 0.3, the cases that all the four nearest stars are present are still less (only 344 out of 1019 images). There were only half of the total images (615 out of 1019) which had three nearest neighboring stars in the image for intensity threshold of 0.4. The number of cases (22 out of 1019) when none of the nearest four neighboring stars are present in the image is low for the intensity threshold of 0.3 compared to that of higher intensity thresholds of 0.4, 0.5 and 0.6. If the intensity threshold increases, the cases of having all the four nearest stars to the reference star in the image decreases. Thus, it was concluded that a lower intensity threshold resulted in more number of nearest stars present compared to those in the scenario of a higher intensity threshold.

2. **Number of false stars in the image** - The second study performed is to check the number of false stars present in the image with varying intensity threshold. Planets, meteors, nearby satellites, and even space debris can be wrongly identified as stars while capturing the image in space. The reflected light from these objects cannot be distinguished from that of the stars in the image. For the case in Fig 3.1 at an intensity threshold of 0.3, there are two false stars present in the image labeled as *F1* and *F2*. If the intensity threshold increases to 0.4, only one false star *F1* will remain; and if the intensity threshold further increases to 0.5, both the false stars *F1* and *F2* will cease to exist in the image. A similar analysis was performed for all the images, and the results are shown in Fig 3.3.

   As can be seen from Fig 3.3, irrespective of a lower or higher intensity threshold, there is quite often a scenario when at least one false star is present in the image. If the intensity threshold is high (0.6), there are less number of cases of having false stars in the image. In fact, when the intensity threshold is 0.6, there is not a single case when there is more than one false star in the image. If the

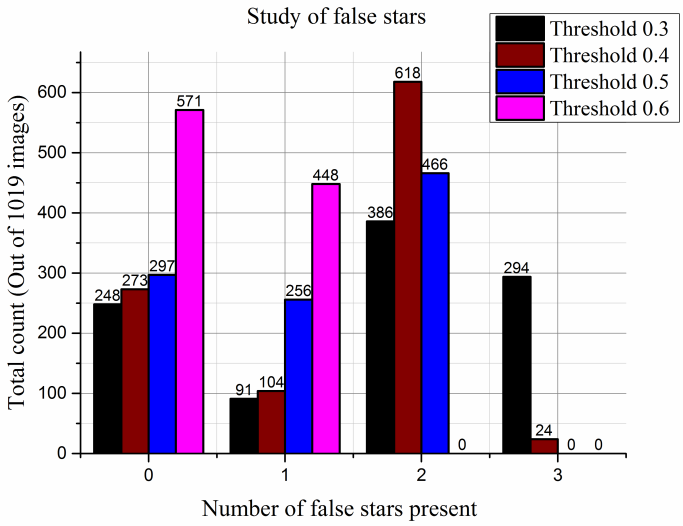


Figure 3.3: Study on real images for false stars.

intensity threshold is low (say 0.3), there are as many as 294 cases when three false stars are present in the image. This problem of false stars in the image can add to the earlier mentioned problem of missing of nearest stars, thus elevating the difficulty of correct star identification.

3. **Amount of patch mismatch in the image** - The third study performed is to analyze the amount of patch mismatch present in the captured star image. The image captured from the star tracker is translated and rotated as compared to the onboard SPD. The above situation leads to a certain number of stars been missed in the FOV of the reference star (in the image) when compared to the number of stars in the FOV of the reference star (in the SPD). The number of stars missed is termed as the amount of patch mismatch and is calculated in patch mismatch (%). For the example in Fig 3.1, the reference star ID 4947 has 24 neighboring stars in its SPD. However, because the reference star does not lie exactly at the center of the image only 14 (excluding the two false stars) of those 24 stars are present in the FOV of the star 4947. Thus, the patch mismatch (%) for star 4947 is around 41% for intensity threshold of 0.3.

If the intensity threshold is increased to 0.4 (and later to 0.5 and 0.6) the number of stars in the FOV of the reference star will decrease which results in a further increase of the patch mismatch. Fig 3.4 shows the analysis of the patch mismatch problem with varying intensity threshold on the real images

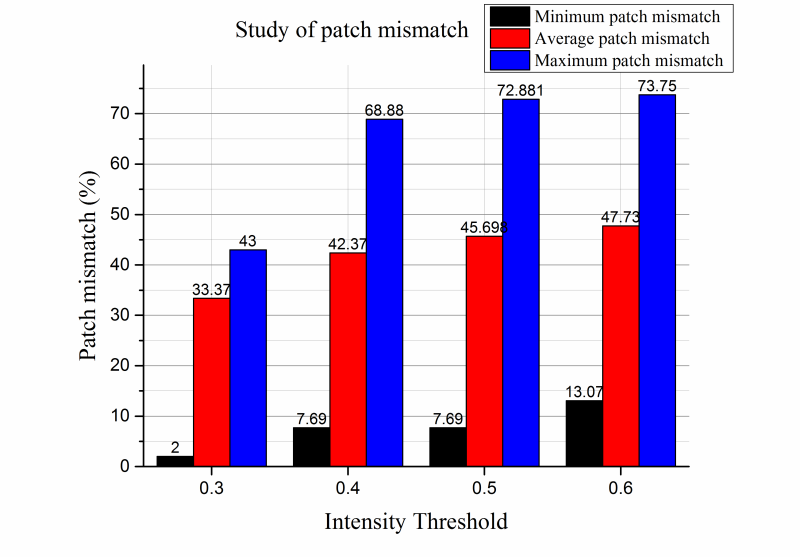


Figure 3.4: Study on real images for patch mismatch.

As can be seen from Fig 3.4, even with a low-intensity threshold, the average amount of patch mismatch (or) missing stars is 33.37%, which is quite high. If the intensity threshold is increased to 0.6, the number of stars missed will be more, and thus the average patch mismatch rises to 47.73% in this case. One of the images out of the 1019 real images, registered a patch mismatch as high as 73.75% with intensity threshold of 0.6.

4. **Pixel deviation in star positions** - The final study performed on real images was to check the amount of positional deviation in the star positions in the image when compared to that with the SPD. While capturing the images of the star, a satellite in orbit may go through some vibrations, unexpected velocity changes, and acceleration. These vibrations distort the star image captured.

This distortion is reflected in the positions of the stars in the image as shown in Fig 3.5. The distortion causes a smear (shift) in the assumed uniform Gaussian function of the star. Due to this shift in the Gaussian function, the centroid calculated for the star will not be accurate enough. Hence the features derived from the image will differ from those stored in the SPD.

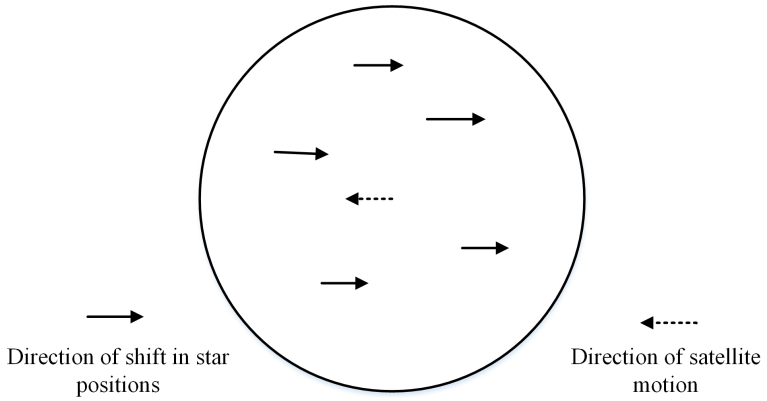


Figure 3.5: Deviation in star positions because of satellite movement.

To quantify the amount of positional deviation usually present in the star images, we perform a study for the same on 1019 already identified real star images. In this study, firstly we calculate the distance of each neighboring star from the reference star in the image. As we know the star IDs of these neighboring stars and the reference star in the image, we calculate the distance between them in the star map (or) the SPD. Later, we compare the two different sets of distances calculated and note the difference between the corresponding pairs in the two sets. Finally, we average the error in the distances for an image. We perform the same steps for all the 1019 real star images, and Fig 3.6 shows the results of the analysis.

In Fig 3.6, the *x-axis* represents the average positional deviation present in an image, and its value ranges from 0 pixels to 3.0 pixels in an interval of 0.5 pixels. The *y-axis* in Fig 3.6 denotes the count of the number of images (out of 1019 in total) for a specific interval of positional deviation. As can be

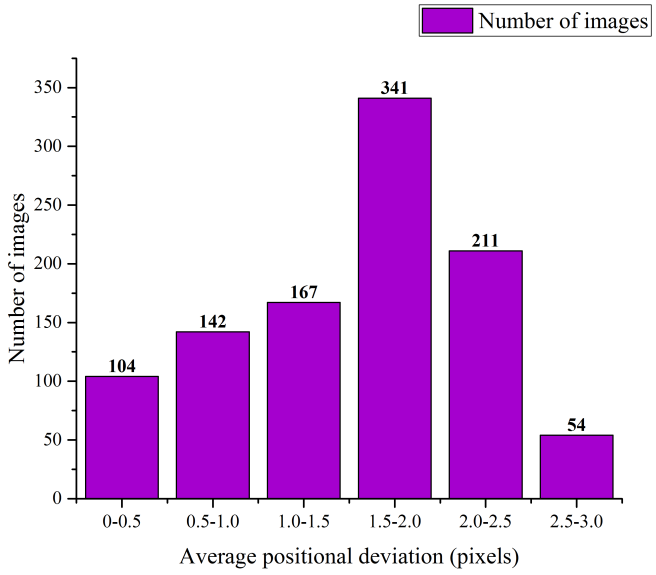


Figure 3.6: Study performed to quantify the positional deviation in real star images.

seen from the results in Fig 3.6, the maximum number of star images have an average positional deviation of 1.5 to 2.0 pixels. There are 211 images (20%) which have an average positional deviation of 2.0-2.5 pixels. Very few star images (54 out of 1019) have an average positional deviation as high as 2.5-3.0 pixels. The rest of the star images have an average positional deviation of fewer than 1.5 pixels. Thus, from this study, it can be concluded that the maximum positional deviation in a star image is bounded around 3.0 pixels. This value will be helpful while selecting the tolerance error for feature comparison.

3.2 State-of-the-art Star Pattern Recognition Techniques

Having performed qualitative and quantitative analysis on the real star images, we will now present the state-of-the-art star pattern recognition techniques. We assume that the pre-processing part of the star image remains the same for every method.

Thus, for every star pattern recognition technique, we will explain the process of feature extraction and comparison with the SPD. Later, we will also provide the failure analysis for each technique based on the problems identified with the process of star pattern recognition earlier.

1. **Liebe algorithm** - Liebe was the pioneer who proposed the star pattern recognition based on a triplet of stars [20–25]. The features are extracted from the star triplets. This algorithm utilizes three stars out of which one star is chosen as the guide star, and the other two stars are chosen as the closest neighboring stars to the guide stars. The feature vector extracted from this star triplet are the two distances and one angle as shown in the Fig 3.7. These features are calculated by the Eq 3.1, 3.2 and 3.3.

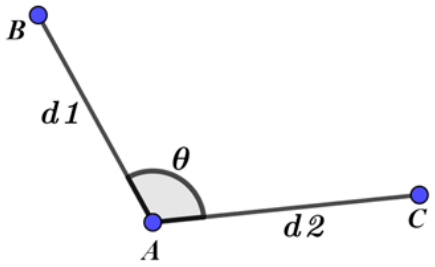


Figure 3.7: Feature extraction for Liebe algorithm.

$$\theta = \cos\left(V_{AB}, V_{AC}\right) \tag{3.1}$$

$$d1 = distance(A, B) \tag{3.2}$$

$$d2 = distance(A, C) \tag{3.3}$$

As can be seen in the Fig 3.7, the feature consists of the two distances from

the guide star (A) to the closest neighboring stars $d1(B)$, $d2(C)$ and the angle θ formed by the two neighboring stars with the guide star. The SPD for this pattern recognition technique is constructed as utilizing the above three mentioned features for each star. The SPD for the Liebe algorithm is shown in Fig 3.8. The first entry in the SPD (in Fig 3.8) is the angle made by the two closest neighboring stars with the guide star, and the next two entries are the distances of the two stars from the guide star.

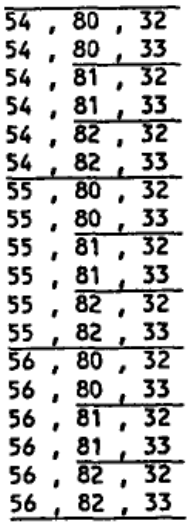


Figure 3.8: A part of the SPD for the Liebe algorithm [20].

In the image captured, the star closest to the center of the image is chosen as the guide star, and two nearest stars to the guide star are chosen for feature extraction as explained earlier. Once the features are calculated, they are looked upon in the SPD to find a match.

Problems with Liebe algorithm: Liebe algorithm fails to achieve the correct star pattern recognition in a lot of scenarios. The accurate star identification for Liebe algorithm depends upon the closest two nearest neighboring stars. Even if one of these two stars is missed to be captured in the image, the features calculated from the image would not match with those stored in the

SPD. Thus, finding the correct match will be difficult. As explained earlier when we identified the problems associated with star pattern recognition, there can be a scenario when as many as four nearest stars are missing in the image captured. In such situations, Liebe algorithm will fail drastically to provide a reliable recognition. Liebe algorithm will also fail in the case of patch mismatch and false stars present in the image for the same reason. Apart from the above problems, Liebe algorithm may also provide more than one match of star ID in the image. This multiple matching arises because of only three features considered for matching. Hence there can be more than one star having very similar features in the SPD and thus can get qualified for the match.

2. **Pyramid algorithm** - The idea of Liebe algorithm was extended by Mortari who proposed the pyramid algorithm [26–28]. The pyramid algorithm considers three closest nearest neighboring stars to the guide star for feature extraction as shown in Fig 3.9. As the name suggests, a four-star pyramid is formed and features of the pyramid, namely the six distances are utilized for comparison. As can be seen in Fig 3.9, $S1$, $S2$, $S3$, and $S4$ are the four stars considered for forming the pyramid with $S1$ being the guide star and the other three neighboring stars to star $S1$. For star $S1$, six inter-star distances [$p1$, $p2$, $p3$, $p4$, $p5$, $p6$] as shown in the Fig 3.9 are calculated as the features for comparison and identification purposes. These features are stored in the SPD for star $S1$ as shown in Table 3.2.

Feature	**Star 1**	**Star 2**
$p1$	$S1$	$S2$
$p2$	$S1$	$S3$
$p3$	$S1$	$S4$
$p4$	$S2$	$S3$
$p5$	$S2$	$S4$
$p6$	$S3$	$S4$

Table 3.2: SPD for star $S1$ for the pyramid technique.

In the image captured by the star tracker, the star closest to the center of the image is selected as the guide star to be identified and the three nearest

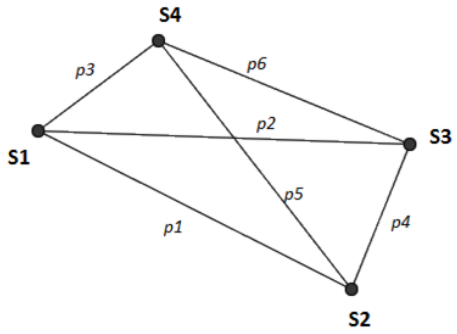


Figure 3.9: Feature extraction for Pyramid algorithm.

neighbors to the guide star are selected for calculating the inter-star distances. Once the features are calculated, they are compared with the ones stored in the SPD. The star ID in the SPD which matches all the six inter-star distances is identified as the correct guide star in the image.

Problems with Pyramid algorithm: The Pyramid algorithm is much more reliable compared to the Liebe as it considers the nearest three stars for feature extraction. Thus, the reliability of finding the correct match is much higher compared to that of Liebe algorithm. Concerning the multiple matches, the Pyramid algorithm will be quite accurate to provide a unique match compared to the Liebe algorithm. However, the algorithm still suffers from the same problems as the ones faced by the Liebe algorithm, i.e. missing of one of the nearest stars, patch mismatch, and nearest star being a false star.

3. **Grid algorithm**- The grid algorithm [29] was the pioneer to propose a pattern based star identification approach. The basic idea is to represent the guide star by a bit-pattern formed by the surrounding stars lying in the FOV of the guide star. The grid algorithm uses the bit-pattern to recognize the stars. The reference star S_i is considered at the center of the image and the surrounding stars lying in the FOV of the reference star are selected for the bit-pattern generation as shown in Fig 3.10.

   The bit pattern is generated by applying the grid size g on the image. The

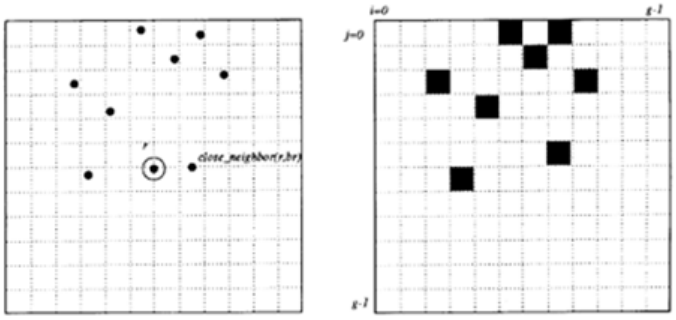


Figure 3.10: Construction of the bit-pattern for the Grid algorithm [29].

grids which contain the stars are marked as ON, and the grids without the stars are marked as OFF. The bit pattern for the guide star is the combination of ON and OFF pattern. The number of ON grids correspond to the number of neighboring stars lying in the FOV of the guide star. However, before forming the bit-pattern, a reference axis must be established as the image captured can be rotated when compared to the SPD. This reference axis is chosen by the nearest star to the guide star as the x-axis as shown in Fig 3.10. After having established the reference axis, the bit-pattern is formed by the location of the ON grids traveling from the top-left to the bottom-right as shown in Fig 3.10. For example, the bit-pattern formed for the guide star in Fig 3.10 is {5, 7, 18, ...}. These patterns are formed by considering every star as the guide star and are stored in the SPD as shown in Fig 3.11.

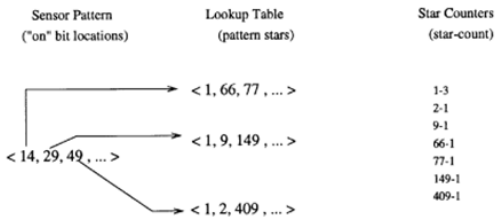


Figure 3.11: SPD and matching process for Grid algorithm (adapted from [29]).

Once the star tracker captures the image, a similar process to forming the SPD

is employed, and a bit-pattern is built for the image. The bit-pattern from the image is then compared with the ones stored in the SPD with all the star IDs having an initial counter of 0 before the process of matching begins. For every match of the bit-pattern, the counter is increased by one as shown in Fig 3.11. The guide star is identified as the star ID in the SPD which receives the maximum vote.

Problems with Grid algorithm: Grid algorithm considers all the stars present in the FOV for forming the bit-pattern and thus, is very robust to missing stars due to patch mismatch and the false stars. However, it has its concerns when it comes to forming the reference axis which depends upon the nearest star to the guide star. In the case that the nearest star is missed due to magnitude uncertainty or if the nearest star is a false star, the bit-pattern formed from the image captured is going to be completely different from the ones stored in the SPD of the guide star. Hence no match could be found in the SPD for such a scenario. Apart from the above scenario, due to the positional deviation of the stars in the image captured, the position of the star in the image and the SPD will be different. Thus, the matching of the bit-pattern between the image and the SPD will depend upon the grid size adopted. Hence the matching process is also highly sensitive to the grid size selected for this algorithm. Although the Grid algorithm suffers from the above disadvantages, this idea of bit-pattern approach was widely adopted by the research community, and many improved versions of the grid algorithm have been developed in the recent years.

4. **Fast planar triangle algorithm** - While the Liebe, Pyramid, and Geometric voting algorithm utilize the distances and angles between the stars as the features for preparing the SPD and comparison, a technique was proposed in [30] which employs the planar triangle area and spherical triangle polar moment of a star triplet for recognition as shown in Fig 3.12. The SPD for this method is constructed by taking the combination of triangles formed by three stars lying in the image sensor FOV and calculating the area and polar moment created by the triangle. The area and polar moment are computed by the Eq. 3.4,

3.5, 3.6, 3.7, 3.8, and 3.9.

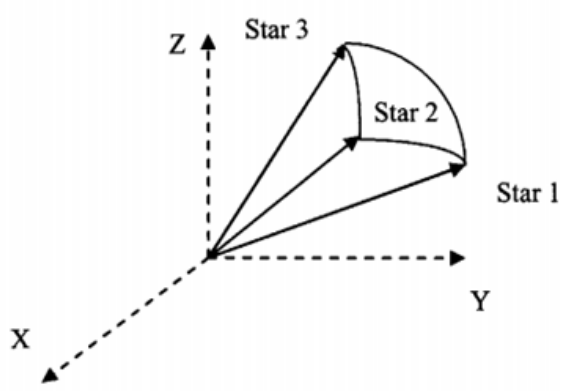


Figure 3.12: Feature extraction for Fast planar triangle algorithm.

$$A = \sqrt{s\,(s-a)\,(s-b)\,(s-c)} \tag{3.4}$$

where s is the semi-perimeter of the triangle given by:

$$s = \frac{1}{2}\,(a+b+c) \tag{3.5}$$

a,b, and c are the sides of a triangle given by:

$$a = |V_1 - V_2| \tag{3.6}$$

$$b = |V_2 - V_3| \tag{3.7}$$

$$c = |V_1 - V_3| \tag{3.8}$$

where V_1, V_2, V_3 are the corresponding vectors of the stars 1, 2, and 3 respectively. Polar moment of the triangle is then calculated by Eq. 3.9.

$$I = \frac{A \times (a^2 + b^2 + c^2)}{36} \tag{3.9}$$

In the image captured, the star closest to the center of the image is selected as the guide star. For extracting the area and the polar moment, the two nearest stars to the guide star in the image are selected. Once the area and the polar moment are calculated from the image, they are compared with the ones stored in the SPD. The matching star ID is identified as the guide star.

Problems with fast planar triangle algorithm: Although the fast-planar triangle utilizes two features - area and the polar moment, it still relies upon the nearest two stars for calculating these features. Hence this technique will also suffer from the same problems as those faced by the Liebe and Pyramid algorithm. A variation of the planar triangle was proposed in [31] which considered different orientations of the triangles for extracting the features.

5. **Geometric voting algorithm** - A star pattern recognition method based on voting of radial distances was proposed in [32]. This method considers all the neighboring stars which lie in the FOV of the guide star as opposed to the nearest two or three neighboring stars considered by Liebe and Pyramid algorithm previously. The feature utilized by the geometric voting method is the distance of each neighboring star from the guide star as shown in the Fig 3.13. $\{r_1, r_2 \ldots r_n\}$ are the distances from the center star which are utilized as the feature vectors for this technique. Hence, the star pattern database of this method comprises of the guide star ID and its corresponding distances of the neighboring stars which lie in the FOV.

   Once the image is acquired, the star nearest to the center of the image is selected as the guide star to be identified. All the stars whose distance r_i is less than the FOV are considered as the neighboring stars to the guide star. The distances of the neighboring stars are calculated from the guide star. Each star ID in the SPD is initialized with a zero count. The distances in the image

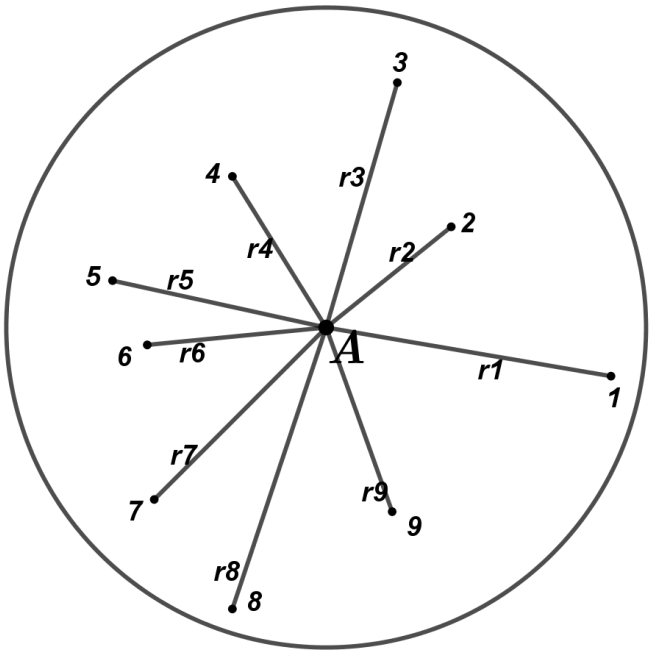


Figure 3.13: Feature extraction for Geometric voting algorithm.

are compared with each star entry in the SPD and are a match if they lie in a defined region of r_i (SPD) - $e < r_i$ (image) $< r_i$ (SPD) + e, where e is a chosen tolerance value. For every match of the distance, the count for that star ID is incremented by one. This process is carried out for the whole SPD. The star ID which has the highest count after the process is selected as the correct star ID for the guide star in the image. This is followed by a validation stage in which the distance of the guide star and the star nearest to the guide star in the image is compared with that of the identified star ID in the SPD. If it matches, then the star entry is recognized as the correct star ID.

Problems with Geometric voting algorithm - The technique employed by the geometric voting method will eliminate the problems of patch mismatch and false stars. However, if the number of stars captured in the image is less than the technique fails to recognize the guide star correctly as the highest vote usually gets assigned to the star ID in the SPD which has higher number of

neighbors in its FOV. This is the main drawback of this technique. Apart from this, the method may return multiple star IDs in some cases. Adding to the above two, the geometric voting technique will require significant memory to store the SPD because it considers each neighboring star which lies in the FOV of the guide star.

6. **Star tree with optimized database** [15,33] proposed a unique approach for forming the SPD and storing the features. Popularly known as the star tree algorithm, the search tree with optimized database (STOD) technique stores the features in a hierarchical tree-like structure.

ID	N	D_1	D_2	D_3	D_4	D_5
1284	5	5	12	28	29	30
1286	5	5	13	23	26	31
292	5	6	11	23	29	30
1622	5	6	11	25	27	30
1395	5	6	13	20	20	27
1599	5	6	16	21	24	31
27	5	6	17	22	22	27
66	5	6	18	26	29	31
629	5	6	22	23	27	29
71	5	6	23	23	24	27
531	5	7	11	19	21	26
583	5	7	15	22	24	31

Figure 3.14: Feature extraction for STOD algorithm [15].

The features extracted by this algorithm are the number of nearby stars (N) and the distances of those neighboring stars from the guide star as shown in Fig 3.14. Later, these features are arranged in a tree-structured SPD as demonstrated in Fig 3.15. As can be seen in Fig 3.15, the top layer of the tree is the number of neighboring stars which lie in the FOV of the guide star. From the next layer onwards, the distances of those nearby stars from the guide star

are stored in increasing order. Thus, the layer D1 in Fig 3.15 will have the distance of the nearest star to the guide star. Going down, the penultimate layer of the tree will be the distance of the farthest star from the guide star, and the final layer will contain the star ID of the guide star.

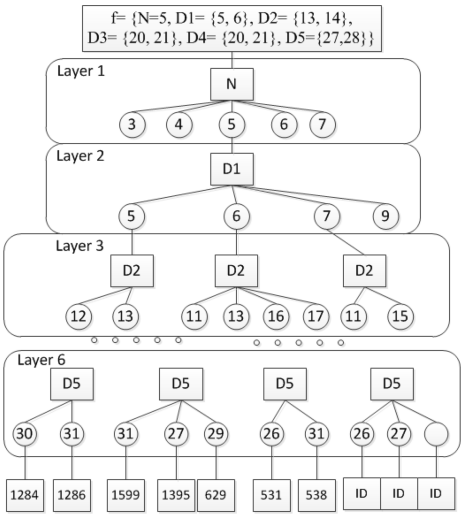


Figure 3.15: Tree SPD of STOD algorithm (adapted from [15]).

Once the image is acquired, the star nearest to the center of the image is selected as the guide star. The nearby stars which lie in the FOV of the guide star are counted, and their distances from the guide star are arranged in increasing order. With this information, the tree is traversed along the path which finds a match at each layer. The tree is traversed until we reach the final layer where the star ID is stored. Hence the guide star is identified as the star ID for which the traversal path is completed till the last layer is reached. This algorithm was developed specifically for increasing the speed of star identification.

Problems with star tree with optimized database: The most significant advantage of the STOD algorithm is the drastic increase in the speed of star pattern recognition by employing a parallel search on the hierarchical tree SPD. However, the top layer of the tree contains the number of neighboring stars as the feature for traversal. This feature makes the algorithm unreliable when there

are stars missed due to patch mismatch and magnitude uncertainty. As well as when there are false stars in the image, it will affect the top layer of the tree SPD. Thus, the algorithm has many chances of failure due to the sensitive top layer of the tree.

7. **Grid-based back propagation network** - Alveda, P. *et al.* in [34] initiated the neural network approach for solving the problem of reliable star identification. Later, the research community adopted to integrate the varieties of features with the neural network models. One of those interesting approaches was the back propagation (BP) combined with the grid algorithm [35]. The proposed neural network model for the recognition system is shown in Fig 3.16.

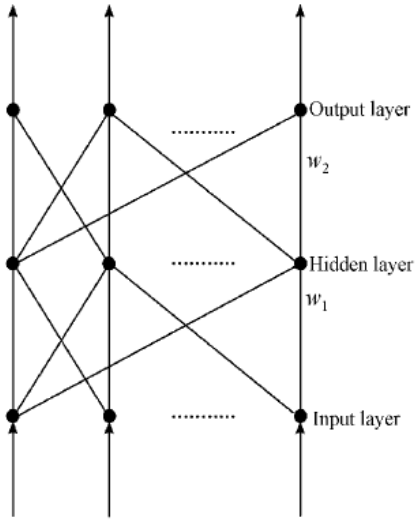


Figure 3.16: Grid based back propagation neural network.

The recognition system consists of a classifier and parallel subnetworks which finally give the output for the input star pattern from the image. For generating the SPD, a 13-bit binary pattern is developed for each guide star after applying the grid algorithm. For classifier, a BP network is adopted. Firstly, the classifier is trained by giving the input bit patterns of all the guide stars in the SPD. With this training, the classified input samples are trained to allocate the

weight matrix for the network. An optimal number of 30 hidden nodes are determined for the best performance of the network according to the authors of this technique.

Once the image is acquired, it is given as an input to the classifier, and it gets classified by one of the sub-neural networks depending upon the weight assigned to it. So, the input pattern from the image will pass through a specific path in the network. Through this approach, the input pattern from the image will be classified and recognized correctly.

Problems with the grid-based back propagation neural network: The back propagation based supervised learning approach faces the same drawbacks as that of the grid algorithm. The reason being the classifier is trained with the input bit-pattern developed after the implementation of the grid algorithm. However, a supervised learning method like back propagation can be efficiently used to solve the problems of missing stars and false stars. While training the classifier, the input bit-pattern should be generated by missing some stars and adding some false stars to the grid pattern of every guide star in the SPD. In this way, the weights of the network will be assigned accordingly to the problems present in the star images captured.

8. **Mixed star pattern with multilayer Self-Organizing Map (SOM) neural network** - The grid-based back propagation approach was a supervised training approach. However, the features extracted from the star image can also be learned through an unsupervised method. Many such unsupervised [36–38] approaches were also proposed in the past two decades. One of those unsupervised neural network approaches was based on the multi-layer and hierarchical SOM neural network [39]. Fig 3.17 shows a neural network SOM. A SOM network is an unsupervised learning technique which is based on a winner-take-all neuron strategy decided by the Euclidean distance parameter.

   A mixed star pattern consisting of the distances and the magnitude of the stars is developed as an input for the SOM network. Four nearest stars to the guide star are chosen for forming the mixed star pattern for the guide star. Instead of

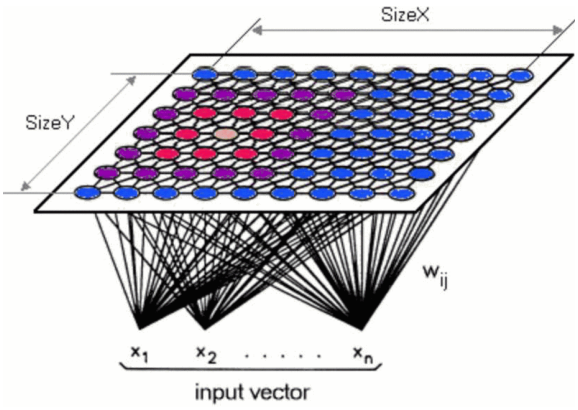


Figure 3.17: Neural network self-organizing-map (SOM).

adopting a conventional mono-layer SOM network, a multilayer tree structure SOM network is developed as shown in Fig 3.18.

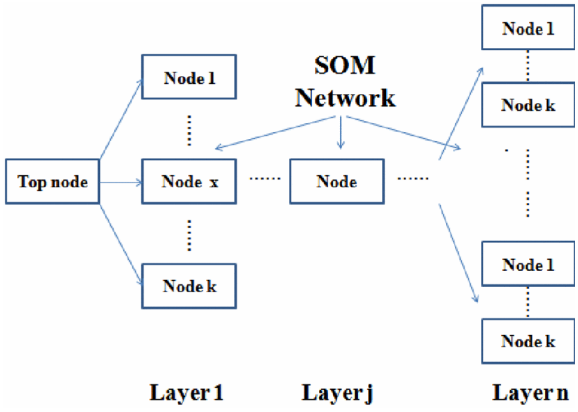


Figure 3.18: Multilayer SOM neural network (adapted from [39]).

There are n layers with each layer having k nodes in the structure. The mixed star pattern is given as an input to the top node of the network. The neuron nearest to the input will be chosen as a winner in each layer, and the same follows for each layer. Thus, the pattern is propagated through the network, and the weights are updated accordingly depending upon the path traversed in the network by the pattern. Hence the network weights are trained in an unsupervised manner.

Once the star tracker acquires the image, the star nearest to the center of the image is selected as the guide star. The four closest stars to the guide star are selected for forming the mixed star pattern as was used to train the network. This star pattern is now given as an input to the top node of the multilayer SOM neural network. This input pattern from the image will traverse along a path in the network to reach the final node and hence will fall into one of the paths of the star IDs in the SPD. The path of the input pattern which matches with the path of the star ID in the SPD is identified as the guide star in the image.

Problems with the multilayer SOM neural network approach: The multilayer SOM approach suffers from the drawback of utilizing the magnitude of the stars in the star pattern developed as an input to the network. The magnitude uncertainty of stars is a major problem faced by star pattern recognition techniques as was explored and analyzed earlier in this chapter. Because the star pattern is composed of radial distances along with the star magnitudes, the matching process is going to be unreliable. Apart from the star magnitude problem, the SOM only employs the nearest four stars to the guide star for extracting the star pattern. Thus, it will suffer from the same problem as that of Liebe and Pyramid algorithm. If a higher number of stars are considered for forming the star pattern, the weight matrix of the SOM network will be huge which will result in a low speed of recognition. Hence there is a trade-off between the number of stars selected for training the network (for improving the reliability) and the speed of star identification.

9. **Hidden Markov Model (HMM) based star identification algorithm** - A probabilistic graph approach based on discrete Hidden Markov Model (HMM) was proposed in [40] to solve the problem of star pattern recognition. The HMM models are widely used in the domain of speech recognition, language modeling, and handwriting recognition. The idea proposed by the HMM technique was to feed the angular features to generate a unique HMM-based star pattern for every star in the SPD. Fig 3.19 shows a two-layer HMM model proposed by the

authors. The authors characterized the HMM model by five elements, namely $\lambda = N, M, \pi, A, B$ where N is the number of hidden states in the model. The individual states are the angular features of the three nearest stars to the guide star in the SPD, i.e. θ_1, θ_2 ... θ_6. M corresponds to the number of distinct observation symbols per state. Matrix A is the state-transition probability distributions and matrix B contains the symbol probability distributions.

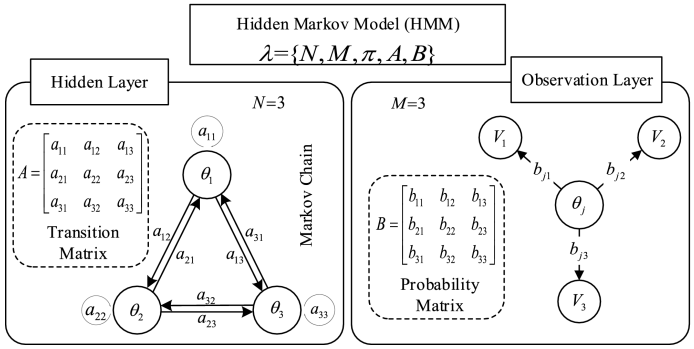


Figure 3.19: Schematic of the two-layer construction of an HMM [40].

Once the image is acquired, the HMM model is formed based on the features extracted by the pyramid algorithm. It is then matched with the models stored in the SPD. The one with the best match through all the layers is selected as the star ID to be assigned to the guide star in the image captured.

Problems with the HMM model based star pattern recognition: The HMM model suffers from the same problems as those dealt by the pyramid algorithm as the HMM considers the angular features extracted from the nearest three stars as the input to the model. If the number of stars and hence the number of features considered are increased, it may solve the problem of magnitude uncertainty and patch mismatch, however, the size of the HMM model will be substantial. This increase in the size will result into a vast SPD size which will further lead to an increase in the time complexity for star pattern recognition.

10. **Multi-poles algorithm** - A star pattern recognition technique specifically for dealing with the high presence of false stars was proposed in [41] which is

named as the Multi-poles algorithm. The basic idea of this algorithm was to select multiple guide stars instead of choosing only one guide star in the image. However, for generating the SPD only one guide star needs to be selected. For making the SPD, the radial distances of the neighboring stars lying in the FOV of the guide star are chosen (like the Geometric voting algorithm) as shown in Fig 3.20.

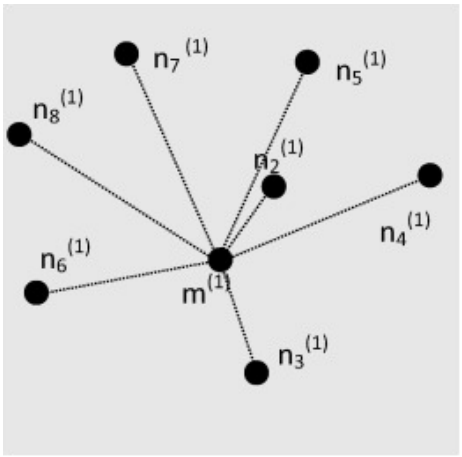


Figure 3.20: Pole star $m^{(1)}$ with associated neighbors $n_j^{(1)}$.

Once the image is acquired, firstly, the star nearest to the center of the image is selected as the pole star to be identified. The radial distances of the neighboring stars are extracted and compared with those stored in the SPD. For every match, the star in the image is also marked with the star ID of the neighboring star matched in the SPD. After finishing the above operation, the nearest star to the pole star in the image is selected as the new pole star. As this new pole star already has a star ID marked from the previous comparison, the features for this new pole star are checked with the features of the star ID assigned to this star. If all the features in the image match with those in the SPD, it is counted as the verification of the previous match and the process is stopped with reliably identifying the correct star ID for the guide star. This process is shown in Fig 3.21. If the features do not match, the next star in the image

is now selected as the pole star, and the process is repeated until we find a correct match in the SPD.

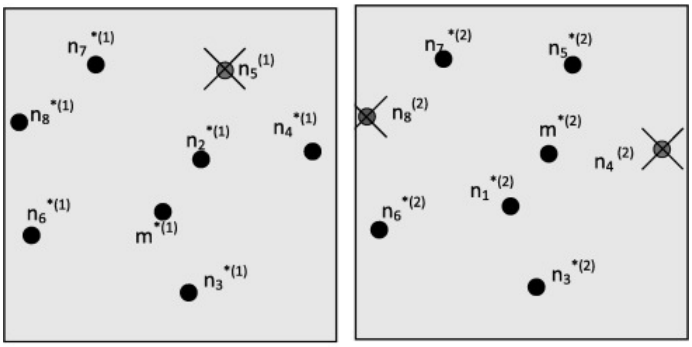


Figure 3.21: Accepted pole stars and the associated neighbor stars.

Problems with the Multi-poles algorithm: The multi-poles algorithm is probably the simplest and the most effective approach for achieving a reliable star identification. Though the features extracted from the neighboring stars are only the radial distances, the multi-poles algorithm will not suffer from the drawbacks of the Geometric voting algorithm. This is because it considers many guide stars (poles) for comparison. Apart from the different guide stars considered, a verification step is also employed for final confirmation of the correct star ID assigned to the guide star. However, the practical idea of considering many poles will lead to an increased amount of time complexity for achieving star pattern recognition. An optimized number of guide stars (poles) must be determined and fixed for comparison and verification purposes to obtain real-time star identification.

We have explained in detail some of the judiciously selected state-of-the-art SPR techniques. There have been many a plethora of SPR techniques proposed in the past two decades. Based on our study, the existing SPR techniques can be classified into the following four categories:

1. **Geometric approach** - The SPR techniques which utilize the radial distances, angular separations, area, etc. as the features for comparison and matching

are categorized as the geometric SPR techniques. The star pattern recognition techniques such as the Liebe algorithm, Pyramid algorithm, Planar triangle algorithm, Geometric voting algorithm fall under this category. The research community has modified this approach by applying different transforms on the geometric features. A log-polar transform utilizing both the distances and the angles was proposed by X. Wei *et al.* in [42]. T. Delabie *et al.* [43] proposed a star identification based on shortest distance transform. J. Li *at al.* [44] proposed to use an iterative approach utilizing the area, angles, and the geometric voting concept. G. Zhang proposed techniques [45,46] which utilized cyclic and radial features for comparison. A star identification based on Voronoi tessellation was proposed recently in [47]. Other SPR techniques which fall under this category are [48–55].

2. **Pattern based approach** - As mentioned earlier, the pattern-based approach, which was firstly proposed in grid algorithm, forms a coded pattern for comparison. Modifications of the grid algorithm were proposed in [56,57] where an optimized grid size was selected, and added verification steps were adopted for ruling out false choices. These two techniques have significantly improved the performance of the original grid algorithm. A pole star algorithm was proposed in [58] where a coded binary string was formed from the neighboring star positions for comparison. A similar approach was adopted in [59], which formed an integer value from the coded string for comparison. Other SPR techniques which fall under this category are [59–62].

3. **Neural network based approach** - The neural network approach deals with forming different network models (both supervised and unsupervised) utilizing the features of the star pattern for matching. The neural network model approach of solving the problem of star pattern recognition has also received significant attention, especially in the recent years with the advent of the robust machine learning techniques. Apart from the earlier described two neural network approaches for star identification, other SPR techniques which fall under this category are [63–65].

4. **Fast and low memory techniques** - Some SPR techniques were specifically developed for achieving a real-time star identification. The earlier described STOD algorithm falls under the same category. These SPR techniques propose novel ways of constructing the SPD which leads to a reduction in time of star identification. One such technique was proposed by D. Mortari, based on the k-vector construction and popularly known as a search-less SPR technique [66]. Another similar idea based on P-vector was proposed by Yang, J. in [67]. Fast access and low memory star pair catalog was proposed in [68]. Other SPR techniques which fall under this category are [69–75].

Apart from the above four general approaches to the problem of star pattern recognition, there have been some approaches [76–78] which utilize the images captured from two different star trackers mounted on the satellite for attitude determination. These techniques are popularly known as the *Multi-FOV* SPR techniques in the research community. Some techniques [79, 80] also utilize the magnitude of the stars for comparison and matching. However, as described earlier, the star magnitude is not a reliable characteristic feature which can be used for matching. The ambiguity in utilizing the brightness of star for comparison was also reported in [81, 82]. A comprehensive review of star identification approaches has been provided in survey papers [83–85], and a book of star identification [86] by G. Zhang. There have also been several doctoral thesis [87–92] based on the highly researched problem of star tracker.

3.3 Performance Evaluation of Star Pattern Recognition Techniques

Having reviewed the state-of-the-art star pattern recognition techniques, we will now mention the parameters based on which an SPR technique is evaluated. SPR techniques are usually assessed based on three parameters. These parameters are explained below in the order of their importance:

1. **Recognition reliability** - *Hitomi* spacecraft was lost due to the incorrect attitude information provided by the star tracker system. Thus, a reliable and accurate star tracker system is a must for the satellites. Hence recognition reliability forms the primary parameter for an SPR technique. Recognition reliability is also known as the identification accuracy of the SPR technique. This parameter corresponds to the measure of correctly identified star images on a set of tested images. The attitude determination techniques such as TRIAD [18] or QUEST [19] require two or three stars to be identified in the image for attitude calculation. Thus, a star image is said to be correctly identified if the SPR technique can identify any selected three stars correctly in the image captured. Even if one of the star is incorrectly identified, the whole identification is treated as false because the calculated output attitude will be false. The higher the recognition reliability, the better is the SPR technique.

2. **Time complexity** - A Low Earth Orbit (LEO) satellite in space usually revolves around the Earth at a speed of *7.8 km/s*. In such a situation, if a high recognition reliable SPR technique consumes *5 seconds* (say) for star identification, the satellite would have already traveled *39 km* in orbit after the results of star identification. Thus, even if an SPR technique is highly recognition reliable, the results of star identification will be meaningless if it cannot be implemented in real-time. In the real-time hardware implementation, the whole process of star tracking (starting from pre-processing until attitude output) is supposed to take less than *hundred milliseconds*. This time corresponds to an update rate of 10 Hz. Amongst all the steps of the star tracker, pre-processing and attitude calculation consume almost 70-80 *msec* on an average (more details in the hardware implementation). Thus, an SPR technique is supposed to provide star identification in less than 20 *msec* to meet the requirement of a real-time star tracking system. Hence an SPR technique is expected to have a low time complexity (or) high speed of recognition along with high recognition reliability.

3. **SPD size** - The SPD for an SPR technique has to be stored in the onboard

computer of the star tracker system. In the 1990s, the onboard computer systems did not have many resources for saving a massive amount of data. However, with the passage of time, and reduction in the size of CMOS transistors, the support of onboard storage has increased significantly especially in the past decade. This lack of storage capacity was the reason that SPD size was one of the major concerns for the SPR techniques dating back to 1990s. Currently, SPD size is still considered as a benchmarking parameter for the SPR techniques. However, the importance given to SPD size has been reduced over the years. Hence it has climbed down in the order of importance compared to the recognition reliability and time complexity of an SPR technique.

A star pattern recognition technique is evaluated based on the above described three parameters with primary importance given to recognition reliability and time complexity of an SPR technique.

3.4 Summary

In this chapter, firstly we provided a quantitative analysis of the problems faced in the process of star pattern recognition. With the studies performed on the real images captured by the star tracker, the gravity of each research problem is now defined clearly. Later, we provided a comprehensive review of the important star pattern recognition techniques. We selected the methods ranging from geometric approach, pattern-based approach to a neural network approach. However, from the review of these techniques, it is clear that recognition reliability and time complexity of the identification is still the primary trade-off for the star pattern recognition techniques. The existing state-of-the-art star pattern recognition techniques still suffer from this trade-off. Thus, we aim to develop an approach to star pattern recognition which is highly robust and give a high speed at the same time. In the next chapter, we will explain the proposed methods for star pattern recognition.

Chapter 4

Proposed Approaches

This chapter comprises the central theme of this thesis. In this chapter, we will explain in detail the developed approaches to achieve a reliable and fast star pattern recognition for an LIS mode star tracker. The algorithmic (software) part of the proposed methods will be explained in this chapter. We have developed three diverse approaches to solve the problem of star pattern recognition for an LIS mode star tracker. Firstly, we will propose a novel generalized framework for the process of star identification. Later, we will explain every developed approach based on the proposed framework in detail. Finally, we will also describe potential failure analysis of the proposed approaches.

4.1 Proposed Generalized Framework for Star Identification

As described earlier in Chapter 3 of this thesis, recognition reliability and speed (time complexity) are the two most important parameters of an SPR technique. For achieving high recognition reliability, more features are preferred to be extracted along with more combinations of star clusters and thus, more comparisons are to be made. However, such an exhaustive process will increase the time required for feature extraction and comparison with the SPD, thus reducing the speed of iden-

tification. On the other hand, if fewer features are used with only some selected combinations of the star cluster, then the number of comparisons to be made are reduced significantly, which increases the speed of identification. However, the accuracy of correct star identification might decrease because of a selected number of features used for extraction and comparison. Thus, both the evaluation parameters of star identification - recognition reliability and speed of recognition are a trade-off to each other.

While developing our approaches, we have kept in mind the trade-off between recognition reliability and time complexity. In our developed methods, we want to address this trade-off. Thus, we have propose a generalized framework for the developed approaches which enables to achieve high recognition reliability along with low time complexity. Fig 4.1 shows the proposed framework.

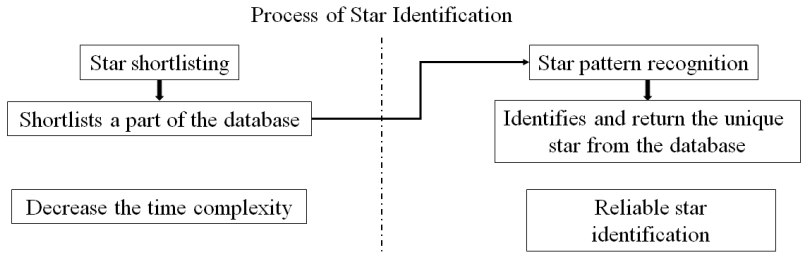


Figure 4.1: Proposed generalized framework for the developed approaches.

As can be seen in Fig 4.1, our framework is divided into two parts - star pattern recognition and star shortlisting. The process of star shortlisting is applied before the actual star pattern recognition. Star shortlisting, as the name suggests, shortlists a part of the SPD. The shortlisting of the SPD is done by the partial matching of the features extracted from the image with those stored in the SPD. Thus, we obtain a shortlisted list of star IDs which seem to be moderately present in the image. Later, this shortlisted list of star IDs is passed to the actual star pattern recognition technique, which is concerned with identifying the correct star ID from the shortlisted list. Thus, the star pattern recognition only needs to operate on a shortlisted list of star IDs (part of the SPD) rather than running on the whole SPD (conventional

way).

The primary purpose of the star shortlisting is to serve for decreasing the time complexity by providing a shortlisted list of star IDs. The SPR technique can then make use of the various combination of features for identifying the stars reliably in the image. Thus, the proposed framework will resolve the trade-off to a considerable extent (justified later with results). Hence the combination of star shortlisting and star pattern recognition technique allows us to achieve high recognition reliability and low time complexity.

The proposed framework places some conditions while developing the star shortlisting technique. The requirement of an efficient shortlisting method would be to have a fast and straightforward feature extraction and comparison process. Apart from being simple in implementation, a shortlisting technique must also make sure that the correct star ID is present in the shortlisted list of the star IDs. This process will enable the shortlisting technique to reduce the time complexity.

The disadvantage of the proposed framework comes in the form of increased SPD size. The conventional framework only needs to store the SPD for the SPR technique. However, for our proposed framework, we will also need to save the SPD prepared for the shortlisting routine. Hence the SPD size for the proposed framework is supposed to be larger compared to that of the conventional techniques. However, as explained earlier, storing up to 5 MB is not a problem for the current onboard computers. So, we need to keep in check that the SPD size of the combined shortlisting and the SPR technique does not exceed more than 5 MB. We will now explain each of the developed approaches in detail.

4.2 Worst-case Patch Mismatch based Shortlisting and Running Sequential Angular based Star Pattern Recognition

We will explain the first approach developed for solving the problem of star pattern recognition. This approach can be regarded as a combination of geometric and

pattern based point of view. In our first approach, we propose a novel combination of star shortlisting and pattern recognition algorithm which excels at both the recognition reliability and the speed of star identification. The first part of the proposed unification deals with shortlisting the stars based on worst-case patch mismatch, and the second part of the algorithm comprises of the pattern recognition technique based on an initial match confirmed by a running a sequential angular match. The former leads to the reduction in time for recognition and the latter leads to attaining high recognition reliability.

This approach is based on the identification of the center star and does not rely on the ambiguous brightness of the stars. Our algorithm utilizes all the stars which lie in the FOV for feature extraction. However, we apply a shortlisting method for selecting only a part of the SPD before actually performing the pattern recognition. The pattern recognition process further comprises of an initial match followed by running sequential angular match. Fig 4.2 shows the overall flow of the proposed algorithm.

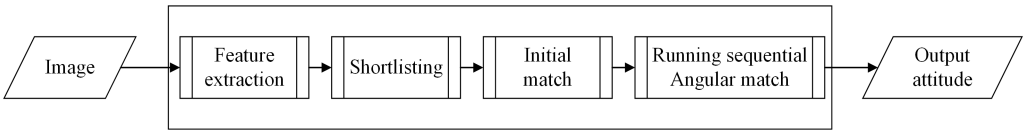


Figure 4.2: Proposed algorithm flow for the first approach.

We will introduce the feature extraction, construction of the look-up-table (LUT) based on worst-case patch mismatch, and the star pattern database generation for the proposed approach in section 4.2.1. In section 4.2.1, the shortlisting technique and the star pattern recognition technique are explained in detail. Finally, it is shown how the combination of shortlisting and pattern recognition technique is used for obtaining high recognition reliability as well as the high speed of identification.

4.2.1 Feature Extraction, Look-Up-Table and Star Pattern Database Construction

Feature Extraction

The proposed technique uses two rotation invariant features - radial distances from the center star and the relative angle that the consecutive stars make with the center star, as shown in Fig 4.3. Every star in SAO J2000 catalog [12] is considered at the center of the FOV, and all the stars which lie within the FOV of the center are taken into account in an increasing anti-clockwise angle which forms a pattern vector as shown in Fig 4.3. The radial distance of each neighboring star from the center star and the angle subtended by the consecutive neighboring stars with the center star are the features to be extracted. These are given by Eq 4.1 and Eq 4.2 respectively.

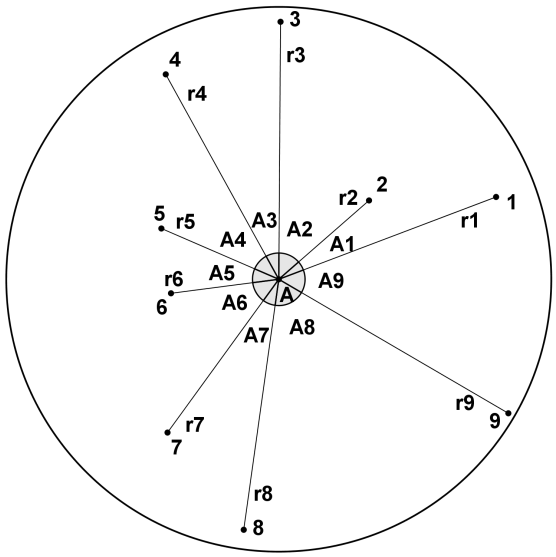


Figure 4.3: Feature extraction for the first approach.

$$r_n = \sqrt{(x_n - x_c)^2 + (y_n - y_c)^2} \tag{4.1}$$

$$\begin{aligned} A_n = \tan^{-1}\left(\frac{y_{n+1} - y_c}{x_{n+1} - x_c}\right) - \tan^{-1}\left(\frac{y_n - y_c}{x_n - x_c}\right); n \neq N \\ A_n = 360 - (A_1 + A_2 ... + A_{N-1}); n = N \end{aligned} \tag{4.2}$$

where x_n and y_n are the pixel co-ordinates of the n^{th} neighboring star in the image, N is the total number of neighboring stars in the FOV of the reference star, and x_c and y_c are the co-ordinates of the reference star A. For the purpose of shortlisting the star entries, only radial distances $\{r_1, r_2 ... r_n\}$ are utilized whereas, both the radial distances and relative angles are used for the generation of SPD, which is explained later in this section.

Worst-case Patch Mismatch and LUT Construction

- **Worst case patch mismatch calculation** - As mentioned earlier in Chapter 3, the major problem that the star pattern recognition techniques face is the patch mismatch between the SPD and the image captured. So, the idea of the shortlisting technique is first to find the worst case patch mismatch that can occur for a center star. We understood that the patch mismatch occurs because of the displacement and the rotation of the image captured when compared to the SPD. By worst case, we mean maximum difference (d_{max}) that can occur in the number of stars between the SPD and the image captured solely because of the patch mismatch. For calculating the d_{max}, a unique simulation approach is adopted which is explained in Fig 4.4.

  For every center star ID, the following steps are performed in a sequence:

  1. Every star which lies in the FOV of the center star (star ID selected) and the center star itself is shifted by steps of 1 pixel in the x-direction as

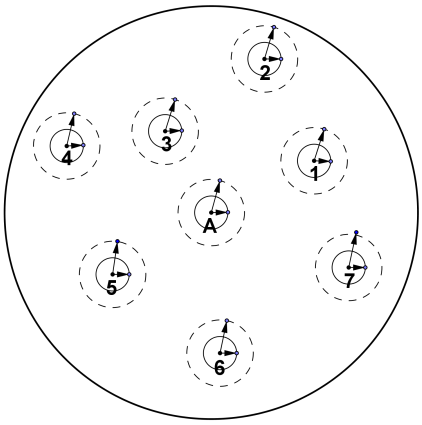


Figure 4.4: Worst case patch mismatch simulation.

shown in Fig 4.4.

2. Every star, after shifting, is rotated by levels of 1 degree in the counter-clockwise direction.
3. The distance of each star from the center of the image is calculated.
4. If the star nearest to the center of the image is still the same as the star ID (star A, in the example in Fig 4.4) for which this operation is performed, then step 2, step 3, and step 4 is repeated. These steps are repeated until the nearest star to the center of the image is different from the center star ID or when the degree of rotation reaches 360^0.
5. If the degree of rotation reaches 360^0, then we shift to step 1, and the whole process is repeated until the point when the star nearest to the center of the image is no longer the same center star ID (star A in Fig 4.4).

When the star nearest to the center of the image changes, the maximum difference occurs between the number of stars in the SPD of the center star ID and the number of stars currently in the image. For example, in Fig 4.4, if any of the neighboring stars 1 to 7, during the above simulation process, become the closest to the center of the image, then the worst case patch mismatch has

occurred for star A. When the above situation occurs, the d_{max} for star A is calculated by Eq 4.3.

$$d_{max} = N_{SPD} - N_{image} \tag{4.3}$$

where N_{SPD} is the initial number of stars in the SPD of the center star (7, in the example in Fig 4.4) before the operation of worst-case patch mismatch began, and N_{Image} is the number of stars which lie within the FOV of the center of the image when the worst-case patch mismatch situation has reached. The quantity d_{max} is calculated for every star having a magnitude threshold (M_v) of less than 6 from the SAO J2000 star catalog. This whole process of calculating the worst case patch mismatch is explained in detail in the flowchart in Fig 4.5.

- **LUT construction based on worst case patch mismatch** - We only use the radial distances of the neighboring stars for constructing the LUT. The LUT consists of four threshold levels for each center star, which are calculated as follows:

  1. Thresh 1: This threshold level represents the scenario when none of the stars miss in the FOV of the center star. Thus, without losing any stars in the FOV of the center star, the summation of the radial distances of all the neighboring stars (N_{SPD}) from the center star is calculated for this threshold level. Eq 4.4 gives this threshold level. It should be noted that this is the maximum value of all the threshold levels, as it considers all the neighboring stars.

$$Thresh1 = \sum_{i=1}^{N_{SPD}} r_i \tag{4.4}$$

  2. Thresh 2: This threshold level is calculated to consider the stars missed

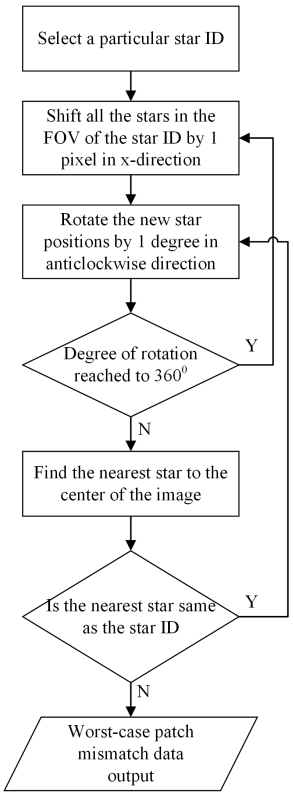


Figure 4.5: Process of worst case patch mismatch simulation.

due to the patch mismatch. It is essential to understand from the process of worst-case patch mismatch simulation that the neighboring stars which are *farther* away from the reference star have a *high probability* of been missed compared to that of the stars which are nearer to the reference star. The above phenomenon is because of the displacement of the image captured when compared to the SPD.

As mentioned earlier, the maximum number of stars that can miss due to patch mismatch is d_{max}. Thus, for calculating this threshold level, we miss d_{max} number of neighboring stars starting from the *farthest star* from the center star. The summation of radial distance is calculated after missing those *farthest* d_{max} stars. Eq 4.5 gives this threshold level. Having calculated Thresh 1 and Thresh 2, it should be clear that the summation

of radial distances of the neighboring stars in the image captured (having *only* patch mismatch) will fall between Thresh 1 and Thresh 2 of the reference star ID in the LUT.

$$Thresh2 = \sum_{i=1}^{N_{SPD}-d_{max}} r_i \tag{4.5}$$

3. Thresh 3: The levels $Thresh3$ and $Thresh4$ are calculated for considering the low magnitude missing stars in the image. As described earlier in chapter 3, magnitude uncertainty adds more missing stars to the problem of patch mismatch. We already considered the *farther* away missing stars due to patch mismatch by calculating the $Thresh2$ level. The *nearest* stars to the reference star have a less chance of getting missed due to patch mismatch. However, they may get missed because of their low magnitude. Thus, we need to consider these low-magnitude nearest stars which may get missed in addition to the missed stars due to patch mismatch.

   The existing approaches such as Pyramid [26] , k-nearest neighbors [47] rely on the nearest four stars for star pattern recognition. Thus, we select four stars as a worst case of the number of *nearest* stars missed due to magnitude uncertainty. For the sake of not making the window of qualification between the threshold levels big, we split the four low magnitude stars into two halves. Thus, for calculating $Thresh3$, we will only consider missing two more stars in addition to the d_{max} stars missed for $Thresh2$. Hence the *farthest* $(d_{max} + 2)$ stars are missed for calculating this threshold level, and the summation of the radial distances of the remaining stars from the center star is calculated. Eq 4.6 gives this threshold level. Having calculated $Thresh3$, it should be clear that the summation of the radial distances of the neighboring stars in the image captured (containing missed stars due to patch mismatch and less than three missed stars due to magnitude uncertainty) will fall between $Thresh1$ and $Thresh3$.

$$Thresh3 = \sum_{i=1}^{N_{SPD}-d_{max}-2} r_i \tag{4.6}$$

4. Thresh 4: As explained earlier while calculating the *Thresh*3 level, the consideration of low magnitude missing stars is divided into two halves. This level is the final level of all the thresholds and considers the second half of the low magnitude missing stars. Thus, for calculating this level, the *farthest* (d_{max} + 4) stars are missed, and summation of the radial distances of the remaining stars from the center star is calculated. Eq 4.7 gives this threshold level.

$$Thresh4 = \sum_{i=1}^{N_{SPD}-d_{max}-4} r_i \tag{4.7}$$

From the above calculation of the threshold levels, it should be noted that for each star ID, *Thresh*1 > *Thresh*2 > *Thresh*3 > *Thresh*4.

A Look-Up-Table (LUT) as shown in Fig 4.6 is then prepared offline with all the four thresholds calculated for each star entry as mentioned earlier. We consider the star entries which have a M_v of less than six from the SAO J2000 star catalog for constructing the LUT. The number of such stars is 4956. For a particular center star, a circular field of view (FOV) of 15^0 is considered for the above operations. There is one more parameter in the LUT, called the check bit, which is initially 0 for every star ID. The purpose of the check bit is explained later in section 4.2.2

Star Pattern Database (SPD) Generation

The SPD is first prepared offline and loaded into the star tracker onboard memory along with the LUT. For constructing the SPD, the same parameters of M_v and FOV

are adopted as considered for constructing the LUT. We will now describe the SPD construction for star A in Fig 4.3.

Referring back to Fig 4.3, we have nine neighboring stars to star A at the center. Hence, the feature vector for star A is constituted as a sequence - $\{r_1,\ A_1,\ r_2,\ A_2,\ r_3,\ A_3 ...\ r_9,\ A_9\}$. If a star at the center has n number of neighboring stars within its FOV, its feature vector would look like $r_1,\ A_1,\ r_2,\ A_2,\ r_3,\ A_3 ... r_n,\ A_n$, where the features r_n and A_n are calculated as was described in section 4.2.1. The LUT and SPD constructed are shown in Fig 4.6.

Look-Up-Table

Index	Check bit	Thresh 1	Thresh 2	Thresh 3	Thresh 4
0	0	4563	2696	1866	1176
1	0	3580	2331	1551	849
2	0	4596	1742	1056	499
⋮	⋮	⋮	⋮	⋮	⋮
4955	0	4637	3263	2435	1729

Star Pattern Database

$\{r_1, A_1\}$	$\{r_2, A_2\}$	...	$\{r_N, A_N\}$
{103.7, 51.3}	{434.8, 19.4}	...	{268.5, 6.27}
{378.1, 39.2}	{347.7, 4.7}	...	{165.5, 73.9}
{351.9, 20.6}	{485.3, 45.6}	...	{347.6, 4.7}
⋮	⋮	⋮	⋮
{400.2, 9.9}	{306.0, 3.2}	...	{423.9, 6.1}

Figure 4.6: Look-Up-Table and Star Pattern Database for the first approach.

4.2.2 Shortlisting and Star pattern recognition

Shortlisting

The process of shortlisting using the LUT is explained as shown in Fig 4.7. As can be seen in Fig 4.7, there are three slabs allotted to the LUT. Slab 1 is between *Thresh*1 and *Thresh*2, Slab 2 is between *Thresh*1 and *Thresh*3, and Slab 3 is between *Thresh*1 and *Thresh*4. Once the image is acquired, the summation of the radial distances of all the neighboring stars which lie in the FOV of the star nearest to the center of the image (reference star) is calculated by Eq 4.8, which is used for shortlisting the star entries.

$$R^I_{sum} = \sum_{i=1}^{n} r_i \tag{4.8}$$

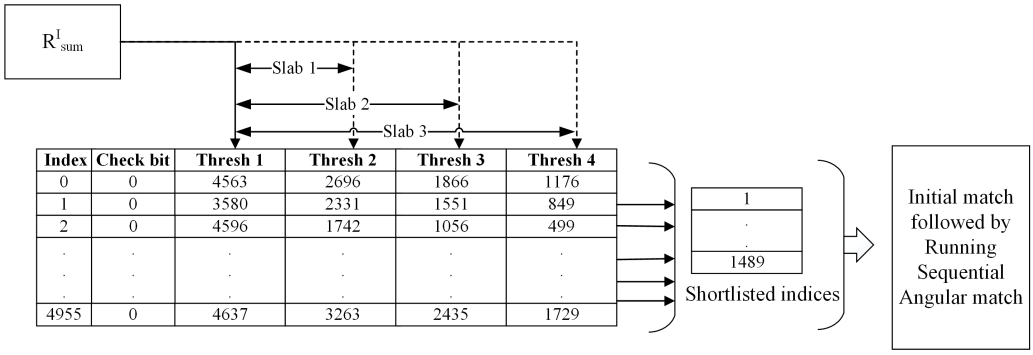


Figure 4.7: Shortlisting technique for the first approach.

Once the R^I_{sum} is calculated, firstly it is checked whether it falls in Slab 1 for each star ID,i.e. it should fall between $Thresh1$ and $Thresh2$, satisfying the following condition:

$$Thresh2 < R^I_{sum} < Thresh1 \tag{4.9}$$

Only those star IDs are shortlisted for which the above condition is correct, and the check bit of those star IDs is *toggled* from 0 to 1. Later only those shortlisted star IDs are checked for the star pattern recognition process described later in this section. In the case that the correct star ID does not get shortlisted in Slab 1, the window is increased and is checked for Slab 2 and later if required for Slab 3. The condition for increasing the window size and shifting to Slab 2 and Slab 3 is dependent upon the result of the star pattern recognition technique. This combination of the shortlisting technique and the star pattern recognition technique is explained later.

Star Pattern Recognition

- **Initial match** - We will explain our proposed pattern recognition technique for identification of correct star and how it also tackles the problems of mismatch between the SPD and the image, magnitude uncertainty, and false stars concurrently with the help of an example in Fig 4.8. The example in Fig 4.8

depicts two circles - the solid circle is the pattern vector of the star A, which is the same as was shown in Fig 4.3 and the dashed circle is the image captured by the star tracker. It can be noted from Fig 4.8 that stars 1, 3, 4 and 9 are not a part of the image captured while they are present in the SPD of star A. Also, the star F (noise) is present in the image but not a part of the SPD. Our proposed technique is to utilize both the radial and angular features of the pattern extracted for finding an initial match in the SPD. In Fig 4.8, it can be clearly seen that the radial distances r_2, r_5, r_6, r_7, and r_8 in the image will match exactly with the radial distances stored in the SPD of star A. We can also note that the angles A_5, A_6, and A_7 in the image will match exactly with the angles contained in the SPD of star A. But the angles A_1, A_2, A_3, A_4, A_{F1}, and A_{F2} in the image will not get matched with the angles stored in the SPD of star A. For the explanation of the initial match step of our technique, let us assume that the angle A_5 is the smallest angle in the image.

Then the initial match will be checked by finding the equivalent of the smallest angle in the image (A_5 in this example) with the angle present in the SPD of the star ID. In this example, A_5 will get matched with the angle A_5 in the SPD of star A. This angle match is then confirmed by finding the match of the radial distances of the stars containing that angle. In this case, it will be the star 5 and star 6 and their corresponding radial distances are r_5 and r_6, which will be matched with the radial distances contained in the SPD of the star A. Thus, only those star IDs are said to have an initial match which confirms both the angular match confirmed by radial distance match.

- **Running sequential angular match** - The first step of our proposed technique is to find the initial match. Later, to confirm that the initial match a particular star ID is indeed correct, a running sequential angular match is performed. The running sequential angular match starts from the angle after the initial match. For example, in Fig 4.8, after the initial match of angle A_5, angle A_6 and angle A_7 will get correctly matched with the angles stored in the SPD of star A. However, when it comes to the angle between the star 8 and star F, it

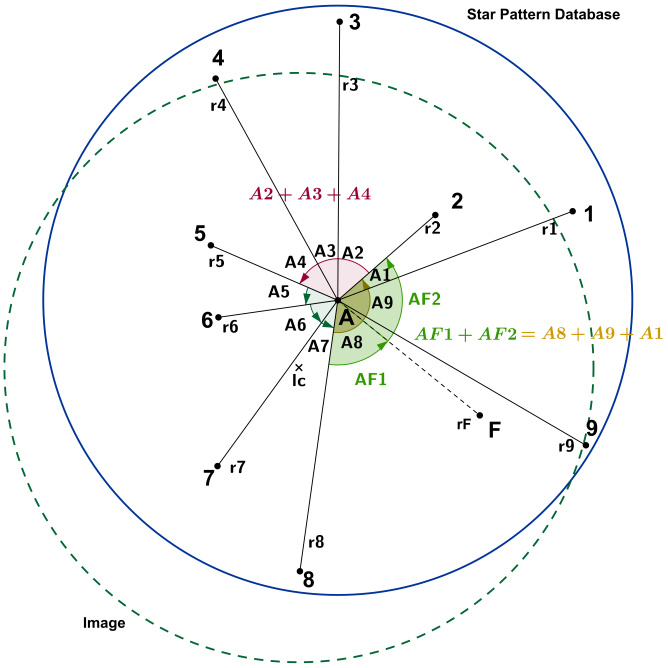


Figure 4.8: First example to explain the developed approach 1.

does not get a match with the SPD sequence of star A. Whenever the angle in the sequence does not match, our technique shifts to check the radial distance match in the sequence. In this example, $r8$ will find the match. However, r_F will not find the match. So, the next radial distance r_2 in the sequence is checked, which will find the match in the SPD of star A. After any two radial matches in the sequence are confirmed, the angle contained in between those matches is compared. In the image, the angle between star 8 and star 2 is the summation of A_{F1} and A_{F2}. In the SPD, the summation of the angles A_8, A_9, and A_1 is the angle between star 8 and star 2. As can be seen from Fig 4.8, this summation of the angles in the image matches with that of the SPD. Thus, the radial match is confirmed by this angular match. For any other star ID, when this process does not find a match anywhere in the sequence, it will break from checking further sequence of the angles and distances. Later, for this example, when we focus on the angle between star 2 and star 5, it is the summation of

angles A_2, A_3, and A_4. After applying the same process of matching the radial distances first and confirming with the angle summation, angle in the image between star 2 and star 5 will get matched with that in the SPD of star A. We can note that for this star A, we started from angle A_5 as the initial match and are back to angle A_5 after the running sequential angular match. Thus, a perfect sequential angular match is obtained for a single cycle of 360 degrees for the star A. Hence the star A is correctly identified as the center star even if there are missing stars due to patch mismatch and false stars in the image.

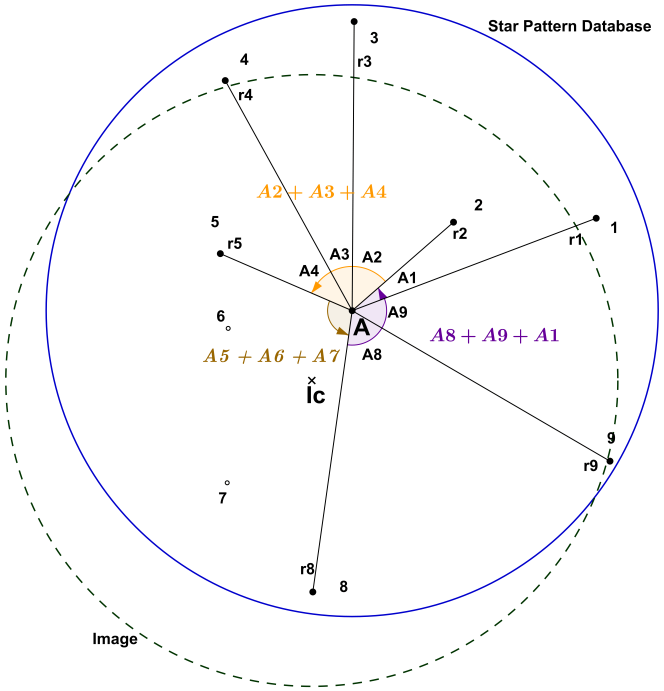


Figure 4.9: Second example involving brightness ambiguous missing stars.

In Fig 4.9, we show another example when the stars are missed because of magnitude uncertainty. In this example, let us consider that the stars 6 and 7 in Fig 4.8 are also missed in the captured image by the star tracker due to their low brightness and no false star is present in the image for this case. Thus, the captured image is the one shown in Fig 4.9 which contains only stars 2, 5 and

8 as the neighboring stars to star A.

For the explanation of the working of our technique in the example in Fig 4.9, let us assume that the angle between the star 5 and star 8 is the smallest angle in the image. Starting the initial match of our proposed technique, star 5 and star 8 will get initially matched for radial distances which will be found in the pattern vector of star A in the SPD. Their corresponding sequential angular summation in the SPD is $A_5 + A_6 + A_7$, which is the correct angle match between star 5 and star 8 in the image. Thus, the initial match is fixed in the SPD of star A.

The latter process of the running sequential angular match technique will remain the same as was described for the scenario in Fig 4.8. Thus, star A will be correctly identified in Fig 4.9 even if there are low magnitude missing stars. Hence according to the proposed star pattern recognition technique, a star will be correctly identified if and only if it satisfies both the initial match and the running sequential angular match for the single cycle of 360 degrees.

Combination of Shortlisting and Pattern Recognition Technique

The pattern recognition technique, consisting of the initial match and the running sequential angular match is applied only to the star IDs that get shortlisted. The shortlisted star IDs are the ones that lie under the Slab 1 of the LUT constructed from the concept of the worst case patch mismatch as was explained in section 4.2.1. However, there might arise a situation in which the correct star ID gets failed to be shortlisted by the technique, i.e. it does not fall under Slab 1 of the LUT. This situation will arise when there is a high brightness ambiguity, which will add to the already existing problem of patch mismatch. This problem is taken care by the proposed combination of shortlisting and pattern recognition technique. It is important to note that the uniqueness of the proposed pattern recognition technique lies in the perfect single cycle match of the image; otherwise, a match is not found. So, in this case, when the correct star ID is not shortlisted, none of the shortlisted star IDs will qualify for the perfect match by the pattern recognition technique.

Thus, the shortlisting method will have to increase the window for the qualification of shortlisting. This process of extending the window and thus shifting to Slab 2 and later if required to Slab 3 is explained in Fig 4.10.

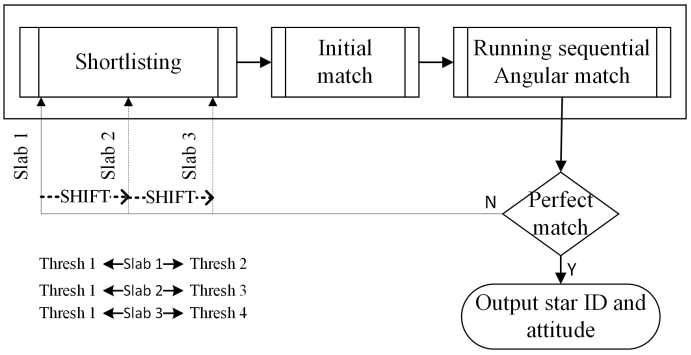


Figure 4.10: Final algorithm flow - Combination of shortlisting and pattern recognition technique.

If the pattern recognition technique is not able to find the perfect match, the summation of radial distances in the image (R^I_{sum}) that was calculated is now checked if it falls under Slab 2, i.e. between the $Thresh1$ and $Thresh3$ as was shown in Fig 4.7. Only those star IDs are checked whose check bit is 0 because the star IDs for whom the check bit is 1 are already tested for the perfect match. If R^I_{sum} falls under Slab 2 for that star ID, it is shortlisted, and the check bit for that star ID is toggled from 0 to 1. The above process will create a new shortlisted star IDs list that will further go through the same process of the pattern recognition technique. If still after checking for Slab 2, the perfect match is not found, it is tried later for Slab 3, keeping in mind to shortlist only those star IDs for which the check bit is still 0. Eventually, if the perfect match is still not found after checking for Slab 3, all the remaining star IDs will be shortlisted and tested by the pattern recognition technique.

Whenever a perfect match is found for the image, the corresponding star ID in the SPD is allotted to the reference star in the image. Later, the neighboring stars to the reference star in the image can be easily identified from the sequence of the SPD of the reference star ID. Two to three correctly identified stars in the image are

enough for attitude calculation, which can be accomplished by the existing QUEST method [19].

4.3 Image to Signal Approach

In this section, we will explain the second approach, which is a novel combination of hamming distance and spearman-correlation to achieve a reliable as well as a fast star identification. The proposed idea is to construct a discrete signal from the features extracted from the star image. We present a generalized framework for the process of feature extraction and construction of Look-Up-Table (LUT) and SPD for the proposed technique. Later, with the help of an illustrative example, we describe the process of using hamming distance for shortlisting the star IDs and spearman correlation for identifying the correct star ID from the shortlisted list. The hamming distance approach leads to fast recognition, and the spearman correlation helps in achieving high recognition reliability. Fig 4.11 shows the overall flow of this proposed approach. The subsequent sections describe the details of the proposed method.

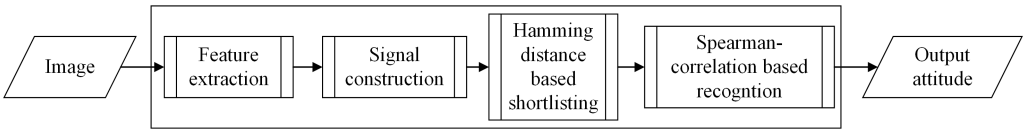


Figure 4.11: Proposed algorithm flow for the second approach.

4.3.1 Feature Extraction, and Signal, Look-Up-Table and Star Pattern Database Construction

Feature Extraction

The proposed technique uses two rotation and translation invariant features - euclidean distances from the center star and the relative angle that the consecutive neighboring stars make with the center star, as shown in Fig 4.12. We use the

star catalog SAO J2000 and every star ID which has a relative magnitude M_v less than 6.0 is considered for forming the LUT and the SPD for the proposed star recognition technique. Each star ID is considered at the center of the FOV, and all the neighboring stars which lie within the pattern radius (the number of pixels corresponding to the FOV) from the center are considered for extracting the two features mentioned above. Eq 4.10 gives the relation between the pattern radius (PR) and the FOV. Both the extracted features are calculated by Eq 4.11 and Eq 4.12 respectively.

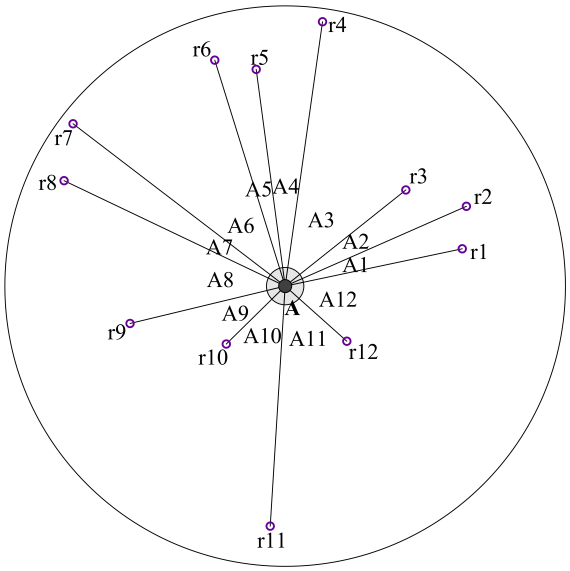


Figure 4.12: Feature extraction for the second approach.

$$FOV = 2\tan^{-1}\left(\frac{PR}{2\,(f/\rho)}\right) \tag{4.10}$$

where FOV is the field of the view of the star tracker, f is the focal length, ρ is the pixel dimension, and PR is the maximum distance in terms of pixels from the center.

$$r_n = \sqrt{(x_n - w/2)^2 + (y_n - h/2)^2} \tag{4.11}$$

$$\begin{aligned} A_n = \tan^{-1}\left(\frac{y_{n+1} - h/2}{x_{n+1} - w/2}\right) - \tan^{-1}\left(\frac{y_n - h/2}{x_n - w/2}\right); n \neq N \\ A_n = 360 - (A_1 + A_2 ... + A_{N-1}); n = N \end{aligned} \tag{4.12}$$

where x_n and y_n are the pixel co-ordinates of the star in the image, N is the total number of neighboring stars, and w and h are the dimensions of the image sensor.

Signal, LUT and SPD Construction

- **Signal construction** - As explained in Fig 4.12, there are two features extracted - relative angles and euclidean distances. We propose that the relative angles extracted can be visualized as time domain and the corresponding radial distances can be visualized as the amplitude of the signal. For example, in Fig 4.12, we have twelve stars surrounding the reference star A. Thus, there are twelve radial distances and relative angles, namely - $\{r_1, A_1, r_2, A_2 ... r_{12}, A_{12}\}$. Every radial distance r_n corresponds to its relative angle A_n i.e. r_1 corresponds to A_1, r_2 corresponds to A_2 and similarly r_{12} corresponds to A_{12}. Thus, we construct a signal containing data points as shown in Fig 4.13, which depicts the star pattern of the reference star A. As shown in Fig 4.13, the signal has relative angles (A_n) in increasing order of their value on the x-axis (visualized as time domain) and their corresponding radial distance value (r_n, visualized as amplitude). The point Sn in Fig 4.13 corresponds to the star n, with its coordinates being (A_n, r_n). For example, A_5, the smallest angle in Fig 4.12 and its corresponding amplitude r_5 is represented as the point S5 in Fig 4.13. The maximum relative angle between two stars in the image can be 180^0 or π radians. The maximum radial distance between any of the neighboring star from the center star can be PR pixels. This constructed signal is now utilized

for LUT and SPD preparation as explained later in this section.

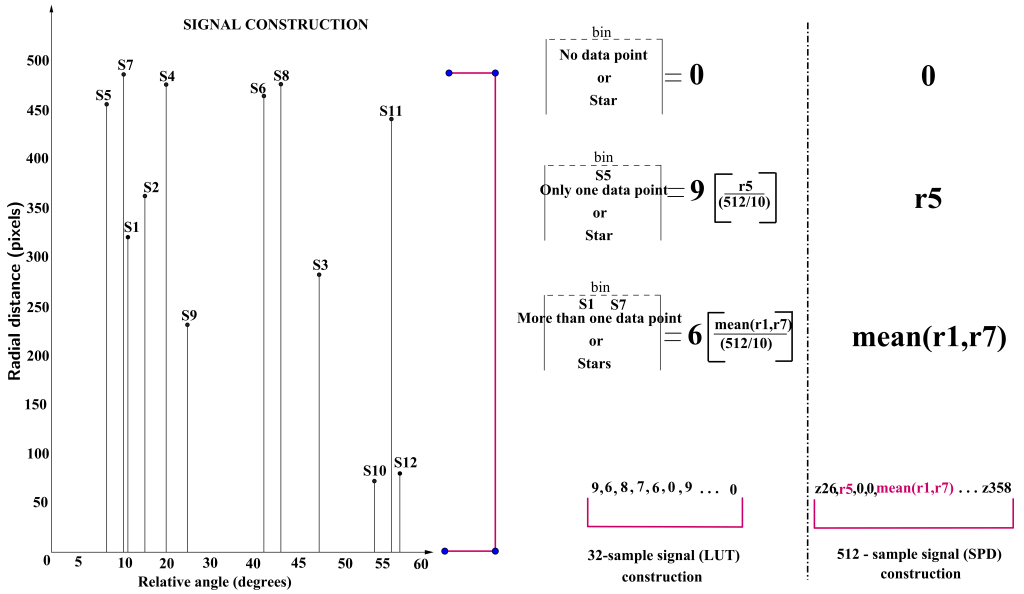


Figure 4.13: Signal, LUT and SPD construction from the extracted features.

- **Look-Up-Table (LUT) construction** - The purpose of the LUT construction is for shortlisting the star IDs which seem to be present in the image. Thus, LUT will contain a coded signal for every star ID. The LUT can be constructed as an N_{LUT} sample signal for every star ID, where N_{LUT} is given by Eq 4.13. The entire signal is then broken into N_{LUT} samples (bins). At the same time, the amplitude will also fall in either of slabs from 0 to S (total number of slabs are S+1). The division of the amplitude into slabs corresponds to describing the radial distance of the particular data point when compared to the maximum distance (pattern radius). Thus, S is calculated by Eq 4.14. Hence, we divide the signal into N_{LUT} bins and each bin can take a value from 0 to S.

$$N_{LUT} = 2^k; k = 1, 2, 3, \ldots \tag{4.13}$$

$$S = round\left(\log_2 PR\right) \tag{4.14}$$

Depending upon A_n, every data point of the signal will fall into a certain bin number given by Eq 4.15 and the amplitude of that data (r_n) will fall into one of the S+1 slabs given by Eq 4.16.

$$bin_{LUT} = round\left(\frac{Angle^0}{(180^0/N_{LUT})}\right) \tag{4.15}$$

$$slab_{LUT} = round\left(\frac{Radialdistance}{(PR/(S+1))}\right) \tag{4.16}$$

For visualization and better explanation, we will describe the construction of the LUT for the parameters of star tracker SST-20S [9] (in Table 3.1). For the parameters of the star tracker SST-20S, PR is calculated to have a value of 512 pixels and the corresponding S calculated is 9. Thus, the total number of slabs are ten i.e., 0 to 9. We have selected the value of k as 5 in Eq 4.13, which gives us the value of N_{LUT} as 32. The reason for this selection is explained later in section 4.3.3. For now, we will explain the construction of LUT with the values calculated above for the reference star A (in Fig 4.12).

If there is no data point or a star in a specific bin, it is allotted a $slab_{LUT}$ value of 0, as shown in Fig 4.13. If there is a single data point in a bin, it is assigned the $slab_{LUT}$ calculated by Eq 4.16. For example in Fig 4.13, only one point S5 lies in a particular bin. Thus, the $slab_{LUT}$ value of point S5 calculated by Eq 4.16 is equal to 9. If there is more than one point in a particular bin, the mean of their amplitude values (r_n) is considered for calculation. For example in Fig 4.13, the $slab_{LUT}$ value calculated for two points S1 and S7 falling in the same bin is 6. Hence, using Eq 4.15 and Eq 4.16, a 32 sample signal - {9,

6, 8, 7...0} as shown in Fig 4.13 is prepared for the pattern of reference star A. Similarly, the 32 sample signal for each star ID is constructed for preparing the LUT.

- **Star pattern database (SPD) construction** - The SPD is also constructed as a N_{SPD} sample signal for every star ID, where N_{SPD} is calculated by Eq 4.17. The sample signal constructed for the SPD is expected to have a higher number of samples because this signal will be used for identifying the correct star from the shortlisted entries. Thus, the value of N_{SPD} must be high compared to that of N_{LUT}. For the parameters of the star tracker SST-20S, we have selected the value of k as 9 in Eq 4.17, and thus the N_{SPD} calculated is 512. The reason behind this selection is explained later in section 4.3.3. For now, we will explain the construction of the SPD for the reference star A (in Fig 4.12) with N_{SPD} as 512.

$$N_{SPD} = 2^k; k = 1, 2, 3, \ldots \tag{4.17}$$

For the SPD, the signal is broken into 512 samples (bins) and the amplitude remains the same as the radial distance of the data point. For a particular bin, if there is no data point, it is assigned a value of zero; if there is a single data point, it is assigned the value of the radial distance of that data point; if there is more than one data point, it is assigned a value equal to the average of the data points, all the three cases being shown in Fig 4.13. The bin number for a particular data point for the SPD is given by Eq 4.18.

$$bin_{SPD} = round\left(\frac{Angle^0}{(180^0/N_{SPD})}\right) \tag{4.18}$$

As there are 512 bins, it is obvious that there will be many bins which will have zero value in between two data points. So, if there are more than three zeroes

between two bins, they are represented by *z_number_of_zeroes*. Hence, the 512 sample signal for the pattern of reference star A (in Fig 4.12) is prepared as - $\{z_{26}, r_5, 0, 0, \text{mean}(r_1,r_7)...z_{358}\}$ as shown in Fig 4.13. Similarly, the 512 sample signal is constructed for every star ID, and the SPD is prepared.

As explained earlier, the 32-sample signal and the 512-sample signal is constructed for every star ID. The LUT and SPD prepared for the proposed technique are shown in Fig 4.14. The reason for selecting 32-sample for LUT and 512-sample for SPD is explained later in section 4.3.3.

Look-Up-Table

Index	32 sample signal						
0	6	6	4	9	...	0	0
1	7	8	2	5	...	0	0
2	6	6	2	5	...	0	0
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
4955	4	6	9	7	...	0	0

Star Pattern Database

512 sample signal						
0	333	z4	452	z8	...	z291
z5	414	z7	348	z10	...	z301
z5	108	338	0	403	...	z265
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
0	138	282	0	0	...	z313

Figure 4.14: Look-Up-Table and Star Pattern Database for second approach.

4.3.2 Shortlisting and Star Identification

To explain the process of shortlisting using hamming distance and star identification using spearman correlation, we will be using the example as shown in Fig 4.15. The image captured is represented by the dotted circle, with I_c being the center of the image. Let us say that star A is the closest I_c and thus, it will be selected as a reference star. The SPD of star A is represented by the solid circle in Fig 4.15, which is the same as was shown in Fig 4.12.

It should also be noted from Fig 4.15, that the captured image does not contain the stars 4, 7, 8, 10 and 12 which are a part of the SPD of star A (in Fig 4.12). Also, the image contains a false star F which is not a part of the SPD of star A. Stars 4, 7 and 8 are missed due to patch mismatch and stars 10 and 12, which are also nearest to the reference star A, are missed due to magnitude uncertainty. Thus, the example

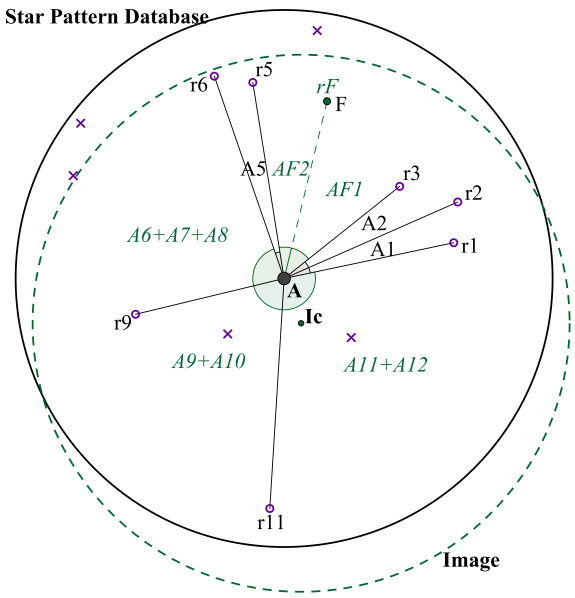


Figure 4.15: An example of the image captured for explaining the second approach.

shown in Fig 4.15 has all the problems which a star pattern recognition technique will face (also identified in chapter 3 of this thesis). Hence, the proposed technique aims to identify the stars in the image correctly and also provide fast recognition at the same time. We will explain the process of shortlisting using a 32 sample signal and star identification using a 512 sample signal as both LUT and SPD are prepared using the same parameters.

Shortlisting using Hamming Distance

Once the radial distances and relative angles are extracted from the image, a 32 sample signal is constructed following the same procedure as was used for preparing the LUT. For the image acquired in Fig 4.15, the 32 sample signal constructed is shown in Fig 4.16 and it is labeled as I_{32}. Hamming distance comparison plays an important role in a large variety of applications, especially in coding theory for comparing equal-length strings [93]. Hamming distance is also widely used in approximate near neighbor search for high dimensional data, such as images and document collections [94]. The hamming distance (D_H) between two strings is defined as the

number of characters in which they differ. It is calculated by Eq 4.19, Eq 4.20, and Eq 4.21.

$$D_H = \sum_{i=1}^{n} |x_i - y_i| \tag{4.19}$$

$$if x_i = y_i \Rightarrow D = 0 \tag{4.20}$$

$$if x_i \neq y_i \Rightarrow D = 1 \tag{4.21}$$

where x and y are two strings of length equal to n.

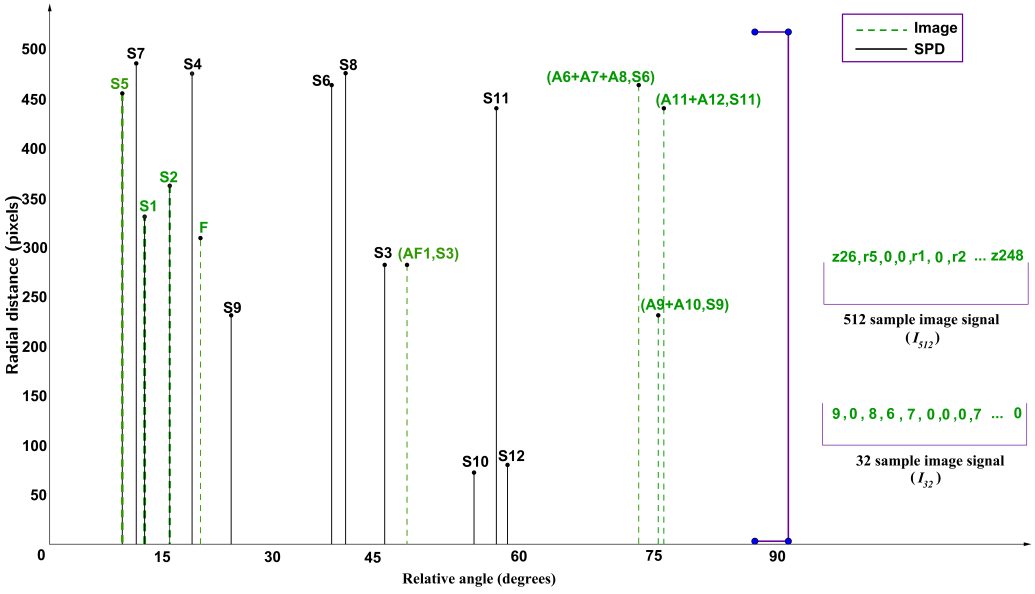


Figure 4.16: Signal construction from the image and comparison with SPD.

In our proposed technique, we construct a unique 32 sample signal for every star

ID and store it in LUT as was explained in section 4.3.1. We use the hamming distance to compare the signal constructed from the image acquired with signals stored in the LUT. In order to demonstrate, we have shown the calculation of hamming distance for the I_{32} and the corresponding 32 sample signal for the reference star A stored in the LUT (LUT_{A32}) in Fig 4.17. The total number of mismatches between I_{32} and LUT_{A32} are highlighted and thus, the calculated D_H equals to 8. The hamming distance between I_{32} and 32 sample signal of every star ID (stored in the LUT) is calculated, and only those star IDs are shortlisted whose hamming distance is less than 16, as shown in Fig 4.18. This process makes sure that the signal received from the image matches with at least half of the signal stored in the LUT for that particular star ID entry. This list of shortlisted star IDs only is then checked for exact star identification using spearman correlation as explained in the next sub-section.

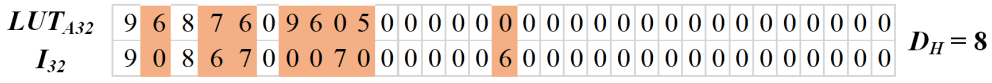


Figure 4.17: Calculation of hamming distance between I_{32} and LUT_{A32}.

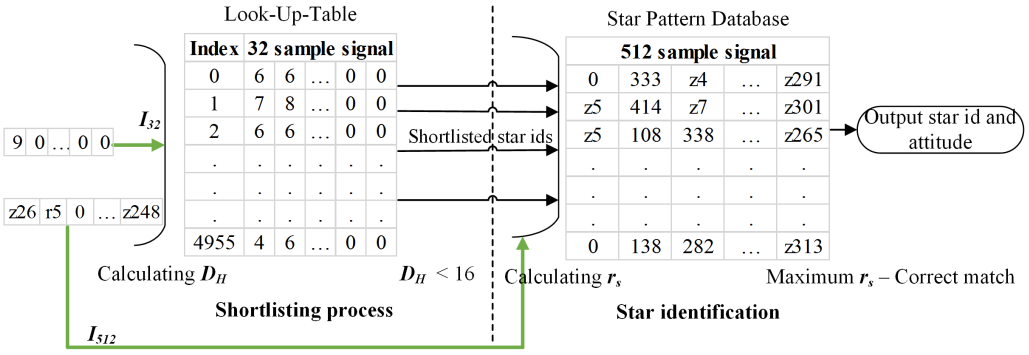


Figure 4.18: Shortlisting and star identification process - second approach.

Star Identification using Spearman-Correlation

The 512 sample signal is constructed from the features extracted from the image by the same process as was used for preparing the SPD for the proposed technique. For the image acquired in Fig 4.15, the construction of 512 sample signal is shown in Fig 4.16 and it is labeled as I_{512}. It is important to note that because of the problems of patch mismatch, magnitude uncertainty, and false stars, there can be many altered data points in the signal constructed from the image acquired when compared to the signal of the reference star stored in the SPD. Due to the above problems, there is no definite relationship between the data points of the image acquired and those of the SPD of the reference star. This lack of relationship is evident for the SPD of reference star A, and the image obtained, as shown in Fig 4.16.

If there is a very high amount of patch mismatch and magnitude uncertainty, the number of data points can also be very less in the signal constructed from the image acquired. Owing to the scenario mentioned above, the best way to measure the match of the image with the SPD is by calculating the *spearman correlation co-efficient* between the signal constructed from the image acquired and the signals of the shortlisted star IDs in the SPD. Normally, correlation refers to calculating the *pearson co-efficient*. However, the data variables must have constant variance and exhibit a linear relationship to calculate the pearson correlation coefficient, as well as, the number of samples must be significant for calculating the pearson co-efficient [95]. As the signal constructed from the image acquired does not satisfy the conditions mentioned above, we do not prefer pearson correlation for star identification. On the other hand, spearman correlation is a non-parametric measure of correlation and is appropriate for ordinal scale signals [96]. It does not assume a normal distribution of data or any definite relationship between the data points. Thus in the conditions mentioned above, spearman correlation coefficient will be an appropriate measure for the match between the image and the SPD. The spearman correlation coefficient (r_s) is calculated by Eq 4.22 and Eq 4.23.

$$r_s = 1 - \frac{6 \sum d_i^2}{n\left(n^2 - 1\right)} \tag{4.22}$$

$$d_i = rg\left(X_i\right) - rg\left(Y_i\right) \tag{4.23}$$

where $rg(X_i)$ and $rg(Y_i)$ are the ranks of each observation in the samples X and Y, and n is the number of samples.

The maximum correlation coefficient calculated between the 512-sample signals of the shortlisted star IDs in the SPD and the I_{512} (signal constructed from the image acquired) is regarded as the correct match for the reference star in the image. Fig 4.18 shows the entire process of shortlisting and star identification. Once the star ID which gives the maximum correlation coefficient is identified as the reference star, the nearest one or two stars to the reference star in the image (which can be found from the sequence of the SPD of the identified star ID) are also identified. As mentioned earlier, the existing QUEST or TRIAD method requires two to three correctly identified stars in the image for determining the attitude.

4.3.3 Selecting 32 samples for Shortlisting and 512 Samples for Recognition

Firstly, we will show the reason behind the selection of 32 sample (value of k as 5 in Eq 4.13 for shortlisting) and 512 sample (value of k as 9 in Eq 4.17 for star identification) based on the test results for different sample sizes. Concerning with the shortlisting process, hamming distance performance for different sample sizes (different of values of k in Eq 4.13) is shown in Fig 4.19. We have varied the value of k from 3 to 7, as the number of samples of the discrete signal required for shortlisting is not that high. The performance is measured in terms of two parameters - the number of shortlisted IDs (N_{hd}) (out of total 4956 star IDs) and the probability that the shortlisted entries contain the correct star ID (p_{hd}). Lower the N_{hd} and higher

the p_{hd}, better the performance. As can be seen in Fig 4.19, the N_{hd} is least (1071) for 8-sample signal ($k = 3$), however, the p_{hd} is also least (0.537) for this case. On the contrary, the p_{hd} is maximum (1.0) for 128 sample signal ($k = 7$), however, the N_{hd} is also very high (3565).

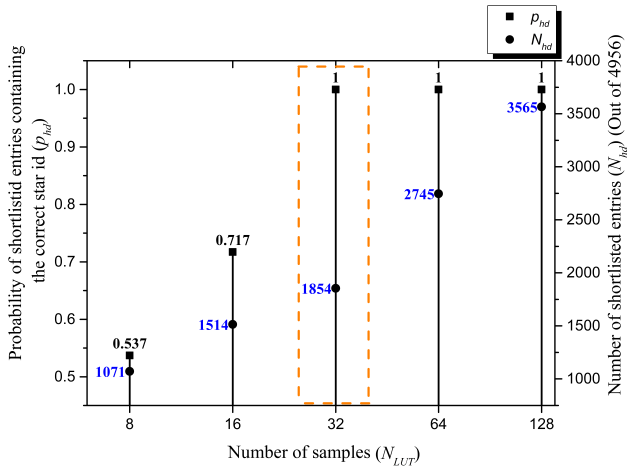


Figure 4.19: Comparison of performance of the second shortlisting approach for a different number of samples (N_{LUT}).

When the number of samples is greater than or equal 32 ($k \geq 5$), the p_{hd} is 1. However, if the sample size still goes on increasing the N_{hd} will also go on increasing. Thus, an optimum sample size of 32 gives a perfect p_{hd} and a low N_{hd} (almost one-third of the total star IDs). Thus, from Fig 4.19 it is evident that the 32-sample signal ($k = 5$ in Eq 4.13) is the perfect choice for the shortlisting technique to attain the best performance.

Concerning with the star identification process, spearman correlation performance for different sample sizes (different values of k in (8)) is shown in Fig 4.20. We have varied the value of k from 5 to 9 as the star identification requires a high number of samples to achieve a reliable identification. The performance is measured in terms of identification accuracy and recognition time required for different sample sizes. As can be seen from Fig 4.20, 32-sample signal ($k = 5$) gives a very low identification accuracy and the identification accuracy increases with the number of

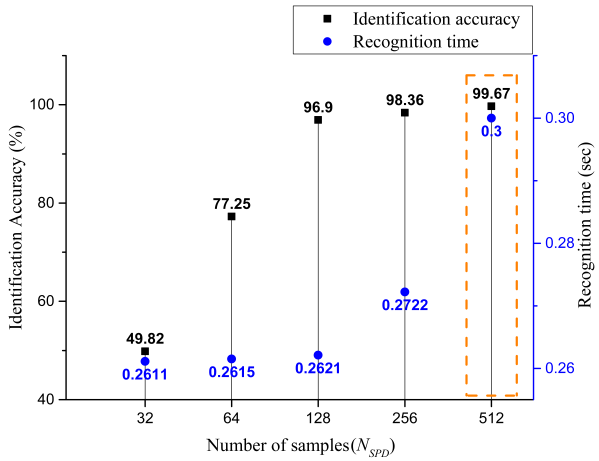


Figure 4.20: Comparison of performance of the second star identification approach for a different number of samples (N_{SPD}).

samples. The recognition time also increases with the number of samples. However, the increase in the recognition time is not drastic. We choose 512 sample size ($k = 9$ in Eq 4.17)) because the identification accuracy is very high and the recognition time is reasonable in this case. Because of the shortlisting due to hamming distance and the performance achieved by it (as explained earlier), this recognition time for the 512 sample size will be reduced.

4.4 Rotation Invariant Vector Frame

Until now we have seen many diverse approaches proposed for star identification. These approaches vary from extracting the geometric properties such as angles, distances, and area to developing a coded pattern formed by the neighboring stars. Any method selected for star identification must make sure that the features extracted for comparison and identification should be translation and rotation-invariant because the star images captured by the star tracker are usually translated and rotated by an *unknown factor* when compared to the ones stored in the SPD.

In this plethora of diverse approaches for star identification, two dimensional (2D) vectors on the image plane have not been used so far for forming the star pattern.

We propose to make use of the *additive* property of 2D vectors to solve the critical problems of patch mismatch, magnitude uncertainty, and noise (false stars) faced in star identification. However, it is important to note that though the 2D vectors connecting the stars on an image plane are by default *translation-invariant*, they are not *rotation-invariant* features. Thus, firstly, we need to make the 2D vectors rotation-invariant and later integrate their additive property to attain reliable star identification.

4.4.1 Construction of Rotation Invariant 2D Vectors

To solve the problem of rotation-variance of the vectors, we propose a novel way to construct a rotation-invariant vector frame for the 2D vectors connecting the stars on a normal X-Y plane. We will explain the construction of the rotation-invariant vector frame and how it is different from the normal 2D vector frame with the help of a simple example shown in Fig 4.21. In Fig 4.21, there are five points (representing stars) - 1, 2, 1', 2', and O on a 2D plane (X-Y) with point O being the center of the plane (center star). The stars 1'and 2' are the rotated version (by an angle θ in an anti-clockwise direction) of the stars 1 and 2 respectively on the 2D plane. Thus, we form four 2D vectors formed by joining the center star O with each of the neighboring stars 1, 2, 1', and 2'. These four column vectors are defined in a normal 2D plane (X-Y), and are labeled in Fig 4.21 as V_1, V_2, V_1', and V_2'. Additionally, we form two more vectors by joining the stars 1 and 2, and the stars 1'and 2' respectively. These two column vectors are again defined in a normal 2D plane (X-Y) and are labeled in Fig 4.21 as V_b and V_b'. As the stars 1'and 2' are the rotated versions of the stars 1 and 2 by an angle of θ in an anti-clockwise direction, the rotated vectors V_1', V_2', and V_b' can be expressed in terms of their corresponding original vectors V_1, V_2, and V_b respectively. This relation is given by a simple matrix multiplication of the rotation matrix R_θ with each of the original column vectors. Equations 4.24, 4.25, and 4.26 describe this relation for all the three vectors V_1', V_2', and V_b' on the 2D plane.

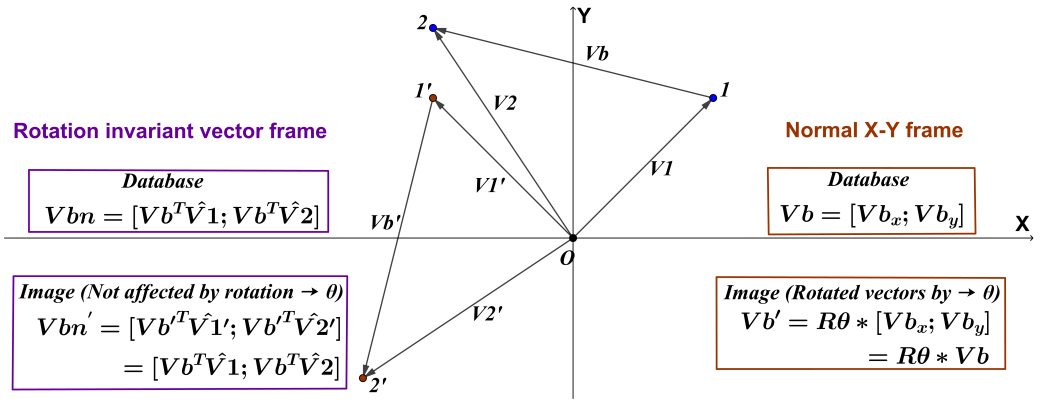


Figure 4.21: Construction of rotation-invariant vector frame for two dimensional (2D) vectors.

$$V_1' = R_\theta V_1 \tag{4.24}$$

$$V_2' = R_\theta V_2 \tag{4.25}$$

$$V_b' = R_\theta V_b \tag{4.26}$$

where R_θ is the rotation matrix for a rotation angle of θ in an anti-clockwise direction and is given by Eq 4.27.

$$R_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \tag{4.27}$$

Let us assume that the vector connecting the star 1 and star 2 is to be stored in the database for center star O i.e. the vector V_b has to be stored in the database

(visualize this as the SPD). In Fig 4.21, this SPD in the normal X-Y frame can be seen as V_{bx} and V_{by}, the corresponding x and y elements of the vector V_b. As stated earlier, stars 1'and 2' are the rotated version (by an angle θ) of the stars 1 and 2 respectively (visualized this as the image captured). Thus, the vector joining the star 1'and star 2'- V_b'(in the image captured) would not match with the vector V_b stored in the database (SPD). This scenario represents the problem of rotation-variance of the 2D vectors defined in a normal X-Y frame.

To solve the above problem, we propose to construct the boundary vector V_b in a new frame. As vector V_b connects the stars 1 and 2, the proposed idea is to construct it in terms of vectors V_1 and V_2. This new representation of the vector V_b is labeled as V_{bn} in Fig 4.21 and is given by Eq 4.28.

$$V_{bn} = \begin{bmatrix} V_b^T \widehat{V_1} & V_b^T \widehat{V_2} \end{bmatrix} \tag{4.28}$$

where V_b^T is the transpose of the vector V_b, $\widehat{V_1}$ and $\widehat{V_2}$, are the normalized (unit) vectors of V_1 and V_2 respectively. Hereafter, an identical symbol for a vector corresponds to the above-mentioned definition unless stated otherwise.

As can be seen from Eq 4.28, the first element of the vector V_{bn} is given by the dot product of V_b^T and $\widehat{V_1}$. The second element of the vector V_{bn} is given by the dot product of V_b^T and $\widehat{V_2}$. This newly constructed vector V_{bn} is now stored in the database for center star O (SPD). Now, for the rotated image captured, the vector connecting the star 1' and star 2' - V_b' in the new frame will be constructed in terms of the vectors from which it is formed i.e. the vectors V_1' and V_2'. This newly constructed vector is labeled as V_{bn}' in Fig 4.21 and is calculated (similarly to V_{bn}) by equations 4.29, 4.30, 4.31, 4.32, and 4.33.

$$V_{bn}^{'} = \begin{bmatrix} {V_b'}^T \widehat{V_1}' & {V_b'}^T \widehat{V_2}' \end{bmatrix} \tag{4.29}$$

where ${V_b'}^T \widehat{V_1}$' is calculated as:

$$(R_\theta V_b)^T \left(R_\theta \widehat{V_1}\right) \tag{4.30}$$

$$= V_b^T R_\theta^T R_\theta \widehat{V_1} \tag{4.31}$$

$$= V_b^T \widehat{V_1} \tag{4.32}$$

Similarly, $V_b'^T \widehat{V_2}$' will be calculated as (8).

$$V_b'^T \widehat{V_2} = V_b^T \widehat{V_2} \tag{4.33}$$

Thus, the vector V_{bn}' is finally given by Eq 4.34 which is the same as the vector V_{bn} (which was given by Eq 4.28).

$$V_{bn}^{'} = \begin{bmatrix} V_b^T \widehat{V_1} & V_b^T \widehat{V_2} \end{bmatrix} \tag{4.34}$$

Hence constructing the boundary vector (V_b) between two stars in terms of their original vectors (V_1 and V_2) as shown by Eq 4.28 will make the new boundary vector (V_{bn}) rotation invariant. This idea of constructing the connecting vector in terms of their original vectors is the newly proposed rotation-invariant vector frame for the 2D vectors connecting the stars on a normal X-Y plane.

4.4.2 Shortlisting Technique based on the Rotation Invariant Vector Frame

In this section, firstly, we will describe the construction of the SPD for the proposed shortlisting technique, and later we will explain the proposed method. SAO J2000 star catalog is used for preparing the SPD. The stars whose relative magnitude (M_v) is less than 6.3 are chosen for forming the SPD. Every star is considered at the center, and all the neighboring stars which lie within the FOV of the center star are considered for building the rotation invariant vector pattern for the center star. For example, in Fig 4.22, ten stars labeled from 1 to 10 which lie within the FOV of the center star O will be considered for constructing the rotation invariant vector pattern for the center star O.

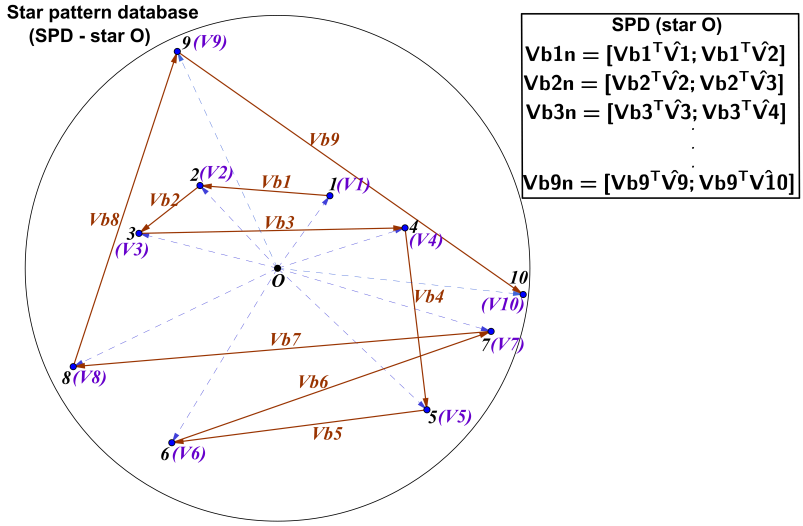


Figure 4.22: Construction of the star shortlisting technique database based on rotation-invariant vector frame.

For constructing the rotation invariant vector pattern for star O, the neighboring stars are taken into consideration in the order of their increasing distance from the star O. A boundary vector path is then traced from the nearest neighboring star to the farthest neighboring star. In Fig 4.22, this path starts from star 1 and ends

at star 10 (shown in brown). However, if these boundary vectors (V_{b1}, $V_{b2}...V_{b9}$) in Fig 4.22 are used directly for constructing the SPD of star O, they will not be rotation invariant as was explained earlier. To make these boundary vectors rotation-invariant, we construct them in a new rotation-invariant frame, which was described in section 4.4.1. For example, the boundary vector V_{b1} in Fig 4.22 will be now constructed in terms of the vectors of star 1 and star 2 i.e. the vectors V_1 and V_2 respectively. This new boundary vector V_{b1n} is given by Eq 4.35. Similarly, the new boundary vector V_{b2n} will be given by Eq 4.36.

$$V_{b1n} = \begin{bmatrix} V_{b1}^T \widehat{V_1} & V_{b1}^T \widehat{V_2} \end{bmatrix} \tag{4.35}$$

$$V_{b2n} = \begin{bmatrix} V_{b2}^T \widehat{V_2} & V_{b2}^T \widehat{V_3} \end{bmatrix} \tag{4.36}$$

Thus, all the nine boundary vectors are converted similarly to a new rotation-invariant frame and stored as the rotation-invariant boundary vector pattern for star O, labeled as SPD (star O) in Fig 4.22. There are totally 6903 stars in the SAO J2000 star catalog which have an M_v of less than 6.3. Hence the SPD is prepared as the rotation-invariant boundary vector pattern for all of the 6903 stars. The SPD prepared for the same is shown later in Fig 4.24.

We will now explain the proposed shortlisting technique with the help of an example in Fig 4.23. In Fig 4.23, we show the visualization of an image captured which is usually swayed and biased when compared with the corresponding entry in the SPD. The dotted green circle in Fig 4.23 is the image captured with I_c as the center of the image, and the solid pink circle is the SPD of the star O (same as Fig 4.22). As can be seen in Fig 4.23, star 7 and star 10 are missed to be captured in the image because the image captured does not coincide exactly with the SPD of star O. Adding to above-missed stars, star 1 (nearest star to star O in the SPD) is also missed because of the magnitude uncertainty of the star tracker.

Moreover, there is also a false star F present in the image captured. Thus, Fig 4.23 contains all the major problems faced by a star tracker- translation of the image (patch mismatch), false stars, and magnitude uncertainty. For an uncomplicated understanding of the process of the shortlisting technique, we have not shown rotation of the image captured in Fig 4.23. Nevertheless, the proposed rotation-invariant boundary vector construction was already explained earlier in section 4.4.1 and in the Star Pattern Database Generation of this section.

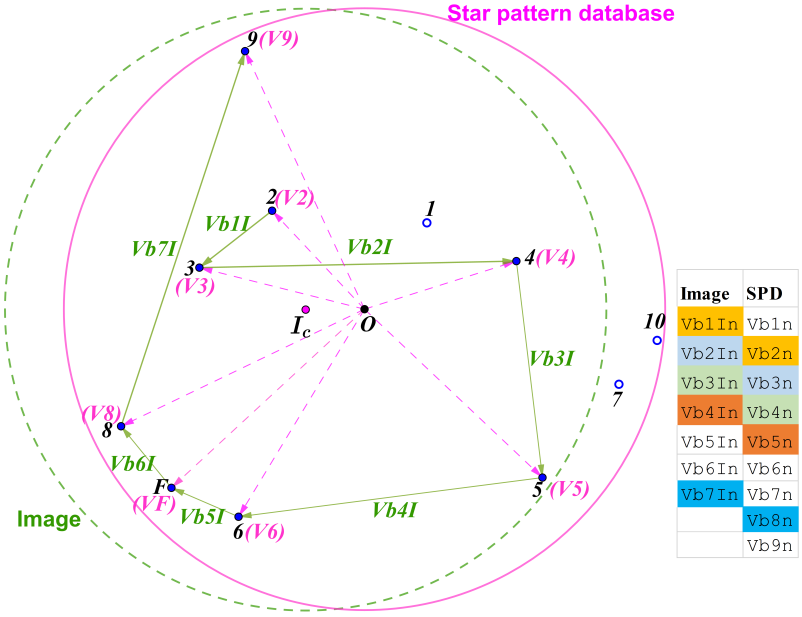


Figure 4.23: An illustrative example to explain the proposed shortlisting technique based on rotation-invariant vector frame.

Rotation-invariant boundary vectors are constructed for the image captured by following the similar method that was explained to generate the SPD for the proposed technique. The travel path (V_{b1I}, V_{b2I} ...V_{b7I}) for the eight stars captured in the image is shown in green in Fig 4.23. The above-listed boundary vectors are then converted to a rotation-invariant frame (similar process as was explained in section 4.4.1 and in the SPD generation) and are constructed as V_{b1In}, V_{b2In} ... V_{b7In} in the image column in Fig 4.23. The vectors calculated from the image are now checked for match with

the SPD of star O. For the case of star O, it can be seen that there are total five matches of the vectors in the image with the ones stored in the SPD of star O, i.e. V_{b1In} matches with V_{b2n}, V_{b2In} matches with V_{b3n}, V_{b3In} matches with V_{b4n}, V_{b4In} matches with V_{b5n}, and V_{b7In} matches with V_{b8n}. These matches are highlighted in Fig 4.23 and they can be easily checked by comparing the Fig 4.23 and Fig 4.22.

Once the rotation-invariant boundary vectors are calculated for the image captured, they are checked for the number of matches with the rotation-invariant boundary vector pattern for each of the star ID in the SPD. This whole process is shown in Fig 4.24. A vote counter is initialized to zero for all the star IDs. Later, the vector pattern of the image (V_{bIn}) is checked for the match (within a tolerance range) with the vector pattern of each star ID stored in the SPD. For every match, the vote counter is increased by 1 for that particular star ID. Finally, all the votes are arranged from highest to lowest, and only the star IDs which lie in a specific top range will be shortlisted, as shown in Fig 4.24. The particular limit of the star IDs selected to be shortlisted is based on the performance of the shortlisting technique on simulated images as explained later in chapter 5.

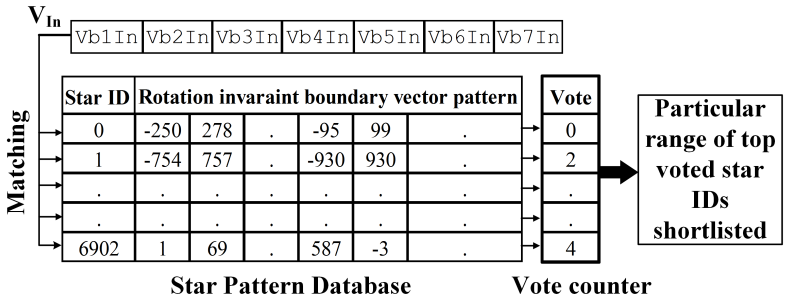


Figure 4.24: Star pattern database and shortlisting of star IDs based on rotation-invariant vector frame.

4.4.3 Star Identification based on Rotation Invariant Vector Frame

We have shown the idea of making the 2D vectors rotation-invariant by constructing them in a new frame in section 4.4.1. This new representation will solve the problem of comparing the vectors even when the image captured is rotated by any arbitrary amount compared to its corresponding entry in the SPD. However, apart from the translation and rotation of the image, there are also missing and false stars which result from the problem of patch mismatch, magnitude uncertainty, and noise in the image. We make use of the additive property of the 2D vectors to solve the above problems. We will now explain the proposed idea to integrate the additive property of the vectors with the rotation-invariant vector frame for solving the problem of missing and false stars.

- **Scenario of missing star** - In Fig 4.25, we present a simple scenario of missing a star in the image when compared to its corresponding entry in the SPD. In Fig 4.25, let us assume that star O is the reference star and stars 1, 2, 3, and 4 are the neighboring stars to star O. So, we aim to form the star pattern for star O. It must be noted from Fig 4.25 that the database of star O contains stars 1, 2, 3, and 4 whereas the image has only stars 1, 3, and 4 (star 2 is missed). We will explain the process of integrating the additive property of the vectors with the rotation-invariant vector frame with the aid of this example in Fig 4.25.

  Considering the database, which contains the stars 1, 2, 3, and 4, we will now explain the formation of the star pattern for star O. In Fig 4.25, we will also assume that the stars 1, 2, 3 and 4 are arranged in an increasing order of their distance from the reference star O. Thus, we define a travel path from the nearest neighboring star to the farthest neighboring star i.e. star 1-> star 2-> star 3-> star 4 in the database. This travel path will comprise of three vectors V_{b1}, V_{b2}, and V_{b3} (not rotation-invariant) as shown in brown in Fig 4.25. As explained earlier in section 4.4.1, we will then construct the first vector V_{b1} in a rotation-invariant frame. Hence the first rotation-invariant vector in the database of star O will be V_{db1} in Fig 4.25, and is given by Eq 4.37.

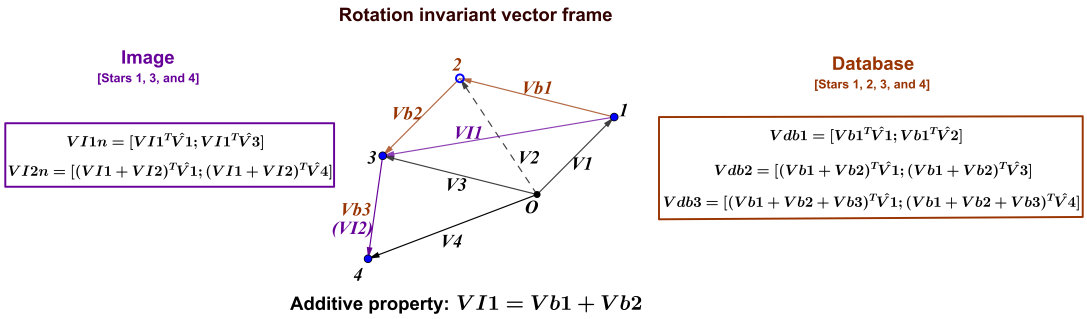


Figure 4.25: Integration of the additive property with the rotation-invariant vector frame for the scenario of a missing star.

$$V_{db1} = \left[V_{b1}^T \widehat{V}_1 \quad V_{b1}^T \widehat{V}_2\right] \tag{4.37}$$

As we travel along the path, we encounter the second vector V_{b2}. Instead of constructing it in terms of vector V_2 and vector V_3 which we explained earlier, we will first add the vectors V_{b1} and V_{b2} and later construct this added vector $V_{b1} + V_{b2}$ in terms of vector V_1 (starting point of the added vector) and vector V_3 (ending point of the added vector). Hence the second vector to be stored in the database will be the rotation-invariant added vector V_{db2} in Fig 4.25, and is given by Eq 4.38.

$$V_{db2} = \left[(V_{b1} + V_{b2})^T \widehat{V}_1 \quad (V_{b1} + V_{b2})^T \widehat{V}_3\right] \tag{4.38}$$

As we go on traveling the path, we keep on adding the vectors that we encounter and construct them in the rotation-invariant frame in terms of vectors of the starting point and ending point of the path. Following the above concept, the final vector to be stored in the database in Fig 4.25 will be the addition of the three vectors V_{b1}, V_{b2}, and V_{b3} constructed in a rotation-invariant frame in terms of vector V_1 (starting point) and vector V_4 (ending point). This final

vector is labeled as V_{db3} in Fig 4.25, and is given by Eq 4.39.

$$V_{db3} = \left[(V_{b1} + V_{b2} + V_{b3})^T \widehat{V_1} \quad (V_{b1} + V_{b2} + V_{b3})^T \widehat{V_4} \right] \tag{4.39}$$

Thus, the database of star O consists of the rotation-invariant additive vectors V_{db1}, V_{db2}, and V_{db3} as shown in Fig 4.25.

Considering the image, which contains only the stars 1, 3, and 4, we will form the star pattern for star O similar to we prepared its database in Fig 4.25. As per the earlier mentioned assumption of the increasing order of distance of the neighboring stars, the travel path in the image will be from the nearest neighboring star to the farthest neighboring star, i.e. star 1 -> star 3 -> star 4. This travel path will comprise of only two vectors V_{I1} and V_{I2} (not rotation-invariant) as shown in purple in Fig 4.25. We will construct the rotation-invariant additive vector pattern for the image similar to we did for the database. Constructing the rotation-invariant additive vector pattern for the image, the first rotation-invariant vector in the image will be V_{I1n}, and is given by Eq 4.40.

$$V_{I1n} = \left[V_{I1}^T \widehat{V_1} \quad V_{I1}^T \widehat{V_3} \right] \tag{4.40}$$

As we travel along the path, the second rotation-invariant additive vector in the image will be V_{I2n}, and is given by Eq 4.41.

$$V_{I2n} = \left[(V_{I1} + V_{I2})^T \widehat{V_1} \quad (V_{I1} + V_{I2})^T \widehat{V_4} \right] \tag{4.41}$$

It is important to note two things from the Fig 4.25 - Vector V_{I2} and vector V_{b3} are the same; Vector V_{I1} is the addition of the vector V_{b1} and vector V_{b2}.

Hence when it comes to comparing the rotation-invariant additive vector pattern of the image for star O (in Fig 4.25) with the one in its database, the vector V_{I1n} will match with the vector V_{db2}, and the vector V_{I2n} will match with the vector V_{db3}. The above matching results from the process of integrating the additive property of the vector with the rotation-invariant vector frame. These rotation-invariant additive vector sequences (of the image and the database) will match somewhere along the path even if some star is missed in between. For example in Fig 4.25, even if the star 2 was missed, the vector sequences matched between the image and the database. This concept can be extended to a situation when any number of stars are being missed in the image when compared to its SPD. We will still be able to find a match between the rotation-invariant additive vector sequences of the image and the SPD somewhere along the travel path.

- **Scenario of false star** - It is a typical scenario that the light reflected from a planet or a meteor in space is falsely recognized as a star in the image captured by the star tracker. Thus, a star pattern recognition technique must be robust to the presence of false stars in the image. In Fig 4.26, we show the scenario of a false star F present in the image when compared to the database. In Fig 4.26, star O is the reference star and the stars 1, 2, and 3 are the neighboring stars present in the database of star O. Considering the image, star F is present along with the stars 1, 2, and 3 as one of the neighboring stars of star O. In Fig 4.26, we will also assume that the stars 1, F, 2, and 3 are arranged in the increasing order of their distance from the star O.

  Considering the database, we will now construct the rotation-invariant additive vector sequence for star O in Fig 4.26 in a similar way described earlier in the scenario of missing star. The travel path in the database of star O in Fig 4.26 will be: star 1 -> star 2 -> star 3. This travel path will comprise of two vectors V_{b1} and V_{b2} (not rotation-invariant) as shown in brown in Fig 4.26. Hence the first rotation-invariant vector in the sequence of the database of star O will be V_{db1} in Fig 4.26, and is given by Eq 4.42.

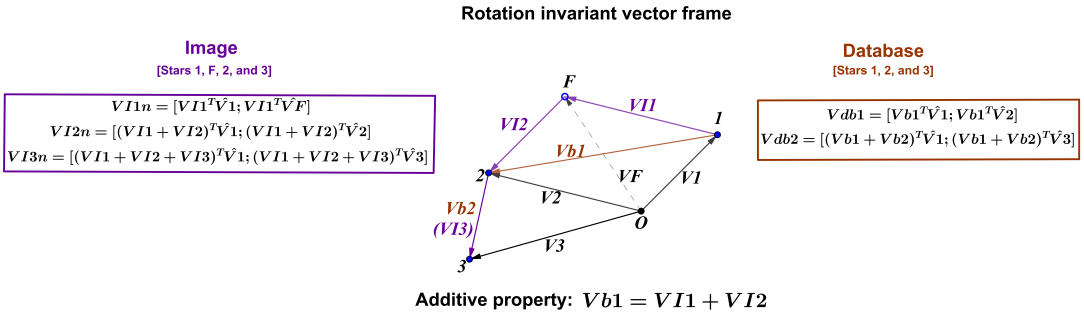


Figure 4.26: Integration of the additive property with the rotation-invariant vector frame for the scenario of a false star.

$$V_{db1} = \left[V_{b1}^T \widehat{V}_1 \quad V_{b1}^T \widehat{V}_2\right] \tag{4.42}$$

As we travel along the path, the second rotation-invariant additive vector in the sequence of the database of star O will be V_{db2}, and is given by Eq 4.43.

$$V_{db2} = \left[(V_{b1} + V_{b2})^T \widehat{V}_1 \quad (V_{b1} + V_{b2})^T \widehat{V}_3\right] \tag{4.43}$$

Considering the image, which contains four stars 1, F, 2, and 3, we will form the star pattern of star O similar to we did for its database. For the case of the image, the travel path will be star 1 -> star F -> star 2 -> star 3. This travel path will comprise of three vectors V_{I1}, V_{I2}, and V_{I3} (not rotation-invariant) as shown in purple in Fig 4.26. Constructing the rotation-invariant additive vector pattern for the image, the first rotation-invariant vector in the image will be V_{I1n}, and is given by Eq 4.44.

$$V_{I1n} = \left[V_{I1}^T \widehat{V}_1 \quad V_{I1}^T \widehat{V_F}\right] \tag{4.44}$$

Moving along the travel path, the second rotation-invariant additive vector in the image will be V_{I2n}, and is given by Eq 4.45.

$$V_{I2n} = \left[(V_{I1} + V_{I2})^T \widehat{V_1} \quad (V_{I1} + V_{I2})^T \widehat{V_2}\right] \tag{4.45}$$

As we come to the ending point (star 3) of the travel path, the final rotation-invariant additive vector in the image will be V_{I3n}, and is given by Eq 4.46.

$$V_{I3n} = \left[(V_{I1} + V_{I2} + V_{I3})^T \widehat{V_1} \quad (V_{I1} + V_{I2} + V_{I3})^T \widehat{V_3}\right] \tag{4.46}$$

It should be noted from Fig 4.26 that the vector V_{I3} is the same as vector V_{b2}. As well as, the vector V_{b1} in the database is the summation of the vectors V_{I1} and V_{I2} of the image.

Comparing the rotation-invariant additive vector sequence of the star of the image for star O (in Fig 4.26) with the one in its database, it can be seen that the vector V_{I2n} will match with the vector V_{db1}, and the vector V_{I3n} will match with the vector V_{db2}.

Thus, it can be seen that the rotation-invariant additive vector sequences (of the image and the database) will match somewhere along the path even if there is a false star present in the image. The same concept can also be extended in the case of more than one false star present in the image. It is important to note that merely storing the distances or vectors between the stars would not solve the problem of missing and false stars faced by the star identification algorithms. Thus, as demonstrated in this section, the proposed approach of integrating the additive property of the vectors with the rotation-invariant vector frame will solve the above problems. We will now explain the SPD construction based on this concept.

Star Pattern Database Construction based on Rotation Invariant Additive Vector Sequence

We will now describe the construction of the SPD based on the concept of the rotation-invariant additive vector sequence. Firstly we will show the importance of selecting the starting point for forming the vector sequence. Later we will show an analysis concerning with the nearest stars to a reference star, which is performed on real star images. Based on this analysis we will arrive at the reasoning behind the selection of a specific starting point for the construction of the proposed vector sequence. Finally, we will explain the process of building of the SPD with the aid of an example.

- **Selection of starting point for the rotation-invariant additive vector sequence** - It is important to note that the procedure of constructing the rotation-invariant additive vector sequence is dependent upon the starting point of the travel path. If the starting point is not the same in the SPD and the image, the vector sequence developed will be completely different. We will explain this using an example in Fig 4.27. In Fig 4.27, star O is the reference star, and it has stars 1, 2, 3, and 4 in its database, however, in the image, star 1 is missed. Let us assume that the stars 1, 2, 3, and 4 are arranged in increasing order of their distance from the reference star O. As can be seen in Fig 4.27, the nearest star to the reference star O is star 1 in the database. However, it is star 2 in the image. Thus the travel path in the database is star 1 -> star 2 -> star 3 -> star 4, however, the travel path in the image is star 2 -> star 3 -> star 4. So, the vector sequence developed for star O in the image is completely different from the vector sequence stored in the SPD of star O (as also depicted in Fig 4.27). The sole reason for the different vector sequences of the image and the SPD in Fig 4.27 is both having different starting points of the travel path, i.e. star 1 in the SPD and star 2 in the image. Hence it is important to make sure that the starting point in the SPD is not missed and is the same as in the image captured.

  Earlier while explaining the construction of the rotation-invariant additive vec-

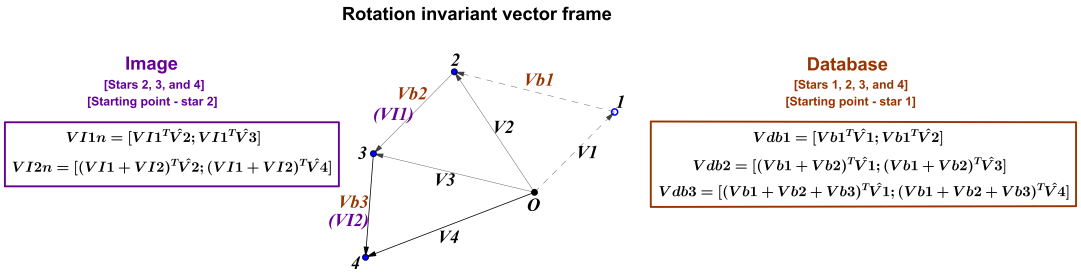


Figure 4.27: The mismatch between the vector sequences of the image and the SPD due to different starting points.

tor sequence, we chose the starting point as the *nearest neighboring star* to the reference star. However, in real images, there is quite often a scenario when the nearest neighboring star to the reference star (in the SPD) is missed to be captured in the image. This scenario is also depicted in Fig 4.27, which happens because of the apparent magnitude (M_v) uncertainty in space and particularly when the nearest neighboring star is low in brightness (or) has a high value of M_v. In general, the intensity of the star (I) in the image is inversely proportional to its relative apparent (M_v) in the star catalog, which can be deduced from Eq 4.47.

$$M_v = -2.512 \log (I) \tag{4.47}$$

As the starting point may be missed because of the low brightness of the star, we propose to select the starting point as the brightest nearest star to the reference star in the SPD. By the brightest nearest star, we mean the star which is the brightest (having the lowest M_v) amongst a selected number of closest stars to the reference star. The number of nearest stars considered for such a selection is based on the results of an analysis performed on 2397 identified real images captured from the star tracker SST-20S.

The purpose of the analysis performed is to fix the number of nearest stars to be considered in the SPD of a reference star when searching for the brightest

star amongst those nearest stars. We aim to make sure that if the brightest star (lowest M_v) amongst those fixed number of nearest stars in the SPD is selected as the starting point, it must not be missed in the image. As all the 2397 images are already identified images, the process of performing this analysis is explained in steps as below:

1. Firstly, we select the star nearest to the center of the image as the reference star.
2. In the SPD of the same reference star, we choose the starting point of the travel path as the brightest star (lowest M_v) amongst the number of nearest stars considered as the starting point of the travel path in the SPD and record the star ID of the starting point.
3. Coming back to the image, we will now consider the same number of nearest stars (as in the above step) to the reference star in the image and record the star IDs of those nearest stars.
4. If the star ID of the starting point in the SPD is one of the registered star IDs of the closest stars considered in the image, then it is ensured that the starting point in the SPD is present in the image. Hence we increase the count by 1 for the particular number of nearest stars considered.

We will now examine the results (shown in Fig 4.28) of the above analysis performed on the real images. In Fig 4.28, the *x-axis* represents the number of nearest stars considered and the *y-axis* represents the total count (out of the 2397 images) when the step 4 of the above analysis is correct for the particular number of nearest stars considered. For example, if three closest stars to the reference star are considered, and it is found that the star ID of the starting point amongst the three nearest stars in the SPD is one of the star IDs of the three nearest stars to the reference star in the image, then the count will be increased by 1. As can be seen from the results in Fig 4.28, when only one nearest star is considered, i.e. the nearest and the brightest star are the same in the SPD, there are only 517 cases when step 4 of the analysis is correct.

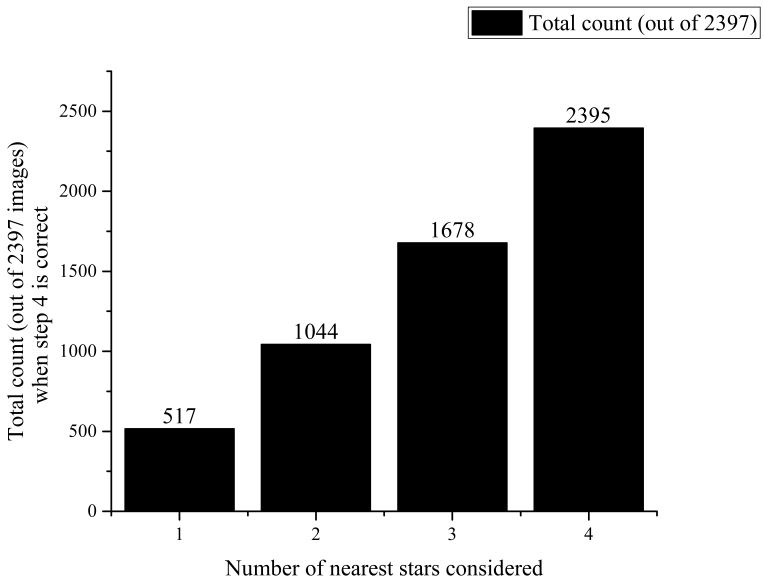


Figure 4.28: Analysis performed on real images to select the number of nearest stars to be considered for choosing the starting point.

When two nearest stars are considered, the number of instances when the step 4 will be correct increases to 44% of 2397 images. Even when three nearest stars are considered, only 1678 out of the 2397 cases satisfy step 4. We stop our analysis when four nearest stars are considered as the results in Fig 4.28 depict that for this case there is almost 100% (2395 out of 2397) chance that the star ID of the starting point amongst the four nearest stars in the SPD is one of the star IDs of the four nearest stars to the reference star in the image.

Thus, from the results (in Fig 4.28) of the analysis on the 2397 real images, we come to a reasoning that when four nearest stars are considered to a reference star in the SPD, the starting point amongst them (brightest star or lowest M_v) will be present in the image. Hence the starting point of the travel path can be made the same in the SPD and the image. The above match can be achieved by storing the *distance* of the starting point from the reference star in the SPD. As this star will be present in the image, we can later compare the distance of the four nearest stars to the reference star in the image and choose

the starting point in the image which matches the distance stored in the SPD. This is explained clearly later where we describe the process of star pattern recognition.

It is important to note that we are not utilizing the brightness of the stars in the image as a feature for comparison purposes. The sole purpose of the intensity is only to make sure that starting point star is present in the image. The brightness of the starting point is not stored in the SPD. Instead, the distance of the starting point from the reference star is stored in the SPD as described earlier. We will now explain the generation of the SPD.

- **Star Pattern Database generation** - We adopt the star catalog SAO J2000 to construct the SPD for our proposed technique. We will explain the process of feature extraction and generation of the SPD with the help of an example in Fig 4.29. This process will be described using a generalized framework. However, we will be using the parameters of the star tracker SST-20S to construct the SPD to test the proposed technique on simulated and real images later in chapter 5.

  1. Arranging the stars - For constructing the SPD of the center star O in Fig 4.29, all the neighboring stars which lie within the FOV of star O and have a M_v of less than 6.3 are considered for forming the rotation-invariant additive vector sequence for the star O. As shown in Fig 4.29, star O contains totally 11 stars in its FOV which are labeled from 1 to 11. Euclidean distance (r_n) of the neighboring stars from the star O is calculated by Eq 4.48.

$$r_n = \sqrt{(x_n - x_c)^2 + (x_n - x_c)^2} \tag{4.48}$$

  where x_n and y_n are the 2D co-ordinates of the neighboring star, and x_c and y_c are the 2D coordinates of the center star O in the X-Y frame.

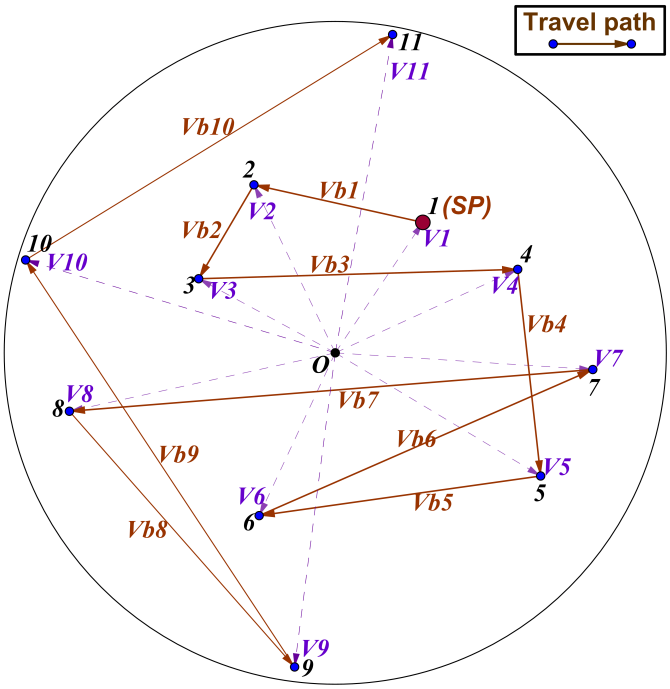


Figure 4.29: Generation of star pattern database for star O.

Once the distance is calculated, the neighboring stars are arranged in an increasing order of their distance from the star O.

2. Selecting the starting point - The next step is to select the starting point (SP) of the travel path. As concluded from the results of the analysis in Fig 4.28, the brightest star (lowest M_v) amongst the four nearest stars to the center star will be considered as the SP. In Fig 4.29, we assume that the star 1 has the lowest M_v amongst the four stars (star 1, star 2, star 3, and star 4) for simple understanding. Thus, we select the star 1 as the SP for the travel path in Fig 4.29. We will also store the distance of this SP from the center star O in the SPD of star O (as shown in Fig 4.30(a)). This distance will be used later for comparing with the distances of the four nearest stars to the reference star in the image.

3. Constructing the rotation-invariant additive vector sequence - Once the starting point is selected and the stars are arranged in increasing order of

	Distance of the SP	Vector sequence				
SPD (Star *O*)	r_n **(SP)**	***Vdb1***		***Vdb2***		…
	r_n **(Star 1)**	$Vb1^T \hat{V}1$	$Vb1^T \hat{V}2$	$(Vb1+Vb2)^T \hat{V}1$	$(Vb1+Vb2)^T \hat{V}3$	…

(a)

Star ID	r_n (SP)	*Vdb(sp)*		*Vdb(sp+1)*		…
0	147	-192	224	-593	594	…
1	113	-217	219	-296	301	…
.	.	.	.	.	.	…
.	.	.	.	.	.	…
6902	88	-508	512	-30	93	…

(b)

Figure 4.30: Star pattern database - (a) Star O. (b) For the proposed technique.

their distance, the travel path is formed. The travel path will start at the SP and end at the farthest neighboring star from the reference star. Thus, for the example in Fig 4.29, the travel path will be star 1 -> star 2 -> ... -> star 11. This travel path is highlighted in a solid brown line in Fig 4.29. Once the travel path is formed we can construct the rotation-invariant additive vector pattern as move along the travel path. The process of forming this vector sequence remains the same as was described in the examples in Fig 4.25 and Fig 4.26. The only difference is that we had considered fewer stars in Fig 4.25 and Fig 4.26, whereas in Fig 4.29 we have considered 11 stars. The distance of the SP and the vector sequence constructed is then stored in the SPD of star O as shown in Fig 4.30(a).

All the stars which have a M_v of less than 6.3 in the star catalog SAO J2000 will be considered for forming the SPD. The selection of the starting point (SP) and the formation of the vector sequence will be similar to described in the process for star O. The SPD prepared is shown in Fig 4.30(b). As can be seen in Fig 4.30(b), the first entry in the SPD of every star ID is stored as the distance of the SP selected for that star ID. The later part of the SPD contains the rotation-invariant additive vector sequence constructed for that star ID.

Star Pattern Recognition based on Rotation Invariant Additive Vector Sequence

We will now explain the process of star pattern recognition using an illustrative example shown in Fig 4.31. In Fig 4.31, the captured image is shown by the dotted green circle and the SPD is shown by the solid brown circle. Point I_c is the center of the image captured. As can be seen from Fig 4.31, the star O (same center star as in Fig 4.29) does not lie at the center of the image i.e. the captured image does not coincide exactly with the SPD of the star O (as in Fig 4.29), which is normally the scenario in real star images. For a straightforward explanation of the proposed technique, we have not shown the rotation of the image captured in Fig 4.31. Nevertheless, the proof of concept for the rotation-invariant vector frame was explained earlier. As star O lies closest to the point I_c, we will choose star O as the reference star to be identified.

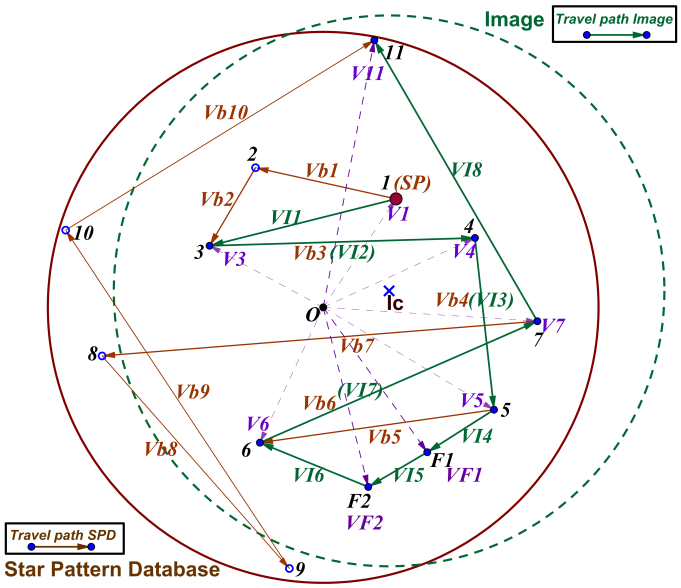


Figure 4.31: Illustrative example to explain the proposed star pattern recognition technique.

It should also be noted in Fig 4.31 that the stars 8, 9, and 10 which are a part

of the SPD of star O are not present in the image captured. These stars are missed because of the patch mismatch. Adding to the above-missed stars, star 2 is also missed in the image obtained because of the magnitude uncertainty, as well as there are two false stars - $F1$ and $F2$ present in the image captured. These two false stars are not a part of the SPD of star O. Thus, the example of the image captured in Fig 4.31 has all the problems faced in star identification. Hence the proposed star pattern recognition technique aims to develop the rotation-invariant additive vector sequence for the star O in the image captured and compare it correctly with the sequence stored in the SPD of the star O.

We will develop the rotation-invariant additive vector sequence for the star O in the image captured in a similar way we prepared the SPD of the star O. Firstly, the neighboring stars to star O are to be arranged in increasing order of their distance from star O. This arrangement of the stars in the image captured in Fig 4.31 starts at star 1 (nearest star) and ends at star 11 (farthest star). Next step is to identify the starting point of the travel path. For this purpose, we select the nearest four stars to the star O in the image, i.e. the star 1, star 3, star 4 and star 5. For these four stars, we check which stars distance matches with the distance stored in the SPD of the star O (Fig 4.30(a)). We can easily confirm that the distance of star 1 will match with the distance stored in the SPD of the star O. Thus, for comparing the vector sequence of star O in the image with the vector sequence stored in its SPD, star 1 is selected as the starting point of the travel path in the image in Fig 4.31. Thus, the travel path for the vector sequence of star O in the image will be star 1 -> star 3 -> star 4 -> star 5 -> star F1 -> star F2 -> star 6 -> star 7 -> star 11. This travel path in the image comprises eight vectors V_{I1}, V_{I2} ... V_{I8} (not rotation-invariant) and is highlighted by a solid green line in Fig 4.31. The final step is to generate the rotation-invariant additive vector sequence as we move along the travel path in the image. Once, this vector sequence of star O in the image is generated, we will check for its match with the one stored in the SPD of star O. For our example in Fig 4.31, the vector sequence generated from the image (with the starting point as star 1) will have six matches with the vector sequence stored in the SPD of the star O. The reference star is identified as the star ID in the SPD which receives the maximum

number of matches between the vector sequence of the image and the ones stored in the SPD.

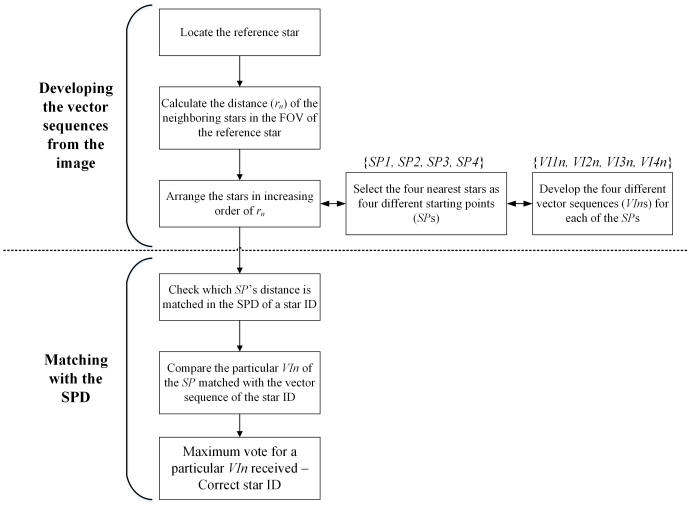


Figure 4.32: Flowchart for the proposed star pattern recognition technique.

It should be noted from the above-described process of matching that the starting point (SP) of the vector sequence (V_{In}) in the image may be one of the four nearest stars to the reference star depending upon the distance matched in the SPD of a particular star ID. Thus, before we start the matching process, we build four different vector sequences (V_{Ins}) considering the four nearest stars independently as the starting points (SPs) in the image. Later, we only need to check which starting point's distance matches with the distance stored in the SPD of a particular star ID. We will then compare the specific V_{In} of the matched SP in the image with the vector sequence stored in the SPD of the star ID. This whole process is shown as a flowchart in Fig 4.32. The reference star in the image is identified as the star ID which receives the maximum number of matches during this process of comparison of the vector sequence.

Once the reference star is assigned a star ID in the image, the other neighboring stars in the image can be easily identified from the SPD sequence of that star ID. Attitude determination techniques like QUEST only require three stars to be identified

in the image for attitude calculation. Hence attitude can be easily calculated after the identification of the reference star and its two neighboring stars in the image.

4.5 Failure Analysis

Having described all the three approaches, it is necessary to do a possible failure analysis of the same. In this section, we will describe some possible scenarios in which the proposed methods may face challenges for reliable star shortlisting and identification.

1. Star nearest to the center of the image is a false star - As explained earlier in all the three approaches, the star nearest to the center of the image is selected as the reference star to be identified in the image captured. However, if the reference star is itself a false star, it does not have a match in the SPD. In this type of scenario, all the three proposed approaches as well as all the star pattern recognition techniques which base their selection of the reference star on the shortest distance criterion will fail. A possible solution is to select the second nearest star to the center of the image as the reference star and re-run the entire process if a match cannot be found in the SPD for the previously selected reference star. This solution will also be adopted to solve a similar scenario when two nearest stars are having the same distance from the center of the image.

2. Initial angle mismatch - The step of the initial match is crucial for the first proposed approach of star identification. There may arise a situation when a false star(s) form the part of the initial match for the first approach. In such a scenario, the first approach would not be able to find an initial match in the SPD and thus, fail in the process of star identification. A possible way to solve the above problem is to consider multiple combinations of the neighboring stars (or) selecting the brightest stars for finding the initial match in the SPD. However, the time complexity of the first approach will increase if such a solution is adopted.

3. Starting point (SP) mismatch - We have laid utmost emphasis on judiciously selecting the SP to construct the vector sequence for the third approach to star identification. To make sure that the SP will be present in the image, we select the SP as the brightest star (lowest M_v) amongst the four nearest stars to the reference star. However, there may arise a situation when a false star(s) is amongst the four nearest stars to the reference star. In a scenario, when the correct SP is probably the third or the fourth nearest star to the reference star in the SPD, and there are one or two false stars which lie closer to the reference star in the image captured, then the correct SP will cease to be present amongst the nearest four stars to the reference star in the image. Due to the above condition, the vector sequence constructed from the image will never match with the SPD. Thus, the third proposed approach will fail in this scenario. However, it is important to note that the probability of occurrence of such a situation is very rare as it has many conditions imposed on it.

4. A high number of both missing and false stars in an image when compared to the SPD - There are some star IDs in the star catalog which contain a less amount of neighboring stars in their FOV. These star IDs lie in the sparse region of the sky catalog as was shown in Fig 2.7. For these star IDs, if there are a lot of missing and false stars in the image captured, then the image contains sparse correct information. In such a scenario, though the proposed approaches of star identification will succeed to identify the reference star correctly in the image, however, the star shortlisting techniques will fail to shortlist the correct star ID from the SPD. The reason being the star shortlisting technique extracts only partial information from the image for comparing with the SPD. Thus, the proposed star shortlisting techniques will fail when there is a scenario of such sparse correct details in the image.

5. Stars at the same distance in the vector sequence - This is an unusual case scenario concerning the third proposed approach of star identification. While building the vector sequence, we consider the stars in increasing order of their distance from the reference star for making the vector sequence. While con-

structing the vector sequence of a star ID in the SPD, there may arise a problem when two neighboring stars lie at the same distance from the reference star (star ID). In such a scenario, we can select any of the two stars to be considered firstly in the travel path for constructing the vector sequence of the star ID in the SPD. However, when it comes to the image captured (having the same problem), we will build two different vector sequences by considering those two equidistant stars taking the first place and later the second place in the travel path. By adopting this methodology, it is obvious that one of the two vector sequences from the image will surely match with the vector sequence stored in the SPD of the reference star (star ID).

4.6 Summary

In this chapter, we proposed a novel framework for the process of star identification for an LIS mode star tracker. In the proposed structure, the first step is to shortlist a part of the SPD initially, and later to provide the shortlisted list of star IDs to the star pattern recognition. The ideology behind this proposed framework is that the shortlisting will lead to reducing the time complexity, and star pattern recognition technique will achieve high recognition reliability. Thus, the developed framework should lead to solving the problem of trade-off between attaining high recognition reliability and low time complexity. We explained three different approaches of star shortlisting and star pattern recognition for the proposed framework. The first proposed approach falls into the category of purely geometric based identification. The second proposed approach is a combination of the geometric and pattern based identification. The final method also falls into the geometrically based identification. We also described some unusual cases when the proposed approaches would face difficulty in correctly shortlisting and identifying stars. Having developed three different methods, we will test the performance of these approaches in the next chapter.

Chapter 5

Simulations and Real-time Testing

This chapter is concerned with the performance evaluation of the proposed approaches and is divided into two parts - simulation and real-time testing. We will begin with simulation setup and parameters selected for testing. Later, we will introduce distortions, magnitude uncertainty, and noise in the simulated image to resemble it to a real image captured. We will then compare and benchmark each of the proposed approaches with state-of-the-art star pattern recognition techniques on the simulated images. In the second part of this chapter, firstly, we will describe the hardware implementation of the proposed approaches and the development of the prototype Lost-in-space mode star tracker. Later, we will show the setting up of a real-time testing system for the star tracker. Finally, we will show the results of the developed prototype star tracker on the real-time testing system.

5.1 Simulation, Testing and Benchmarking

5.1.1 Simulation Setup

The initial testing of the proposed approaches is carried out on the simulated star images. The MATLAB software generates the simulated images. The sky map from SAO J2000 sky catalog was used for creating the synthetic images. The right ascension (RA) angle was varied from 0^0 to 360^0 and the declination (DEC) angle

from -90^0 to $+90^0$ in steps of two degrees. This setup creates 16200 star images covering the full sky scan. To emulate the real images, we chose the parameters of star tracker SST-20S for simulating the star images listed in Table 3.1.

At a particular right ascension (RA) and declination (DEC) angle, all the stars having a relative magnitude (M_v) of less than 6.3 and which lie within a FOV of 15^0 x 15^0 were considered to be captured in the simulated image. The dimensions of the simulated image were 1024 x 1024, and pixel size was taken as 13 μm x 13 μm. To make the simulated image closely resemble a real image, we also need to add positional deviation, magnitude uncertainty, and false stars in the simulated images. Thus, firstly an ideal image was considered for the testing the performance. Later, the robustness was tested for those mentioned above three different scenarios. The setup and the parameters adopted for adding the positional deviation, magnitude uncertainty, and false stars will be discussed in their respective sections later.

5.1.2 Testing and Benchmarking

State-of-the-art star pattern recognition techniques which claim to provide high recognition reliability and high speed of recognition were selected to be benchmarked with the proposed approaches. We have selected a geometric voting algorithm(GMV) [32], search tree with optimized database (STOD) [15], shortest distance transform (SDT) [43], optimized grid algorithm (OPTG) [57], Polestar [58], and star identification using pattern code (SPC) [59] for benchmarking with our proposed approaches. All the above methods select the star nearest to the center of the image as the reference star to be identified. Except for the SDT technique, all the techniques mentioned above do not utilize the ambiguous intensity of the stars in the image captured as a feature for comparison purposes. Thus, the basic scheme adopted by these techniques complies with that of our proposed technique. Moreover, the OPTG technique is implemented using the radiometric clustering (selecting the particular pivot stars) and optimum grid size with the vetoing process which gives the best performance as described in their paper.

STOD [15] and SPC [59] claim to attain a high speed of identification. SDT [43]

and Polestar [58] claim to attain a high identification accuracy. OPTG [57] and GMV [32] claim to maintain a high recognition reliability. In this section, firstly we will show the results of the benchmarking in an ideal case simulated image. We will use the parameters of identification accuracy, run-time, and SPD size for comparison purposes in the ideal case. Later, we will test the robustness of the techniques by challenging them to different scenarios of positional deviation, magnitude uncertainty, and false stars in the simulated image. The recognition reliability of the techniques will be a measure of the robustness of these techniques. We will discuss the simulation setup adopted by us for each of the scenarios and analogize it with the suggested simulation parameters of the benchmarked techniques. Moreover, we will also provide the reasoning behind the performance of all the techniques in each of the scenario.

Ideal Case

The star pattern recognition techniques were firstly tested on ideal simulated images. The ideal images do not contain any noise, magnitude uncertainty or positional deviation of the stars. However, the problem of *patch mismatch* is inherently present in the image as the images are generated at an arbitrary value of RA and DEC. Thus, the image produced will not coincide precisely with the SPD of a particular star ID. Some stars will be missed in the image because of this scenario of patch mismatch.

The performance of the star pattern recognition techniques in the ideal case is evaluated based on three parameters - identification accuracy, run-time, and the SPD size. The identification accuracy is measured as the number of correctly identified images out of the total number of images (16200). An image is marked as correctly identified if the reference star and the nearest two neighboring stars to the reference star in the image are correctly identified. The run-time of the techniques is measured on the MATLAB software (with PC specifications: Intel i7 processor, 3.4 GHz, 8 GB RAM) as the time consumed after the star positions are determined (centroiding process) until the technique returns the star IDs of the three stars in the image. This concept of measuring the run-time is adopted from the STOD [15]. The SPD

size is the size occupied by the database of the proposed technique along with star catalog. It is desired to have a lower SPD size so that the resources required for the onboard memory are reduced. It was a critical parameter in the past decade when the onboard memory was insufficient. However, current day onboard computers do not have a problem with storing up to 5 MB. Out of the above three parameters, the first two are more significant compared to the SPD size. So, SPD size (if less than 5 MB) is not as important as the identification accuracy and the run-time of a star pattern recognition technique. Table 5.1 shows the performance of the proposed technique and the benchmarking results in an ideal case.

Technique	**Identification accuracy (%)**	**Run-time (sec)**	**SPD size (MB)**
GMV [32]	94.3	0.26	1.31
STOD [15]	92.6	0.23	1.85
SDT [43]	95.78	0.337	4.7
OPTG [57]	98.6	0.28	1.3
Polestar [58]	97.86	0.386	0.9
SPC [59]	95.8	0.25	1.85
Proposed approach 1[1]	**99.57**	**0.22**	**1.7**
Proposed approach 2[2]	**99.7**	**0.112**	**1.21**
Proposed approach 3[3]	**99.86**	**0.13**	**1.4**

[1]Running sequential angular match technique
[2]Image to signal approach
[3]Rotation-invariant vector frame

Table 5.1: Benchmarking of star pattern recognition techniques - Ideal case.

As can be seen from the performance results in Table 5.1, the proposed approaches outperforms all the other techniques in terms of identification accuracy and run-time. We will now explain the reasoning behind the performance of the star pattern recognition techniques.

- GMV - GMV considers voting of the radial distances of the neighboring stars

for the identification of a reference star. Due to a simple voting process, GMV can achieve a decent identification accuracy with a reasonable run-time.

- STOD - STOD is the fastest of all the existing state-of-the-art star pattern recognition techniques. This high speed of identification is possible due to the parallel search in the tree-like structure SPD of the STOD. However, STOD suffers in terms of identification accuracy at the same time.
- SDT - SDT utilizes the distance as well as the brightness of the stars for identification. Thus, it can achieve a higher identification accuracy compared to the GMV and STOD in an ideal case scenario. However, it has a higher time complexity compared to both GMV and STOD.
- OPTG - OPTG has various steps comprising of selecting the pivot star, optimizing the grid size, and the vetoing process. The above steps are responsible for the high identification accuracy and a high run-time attained by OPTG.
- Polestar - Polestar algorithm constructs binary coded string from the distances of the neighboring stars. This bit-code string is quite robust in an ideal case scenario. Thus, it achieves a decent identification accuracy. However, it requires comparatively higher time to build the binary coded string.
- SPC - SPC constructs a rotation-invariant code from the distances and angles of the stars. Thus, SPC achieves a decent identification accuracy and consumes a reasonable run-time in an ideal case scenario.
- Proposed approach 1 - The first proposed approach achieves a high identification accuracy compared to the benchmarked techniques because it employs a perfect match of 360 degrees or 2π radians. Due to the shortlisting of the stars based on the worst case patch mismatch, this approach can achieve comparable run-time to the fast techniques such as STOD and SPC.
- Proposed approach 2 - The star pattern recognition approach based on spearman-correlation achieves a high identification accuracy compared to most of the compared techniques. This approach outperforms both STOD and SPC (fastest

state-of-the-art techniques), as well as it is the fastest of all the three proposed methods because of the quick shortlisting achieved by hamming distance comparison.

- Proposed approach 3 - The rotation-invariant additive vector sequence approach achieves a higher identification accuracy of 99.86%, as the problem of patch mismatch cannot affect the matching between the vector sequences of the image and the SPD. The run-time of this approach is also almost half of that of SPC. The reason being apart from the simple vector sequence matching process of the proposed technique, many star IDs do not qualify for the starting point (SP) match itself. This situation of SP not being matched eliminates the further requirement to check the matching of the vector sequence in many of the star IDs in the SPD.

Positional Deviation of Stars

The spacecraft usually undergoes unexpected velocity changes and acceleration. It also experiences vibrations and disturbances throughout the period in its orbit. Due to this, the image captured by the star tracker is distorted. This leads to deviation in position of stars in the image, which further leads to an error in features extracted from the image. To simulate this scenario, we introduce positional deviation (also known as positional noise) of stars in the simulated images. In our simulations, the positional deviation is introduced in the images from 0.1 pixels to 1.0 pixels in the steps of 0.1. This introduction of the pixel positional deviation is equivalent to the addition of random Gaussian noise to the projected locations of the stars in the image (as described in OPTG [57], and Polestar [58]). In Polestar [58], the authors added the positional deviation with a nominal standard deviation (σ) of 0.062 pixels and a maximum of 0.31 pixels. In OPTG, the positional deviation was varied from 0 (ideal case) to 3 pixels in steps of 0.5 pixels. Analogous to the OPTG approach, we also vary the positional deviation from 0 (ideal case) to 3 pixels in steps of 0.5 pixels. The robustness of the star pattern recognition techniques is measured in terms of the identification accuracy of these techniques in such a scenario. The

performance of the techniques in case of positional deviation is shown in Fig 5.1 and Fig 5.2. Geometric distance between the stars is the most affected feature due to the positional deviation of stars. We will now explain the performance results in Fig 5.1 and Fig 5.2 for each star pattern recognition technique.

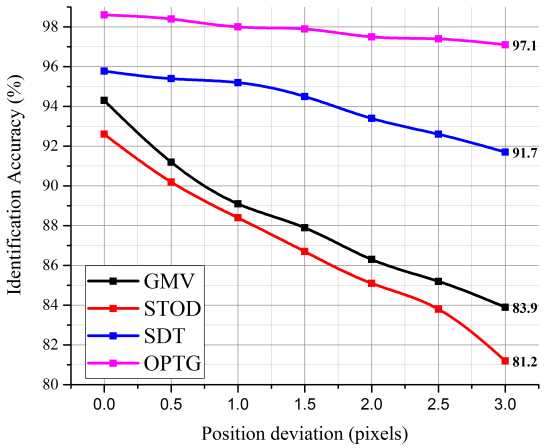


Figure 5.1: Performance of star pattern recognition techniques in the scenario of position deviation (1).

- GMV - GMV utilizes voting of the distances as the only parameter for comparison. As the geometric distance of the stars is the most affected feature due to positional deviation, it fails drastically to identify the stars in this scenario correctly.
- STOD - STOD has distances stored as the features in the nodes of the tree built as the SPD. As it relies upon only the distance to match the stars, STOD also fails in the scenario of positional deviation.
- SDT - SDT does not rely upon the distance as the only feature for comparison. It utilizes the brightness of the stars as an additional feature for comparison and matching. Thus, SDT is quite robust to the distortions caused in the image.

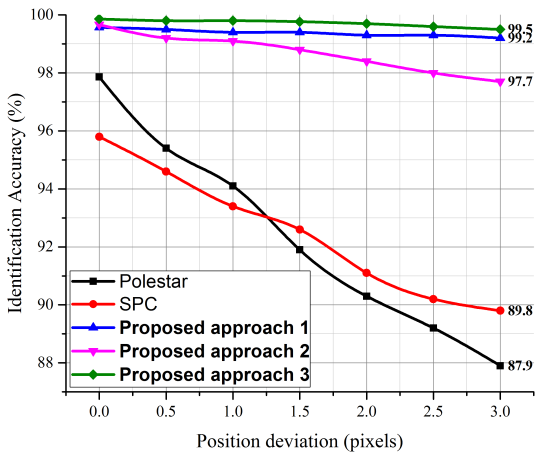


Figure 5.2: Performance of star pattern recognition techniques in the scenario of position deviation (2).

- OPTG - OPTG, unlike the original grid algorithm, constructs an optimum grid size. This optimum grid size makes it robust to the case of positional deviation, and it is thus able to maintain the identification accuracy to 97.1% when the positional deviation is as high as 3 pixels.

- Polestar - Polestar technique fails drastically to recognize stars in this scenario because it utilizes a coded binary string constructed from the geometric distance feature.

- SPC - SPC generates a code based on the distances as well as the angles of the neighboring stars. Thus, it is more robust compared to the STOD and GMV, but still, the code generated from the image differs when compared to the SPD.

- Proposed approach 1 - The first proposed approach utilizes relative angles along with the distance. Moreover, the running sequential angular match of 2π radians makes it robust to the case of positional deviation.

- Proposed approach 2 - The second proposed approach utilizes both the distance and relative angle between the stars as the features extracted. Data points in

the signal constructed do not undergo a high shift in position because the relative angles are used as the time domain in our proposed technique. Thus, the spearman correlation coefficient calculated does not change considerably.

- Proposed approach 3 - In the case of the third proposed approach, the vector sequence is generated by the dot product of the transpose of the added vector and the normalized vector. As the multiplication involves a normalized vector, this dot product is not affected much by the deviation in star positions. Hence the proposed star pattern recognition technique maintains a high identification accuracy in this scenario.

Magnitude Uncertainty

As described earlier in chapter 3, magnitude uncertainty of the stars captured in space is one of the major problems faced by the star pattern recognition techniques. Some low magnitude stars are missed to be captured in the image because of the magnitude uncertainty. This scenario eventually adds more missing stars to the already missed stars due to the patch mismatch. To simulate such a scenario, we add relative magnitude (M_v) uncertainty in the simulated image. This process is equivalent to the approach adopted in OPTG [57] and Polestar [58], where they add Gaussian noise (with a particular standard deviation) to the magnitude of the stars. In Polestar [58], the nominal standard deviation (σ) for the added Gaussian noise is 0.323 M_v, and the maximum value is 0.9 M_v with the magnitude threshold of 7.0 M_v. In OPTG [57], the authors introduced stellar magnitude standard deviation of 0.4. In our simulations, as our magnitude threshold is 6.3 M_v, we vary the magnitude uncertainty till 0.4 M_v, which is enough for performance evaluation. For example, an M_v uncertainty of 0.2 would suggest stars having M_v greater than 6.1 will be missed (normal magnitude threshold being 6.3 M_v). If the variation is further increased, the number of stars in the image will be less than 5, which would not correspond to a real image. The recognition reliability of the star pattern recognition techniques for the case of magnitude uncertainty is shown in Fig 5.3 and Fig 5.4. We will now explain the performance results in Fig 5.3 and Fig 5.4 for each star pattern

recognition technique.

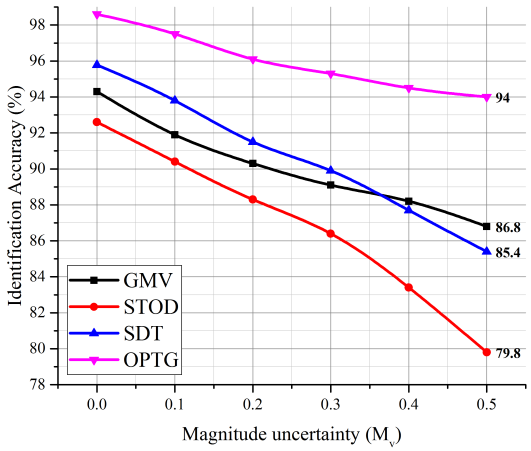


Figure 5.3: Performance of star pattern recognition techniques in the scenario of magnitude uncertainty (1).

- GMV - GMV counts the votes of the matched radial distances of the neighboring stars (in the FOV of the reference star). This approach leads to failure of the technique, as the low brightness neighboring stars are missed in addition to the patch mismatch missed stars.
- STOD - STOD compares the number of neighboring stars (in the FOV of the reference star) as the parameter for recognition. The number of neighboring stars is stored as the head node in the tree SPD of STOD. Thus, it fails drastically too in the scenario of magnitude uncertainty.
- SDT - SDT is a technique which utilizes the brightness of the stars as one of the features for comparison. Thus, SDT fails (identification accuracy falls to 85%) in this scenario as brightness is not a reliable feature in the case of magnitude uncertainty.

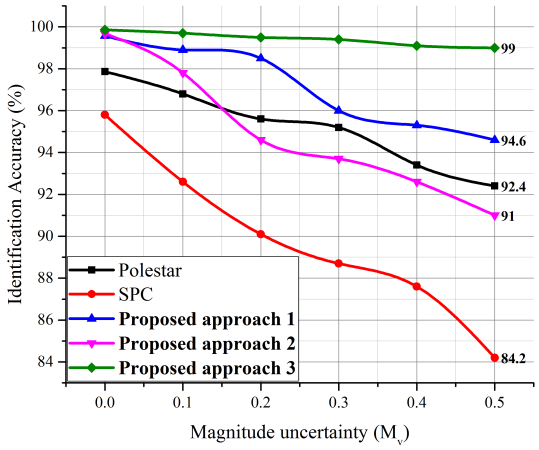


Figure 5.4: Performance of star pattern recognition techniques in the scenario of magnitude uncertainty (2).

- OPTG - OPTG maintains a decent identification accuracy of 94% in this case as it selects the pivot star judiciously. It considers the various combination of the neighboring stars and selects the appropriate one as the pivot star.

- Polestar - Polestar maintains decent recognition reliability in this case because it does not utilize the brightness of the stars as a feature for comparison and it also utilizes the various combination of neighboring stars in the FOV of the reference star for feature construction.

- SPC - The identification accuracy of the SPC is also reduced to 84% as the code generated from the image containing many missing stars differs a lot when compared to the code stored in the SPD.

- - Proposed approach 1 - The first proposed approach maintains high robustness (95%) in this case due to the running sequential angular match technique as was explained in Fig 4.8.

- Proposed approach 2 - The second proposed approach maintains reasonable recognition reliability (91%) because of the exact 512-sample signal construc-

tion and spearman correlation. The functioning of our proposed technique in this scenario was also explained in subsection 4.3.2 of chapter 4 of this thesis.

- Proposed approach 3 - The third proposed approach takes care of matching the vector sequence between the image and the SPD even when there are missing stars as was explained in the scenario if missing stars in subsection 4.4.3 of chapter 4 of this thesis. Hence the proposed technique maintains a high robustness in the case of magnitude uncertainty.

False Stars

Reflection from planets, meteors and space debris is often identified as a false star in the image captured. These false stars are not present in the SPD and thus challenge the star identification algorithms when they match the image with the SPD. To simulate such a scenario, we add false stars with random brightness and position in the simulated image. The number of false stars added to an image is varied from 0 (ideal case) to 5. This approach is adopted from the one described by the authors in STOD [15]. Fig 5.5 and Fig 5.6 show the performance of the star pattern recognition techniques in the case of false stars. We will now explain the performance results in Fig 5.5 and Fig 5.6 for each star pattern recognition technique.

- GMV - GMV fails for the same reason as it did during the scenario of magnitude uncertainty.
- STOD - STOD also fails for the same reason as it during the scenario of magnitude uncertainty.
- SDT - SDT maintains a decent identification accuracy in this scenario as it considers a combination of brightness and distance along with a different number of stars.
- OPTG - OPTG also maintains a decent identification accuracy in this scenario mainly as it has a vetoing process which discards the false matches.

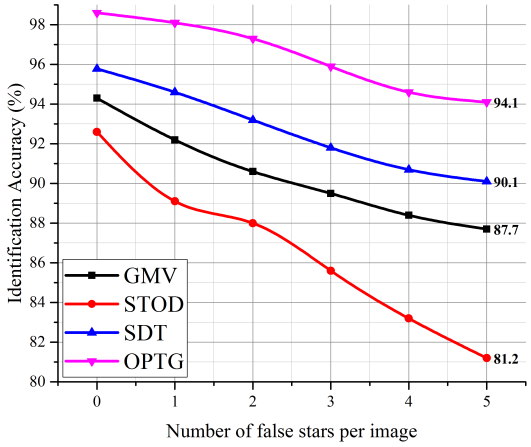


Figure 5.5: Performance of star pattern recognition techniques in the scenario of false stars (1).

- Polestar - Polestar's recognition reliability drops but not as drastically as STOD and GMV, because it takes into consideration various combinations of the neighboring stars in the FOV for feature extraction.

- SPC - SPC fails for the same reason as during the scenario of magnitude uncertainty.

- - Proposed approach 1 - Robustness of the first proposed approach, in this case, is due to the running sequential angular match, which was also explained in the example in Fig 4.8.

- Proposed approach 2 - The reason for the high recognition reliability of the second proposed approach for this case is the same as that for the magnitude uncertainty of the stars.

- Proposed approach 3 - The success of the third proposed approach, in this case, was explained earlier in the scenario of false star present in the images in subsection 4.4.3 of chapter 4 of this thesis.

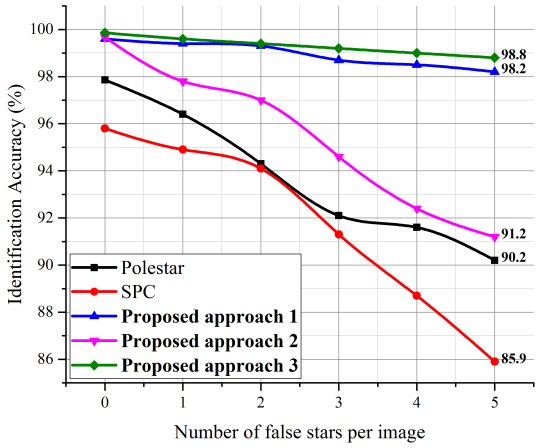


Figure 5.6: Performance of star pattern recognition techniques in the scenario of false stars (2).

Performance of Shortlisting Techniques

We will now analyze the performance of the proposed shortlisting techniques separately. The primary purpose of the proposed shortlisting technique is reducing the overall time required for identification of the star IDs. It accomplishes this task by giving a shortlisted list of star IDs for pattern recognition. The performance of the shortlisting technique is measured in terms of the number of shortlisted entries (N_{sh}) out of the total number of star IDs in the SPD, and the probability that the shortlisted entries contain the correct star ID (p_{sh}). Every shortlisting technique will be analyzed based on the two parameters mentioned above.

- Worst case patch mismatch based star shortlisting - For the first proposed shortlisting technique, recall that there are slabs (or) windows in the LUT. Thus, for this shortlisting technique, the shortlisted entries will vary for every slab. Hence the performance of the first shortlisting technique is evaluated for every slab. For testing this shortlisting technique, we have adopted the relative magnitude (M_v) threshold of 6.0 in the SAO J2000 star catalog, which corresponds to 4956-star entries in the SPD. In Fig 5.7, we display the performance

of the first shortlisting technique.

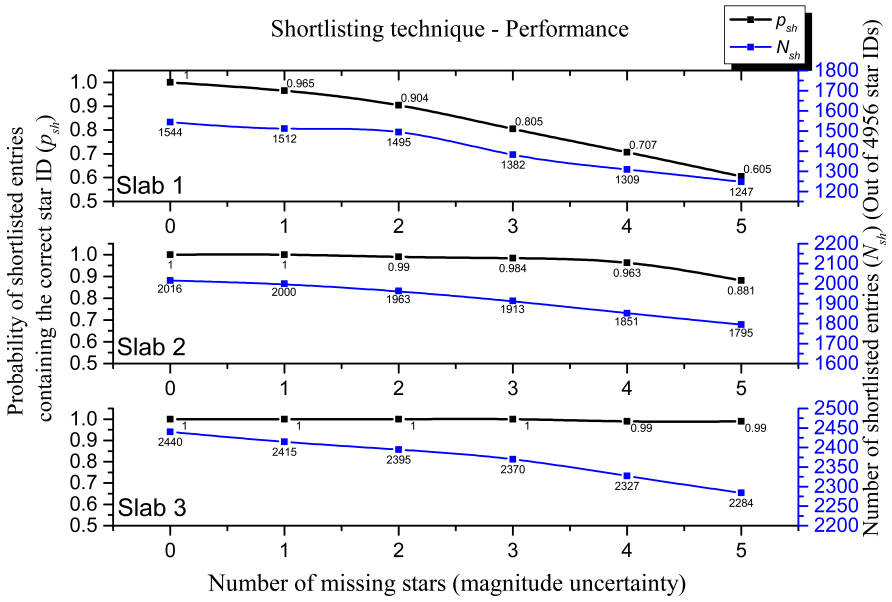


Figure 5.7: Performance of the proposed first star shortlisting technique.

As can be seen from Fig 5.7, the performance for Slab 1 gives the least N_{sh} followed by Slab 2 and later Slab 3. The above result is because the window for Slab 1 is narrow compared to a wider window for Slab 2 and later Slab 3. Thus, the p_{sh} is highest in Slab 3 compared to Slab 2 and later Slab 1. In Fig 5.7, the number of missing stars is due to the magnitude uncertainty in addition to the patch mismatch. Hence, the p_{sh} of the Slab 1 decreases drastically when the number of missing stars increases. Whereas for Slab 3, which has a wider window for shortlisting, the p_{sh} does not fall that drastically. This robustness of p_{sh} for Slab 3 is because the lower threshold of Slab 3 is calculated by missing four more stars adding to the worst case patch mismatch. Whereas, for Slab 1, the lower threshold is calculated by missing only the worst case patch mismatch number of stars. It is evident from the results in Fig 5.7 that the shortlisting technique provides only one-third (1544 / 4956) of the SPD to the pattern

recognition technique in case of Slab 1 and a worst case of half (2440 / 4956) of the SPD in the case of Slab 3. This star shortlisting makes the combination of the worst case patch mismatch and running sequential angular match technique (Proposed approach 1) fast as was depicted in the results shown in Table 5.1.

- Star shortlisting based on hamming distance - The performance for the hamming distance based star ID shortlisting technique, i.e. the second proposed approach of the shortlisting technique was described in the section 4.3.3 of chapter 4 of this thesis. We provided the reasoning behind selecting the 32 samples for the shortlisting technique. The selection was based on the performance of the shortlisting technique for a different number of samples. For a sample size of 32, the shortlisting technique provided a p_{sh} of 1.0 with N_{sh} of almost one-third of the size of the SPD.

- Star shortlisting based on rotation-invariant vector frame - As mentioned earlier in subsection 4.4.2 in chapter 4 of this thesis, the range for the shortlisting technique (based on the rotation-invariant vector frame) is dependent upon the performance of the shortlisting technique. Here we will test the performance and decide on the range of the star IDs to be shortlisted for the proposed third shortlisting technique based on the rotation-invariant vector frame. We choose the star catalog SAO J2000 for simulating the star image, and an apparent magnitude threshold (M_v) of 6.3 for testing this star shortlisting technique. Thus, there are a total number of 6903 star entries in the SPD. We simulate the star images with a positional deviation of 3 pixels and have some missing and false stars in the image as shown in Fig 5.8.

  Fig 5.8 shows the results of the performance of the proposed third shortlisting technique. In Fig 5.8, the *x-axis* represents the number of low magnitude missing stars and false stars in each image. As mentioned earlier, the performance of a shortlisting technique is evaluated based on the probability of correct star ID shortlisted (p_{sh}), and the number of shortlisted entries (N_{sh}). Concerning the N_{sh}, we have varied it from top 500 voted star IDs to top 2500 voted star IDs in the steps of 500. For each category, we measure their corresponding

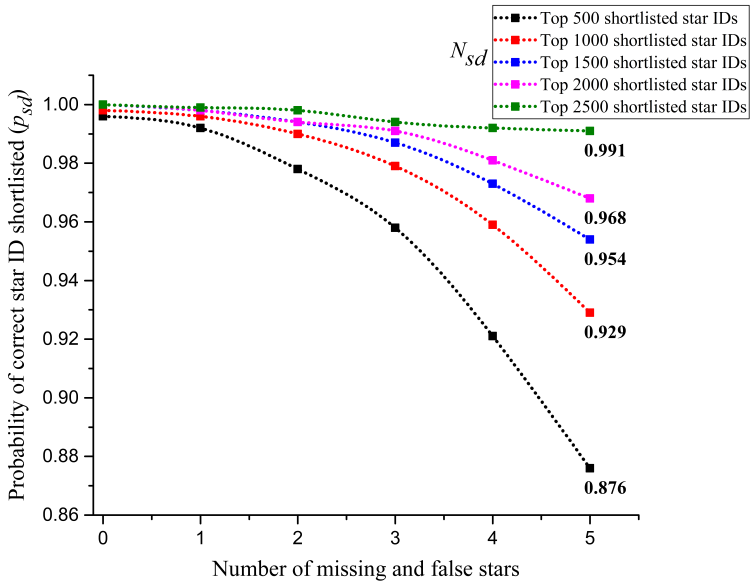


Figure 5.8: Performance of the proposed third star shortlisting technique.

p_{sh}, which is represented on the *x-axis* in Fig 5.8. It can be seen from Fig 5.8 that as the number of missing stars and false stars in the image increases, the p_{sh} decreases. This decrease of p_{sh} is drastic in the case if we only shortlist the top 500 voted star IDs, as it reduces to mere 0.876 when there are five missing and false stars in the image captured. As the N_{sh} considered goes on increasing, the corresponding p_{sh} also shows an increase. From the results in Fig 5.8, we can infer that for the N_{sh} of 2500, the corresponding p_{sh} is still maintained to as high as 0.99 even when there are 5 missing and false stars present in the simulated image. Thus, we select the top 2500 voted star IDs as the particular range for shortlisting. Hence the number of shortlisted entries is almost reduced to one-third of the total number of entries (6903) in the SPD.

5.2 Real-time Testing

Having tested the proposed approaches on simulated images, the next step is to check the real-time testing of the proposed methods. For this purpose, we design

a real-time state-of-the-art star tracking testing system. We test the Lost-in-space mode star sensor based on the proposed approaches on the testing system. In this section, we will first describe the testing setup and configuration. Later, we will test and compare all the three developed approaches on the real images captured by the star tracker SST-20S.

5.2.1 Testing Setup

The testing setup for an LIS mode star tracker is commonly known as the hardware-in-loop test of the star sensor. The setup configuration and testing procedure were adopted from the research paper by G. Rufino *et al.* [97] and Wei *et al.* [98] which deal with the star tracker algorithm validation and hardware-in-loop test configuration. Fig 5.9 shows the schematic of the testing setup. The setup consists of the following parts:

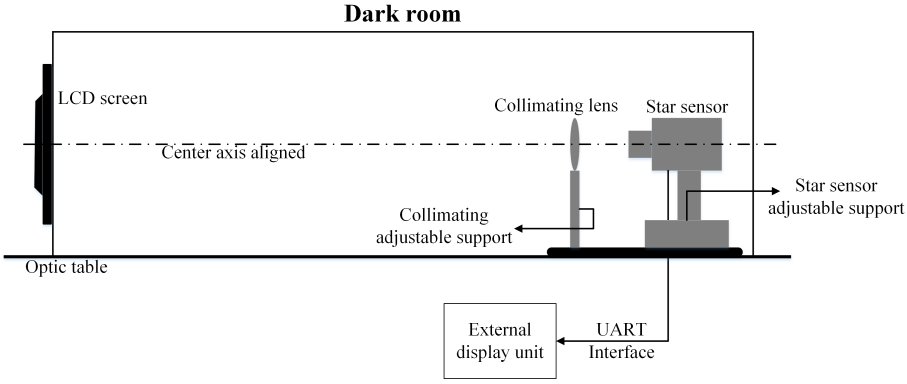


Figure 5.9: Schematic of hardware-in-loop test for the developed Lost-in-space mode star tracker.

1. LCD screen - The LCD screen is used for displaying the real star images (captured from the SST-20S tracker) in a loop. The speed of the display of images on the LCD screen is computer-controlled.

2. Collimating lens - A collimating lens makes the object to appear as if it is present at an infinite distance. Hence the collimating lens is placed in front

of the star sensor so that the stars in the image captured by the star tracker appear to be present at an infinite distance. The collimating lens is positioned appropriately so that the FOV of the star tracker covers the whole LCD screen. Moreover, we have made sure that the center axis of the lens coincides with the center of the star sensor and the LCD screen as shown in Fig 5.9. A vertical and horizontal adjustable support of the collimating lens is needed for the above two purposes.

3. Star sensor - We have developed an LIS mode star sensor as shown in Fig 5.10(a). It consists of a lens, an image sensor, and an FPGA board shown in Fig 5.10(b). The lens is mounted on the sensor with a C-mount type. The lens and image sensor characteristics are replicated to be the same as used for the SST-20S star tracker. The star sensor can also be rotated and translated by the adjustable screws to coincide exactly with the collimating lens axis. The FPGA board is a miniaturized replica of the Zynq 7000 Zedboard [99] which contains a single-core ARM Cortex A9 processor (650 MHz). Any of the proposed approach is selected and then embedded in C in the ARM processor of the FPGA board. The SPD of the selected approach is stored in the onboard SD card of the FPGA board. The image sensor captures the real star image displayed on the LCD screen and the raw data of the image is then fed to the FPGA board for processing. The selected approach is then applied to the raw data to locate the star positions (centroiding process), implement the proposed star identification technique (identifying the stars in the image), and finally calculate the attitude by the QUEST method (attitude determination). The positions and star IDs of the stars identified in the image and the attitude calculated is then passed to the external display unit by a UART interface.

4. External display unit - The purpose of the external display unit is to reproduce and display the image captured by the star sensor along with the star IDs of the identified stars, run-time, and the calculated attitude. This task is achieved by implementing a simple display code in Python which displays all the above information received via the UART interface from the FPGA board.

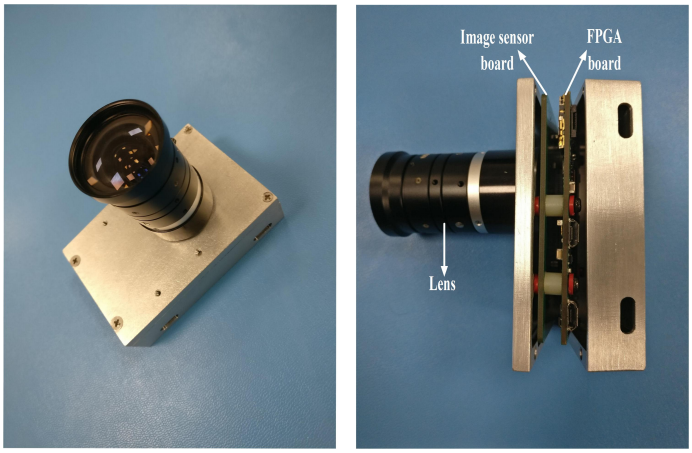


Figure 5.10: (a) LIS mode star sensor. (b) Components of star sensor.

5. Optic table - The parts 1), 2), and 3) namely the LCD screen, the collimating lens, and the star sensor are fixed with their center axis aligned on the optic table for the testing purpose.

6. Dark room - The entire setup is positioned in a state-of-the-art dark room that prevents any external light source so that the testing process is not compromised.

Fig 5.11 shows the configured real-time testing setup.

5.2.2 Real-time Testing Results

We will now show the testing results of the developed star sensor based on the proposed approaches on the real images. For a real star image displayed on the LCD screen, the above-described flow of events takes place. The attitude calculated based on the stars identified by the selected proposed technique is compared with the attitude obtained from the SST-20S star tracker data. If the difference between the calculated and the received values of attitude is less than 40 arc seconds (0.01^0), then the star identification is counted as a success. If the difference is larger than 40 arc seconds, then it is counted as a failure.

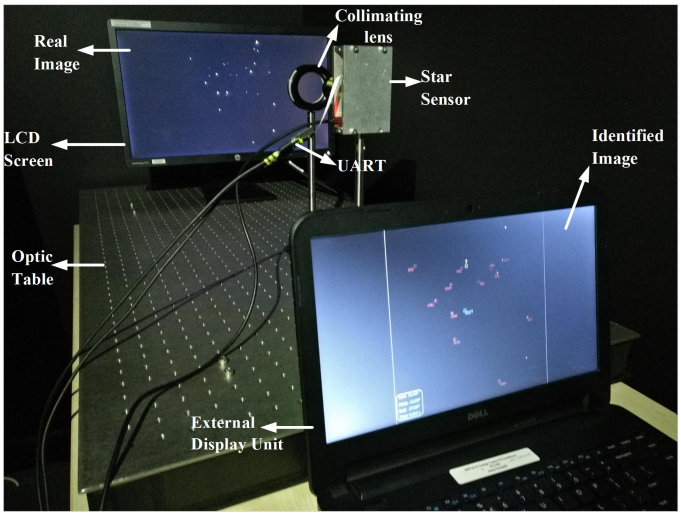


Figure 5.11: Real-time testing setup for the LIS mode star sensor.

- Proposed approach 1 - The proposed approach of running sequential angular match based star pattern recognition was implemented on 3104 real star images. It was able to identify 4648 images correctly out of the 4817 images, thus giving an identification accuracy of *96.49%*. The technique failed to identify 2 out of 102 sets. The reason for the failure was because it was unable to find an initial match in these two sets of images. The proposed technique took 97 *ms* on an average to identify the stars of which reading the image data, pre-processing and centroiding operation took 73 *ms*. Implementation of the proposed idea and to output the identified star IDs occupied only 14 *ms*, and calculating the attitude took 10 *ms*. Fig 5.12 shows one of the identified real images by the proposed technique where the center of the image is marked with purple (I_c). The star nearest to I_c i.e., the reference star is marked with orange and is identified by the star ID 4908 from the SPD. From the sequence of the star ID 4908 stored in the SPD, the rest of the identified neighboring stars are marked with yellow.

- Proposed approach 2 - The proposed approach of hamming distance based shortlisting and spearman-correlation based star identification was implemented on 4817 real star images. The proposed technique correctly identified 4499 out

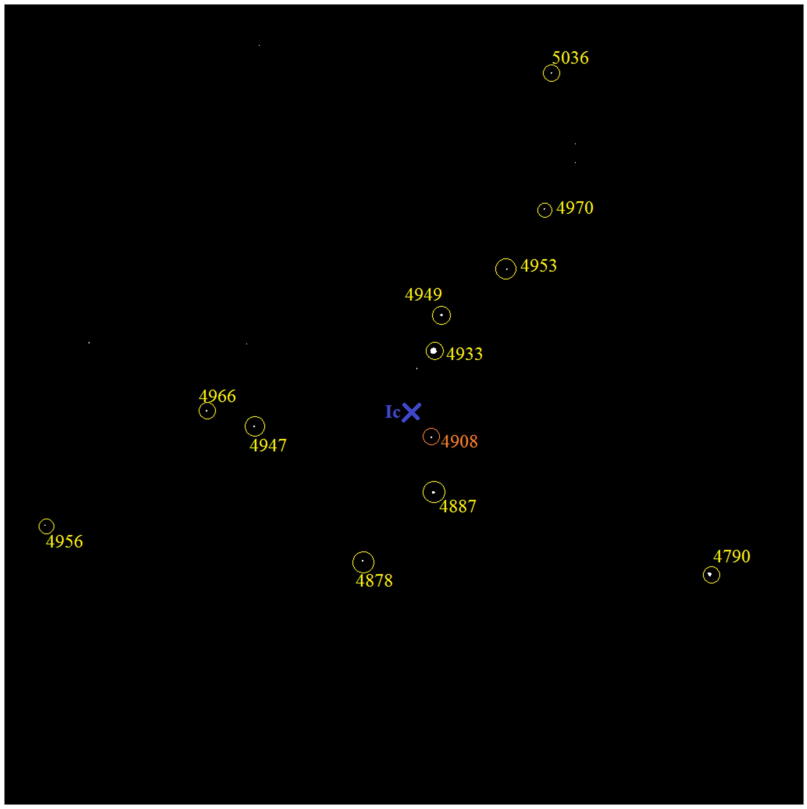


Figure 5.12: An identified real image by the first proposed approach.

of 4817 real images, thus, achieving an identification accuracy of *93.39%*. The method failed to identify 3 out of 100 sets of images. The reason behind the failure of the technique was the maximum spearman-correlation co-efficient obtained for all these three sets was less than 0.5, thus, suggesting that the match between the signal acquired from the image and the signal stored in the SPD of the shortlisted entries is very low and cannot be trusted. The proposed technique took only 93 *ms* on an average to identify the stars of which reading the image data acquired, pre-processing and centroiding operation occupied 73 *ms*. Implementation of the proposed idea for identifying the stars took only 11 *ms*, and calculating the attitude consumed 9 *ms*.

In Fig 5.13, we show one of the identified real images by the proposed technique. The center of the image is marked with a cross (I_c). The star nearest to the

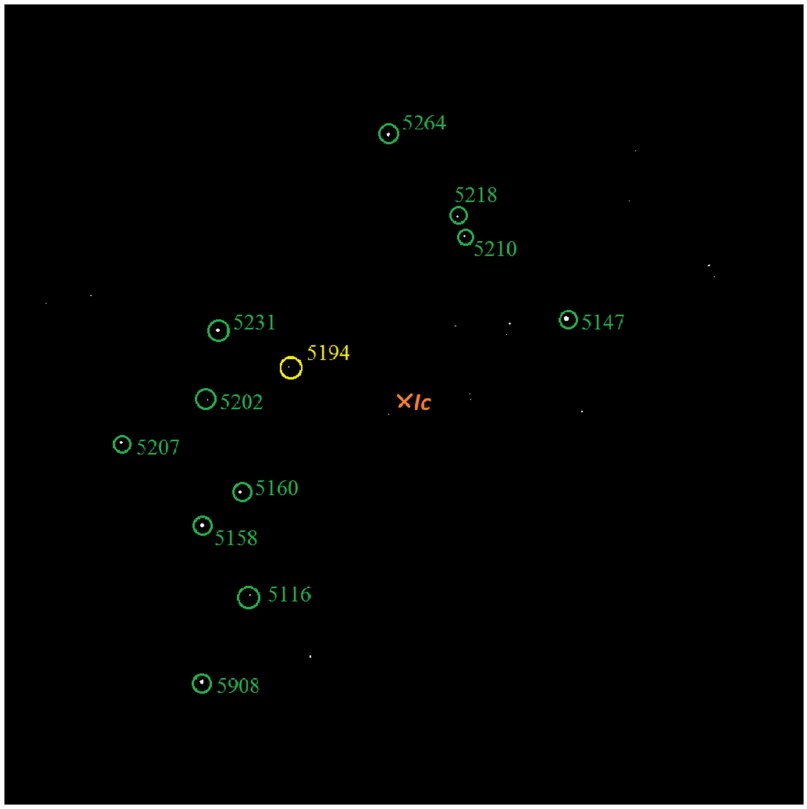


Figure 5.13: An identified real image by the second proposed approach.

center of the image is selected as the reference star to be identified and is marked with yellow. This reference star is identified as 5194 from the SPD and the spearman correlation achieved for this image is 0.79. From the SPD sequence of the reference star ID 5194, rest of the stars in the image identified are marked with green.

- Proposed approach 3 - The proposed approach of rotation-invariant vector frame was implemented on 4817 real images. It was successful in identifying 4756 real images correctly out of the 4817 images tested, thus achieving an identification accuracy of *98.7%*. The reason for the failure of the proposed technique on the 61 images was the mismatch of the starting point between the image (SP being a false star) and the SPD. The overall time taken was

on an average 95 *ms* for the whole process. This time includes the time taken by the centroiding process (74 *ms*), proposed star pattern recognition (12 *ms*), and the calculation of the attitude (9 *ms*).

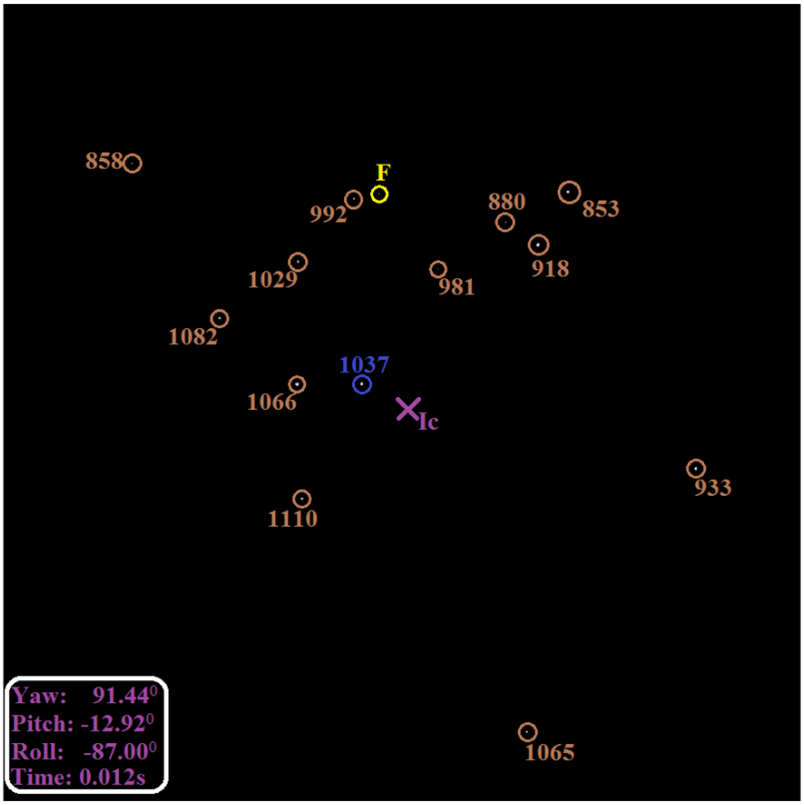


Figure 5.14: An identified real image by the third proposed approach.

Fig 5.14 shows one of the identified real images by this approach. In Fig 5.14, the center of the image is marked with a purple cross and labeled as I_c. The star nearest to the center of the image is marked with a dark blue circle and is identified as 1037 star ID from the SPD. The rest of the stars in the image are marked with the brown circle and are labeled with their corresponding identified star IDs. The attitude calculated is represented in terms of yaw, pitch, and roll angles along with the run-time of the proposed technique on the bottom left of Fig 5.14.

5.3 Summary

This chapter dealt with testing the performance of the proposed approaches and benchmarking them with the state-of-the-art star pattern recognition techniques. The testing is performed both on simulated images in different scenarios as well as on the real star images captured by the star tracker SST-20S. The performance and benchmarking results of the proposed approaches on the simulated images suggest that the proposed framework and the developed star shortlisting and pattern recognition techniques are highly robust and fast at the same time. These results show that the proposed approaches overcome the disadvantage of the existing state-of-the-art star pattern recognition techniques which are either highly recognition reliable or fast. The performance of the proposed methods on the real images justify the results achieved on the simulated images, thus suggesting that the proposed methods can be realized in real-time and are suitable for actual space applications.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, we started by realizing the importance of building a reliable and autonomous attitude determination system for satellite missions. We reviewed several existing attitude sensors and concluded that star trackers provide highly accurate attitude information compared to the existing attitude sensors. Thus, satellite missions which require attitude accuracy in terms of arc seconds should adopt star trackers as the attitude determination system for their purpose.

We presented the fundamental working principle of a star tracker system and the two modes of operation - Lost-in-space (LIS) mode and tracking mode. We realized that Lost-in-space mode of a star tracker is crucial compared to the tracking mode as the initial attitude information needs to be established in this mode. Later, we deeply analyzed the working of the star tracker in the LIS mode and concluded that the process of star pattern recognition forms the most critical part of the LIS mode of the star tracker.

Concerning the process of star pattern recognition, we aimed first to identify and analyze the problems associated with it. So, we performed a study on 1019 real images captured by a star tracker SST-20S currently mounted on the satellite VELOX-CI. We identified the problems of patch mismatch, magnitude uncertainty,

and false stars associated with the process of star pattern recognition. Moreover, we also provided a quantitative analysis of the problems mentioned above, which helped us realize the gravity of these problems. We also reviewed the state-of-the-art star pattern recognition techniques that have been developed in the past two decades for attaining reliable and fast star identification. We analyzed some selected star pattern recognition techniques in detail and understood their approach to solving the problem of star pattern recognition. After having studied these techniques in deep, we concluded that the parameters of recognition reliability and speed are of prime importance for a star pattern recognition technique.

Having understood the problems faced in star pattern recognition, and analyzed the existing state-of-the-art star pattern recognition techniques, we proposed three novel approaches for attaining high recognition reliability and fast star identification. Firstly, we built a generalized framework, which forms the basis for all the three methods. The framework proposes a novel two-step approach for achieving reliable and fast star identification. The star shortlisting technique constitutes the first step of the framework which is responsible for providing a list of shortlisted star IDs from the star pattern database (SPD). This step leads to the reduction in time complexity of the technique, as the star pattern recognition technique need not check and compare the whole SPD for matching with the image. It only operates on the shortlisted list of the star IDs provided, which helps in reduction of time complexity. The second step of the framework is the proposed star pattern recognition techniques which consider every star in the field of view of the reference star for feature extraction and comparison. This makes the star pattern recognition technique extremely recognition reliable and robust to the problems faced in the process of star pattern recognition.

The first proposed approach is a kind of geometric-based approach which extracts the radial distances of the neighboring stars and the relative angles that the consecutive neighboring stars make with the reference star for comparison and forming the SPD. Only radial distances are utilized for constructing the LUT, which constitutes the part of the shortlisting technique based on the worst-case patch mismatch. Both the radial distances and relative angles are used for star pattern recognition. The star pattern recognition technique further comprises of an initial match followed by

a running sequential angular match, which gives a perfect match of running the cycle for 360^0 or 2π radians.

The second proposed approach is a combination of geometric and pattern-based approach, which constructs a signal from the features extracted from the star image. The basic idea is to visualize the relative angles in the time domain, and the radial distances as the amplitude of the signal. Thus, a discrete point signal is constructed from the features extracted from the image. The signal is then divided into bins, and a 32-sample signal is stored in the (LUT) for shortlisting, and a 512-sample signal is stored in the SPD for star pattern recognition. The shortlisting of the star IDs is accomplished by comparing the hamming distance between the signal constructed from the image those stored in the LUT. The process of star pattern recognition is completed by calculating the spearman-correlation co-efficient between the signal constructed from the image and those stored in the SPD of the shortlisted star IDs.

The third proposed approach is the construction of a rotation-invariant vector frame for 2D vectors. In a general reference frame, 2D vectors are translation invariant. However, they are not rotation-invariant features. Thus, if 2D vectors are directly compared between the SPD and the image captured, they won't match. So, we proposed to construct the 2D vectors in a new locally built frame, which makes them rotation-invariant. However, this representation of the vectors in the new frame did not solve the problem of missing stars and false stars. Thus, we combined the additive property of 2D vectors along with the rotation-invariant vector frame for defining a new vector sequence. This vector sequence is highly robust to missing and false stars and is also rotation-invariant at the same time. We also proposed a shortlisting technique based on the idea of rotation-invariant vector frame.

After having proposed three approaches for solving the problem of star pattern recognition, we test and benchmark the proposed methods with the state-of-the-art star pattern recognition techniques. Simulated images are generated with the MATLAB software for the initial testing process. The simulated images are made to resemble as close as possible to the real images by adding magnitude uncertainty, positional deviation, and false stars to the images. Later, the simulations are followed by real-time testing of the proposed approaches. For this purpose, the proposed

methods are implemented on the hardware, and a prototype LIS mode star tracker is developed. The real-time testing is performed on the real star images captured by the star tracker SST-20S. The performance results both on the simulated and real images show that the proposed approaches are highly robust and recognition reliable providing a high identification accuracy along with the high speed of star identification.

Having summarized the research work pursued in this thesis, we can safely conclude that we are successful in achieving the aim of the thesis listed in the first chapter. Specifically, we have provided a quantitative analysis of the problems faced by a star pattern recognition technique. Based on the problems analyzed, we have also proposed three novel approaches for achieving a high identification accuracy, robustness, reliability, and speed of recognition. Moreover, the star shortlisting techniques proposed by us can be integrated with any of the existing state-of-the-art star pattern recognition techniques which have a high time complexity. This process of integration will help to reduce the time complexity of those techniques.

Having achieved the above major goals of the thesis, we propose some prospects that can be carried out for extending this research work and at the same time pursuing new approaches for improving the performance of the star pattern recognition techniques.

6.2 Future Work

Throughout this thesis, we have focused on the problems of patch mismatch, magnitude uncertainty, and false stars. These problems get transformed into trying to match a query image having false and missing data (stars in our case) with the database. This visualization of the problem can lead to many possible approaches when viewed concerning the methodologies in data science. Many data augmentation techniques are present in the literature which can be utilized to solve this problem of matching the query having missing and false data when compared to the database. These data augmentation techniques can be exploited to achieve better results.

When viewed in terms of data science, the techniques from machine learning and

even deep learning can be utilized for solving the problem. There are many machine learning techniques [100–104] which can learn the database images and find a match in the database when an input query image with missing and false stars is presented to it. Convolutional neural networks (CNN) [105] are another possible prospect of solving this problem of reliable star identification. Specifically, we believe 1D CNN [106], and Recurrent Neural Networks (RNNs) [107, 108] can be applied to our proposed image to signal approach for training the network. The signal can be served as an input to the 1D CNN, and the output of the system can be a correct class of stars (equivalent to shortlisting) or the star identification itself. The parameters of the network can be tweaked accordingly for achieving higher results of identification accuracy compared to the existing approaches.

Another possible approach for improving the reliability of the star pattern recognition is to differentiate the false stars from the real stars in the image. As the false stars are either reflection from the space debris, meteors, asteroids or source light of the planet, we can develop a methodology to differentiate them from the source light of the real stars in the sky. If such a distinguishable technique can be developed and realized in real-time, it will lead the star pattern recognition techniques to be only employed on the real stars in the image. Thus, the problem of false stars in the image will be inherently solved for a star pattern recognition technique.

Having solved the problem of reliable star identification, the further work includes increasing the orientation accuracy, i.e. the pointing accuracy of the star tracker. The pointing accuracy of the star tracker is calculated from the star vectors extracted from the image and the corresponding star vectors (of the identified stars) in the catalog. As the vectors in the star catalog are fixed, this pointing accuracy thus depends upon the star vectors extracted from the image captured. The accuracy of the star trackers extracted from the image further depends upon the centroiding accuracy. Therefore, improving the centroiding accuracy will lead to enhancing the pointing accuracy of the star tracker. There has already been some existing work pursued [109–111] for achieving a high pointing accuracy for the star tracker. The work mentioned above can be adapted as the basis for increasing the potential orientation accuracy of the star tracker.

Closing note: This thesis ends here, with the hope and belief that the in-depth quantitative analysis pursued, proposed approaches, and testing results will assist the research community in the domain of attitude determination of satellite to develop state-of-the-art attitude determination system based on star trackers.

Appendix A

Pseudo code of the Proposed Approaches

The pseudo code of the proposed approaches is presented here.

Algorithm 1 Proposed approach 1

```
1: procedure FEATURE EXTRACT
2:     fail = 0
3:     Slab = 1
4:     ref = null
5:     for i = 1 to n do
6:         dist = distance(i, I_c)        ▷ i - star in the image, I_c - center of the image
7:     end for
8:     ref = min(dist)                                          ▷ Finding reference star
9:     R^I_Sum = 0
10:    for i = 1 to n − 1 do
11:        dist = distance(i, ref)
12:        if dist <= FOV then
13:            R^I_Sum = R^I_Sum + dist
14:            rel_angle = angle(i, ref + 1)
15:        end if
16:    end for
17: end procedure
```

```
18: procedure SHORTLISTING TECHNIQUE
19:     function SHORTLIST(fail, Slab)                    ▷ Initially fail = 0, Slab = 1
20:         for i = 1 to size(SPD) do
21:             if checkbit(i) == 0 then
22:                 if R^I_Sum <= Slab then
23:                     checkbit(i) = 1
24:                     starshortlist = append(i)
25:                 end if
26:             end if
27:         end for
28:         patternrecognition(dist, rel_angle, starshortlist)    ▷ Shift to star pattern recognition
29:     end function
30: end procedure
31:
32: procedure STAR PATTERN RECOGNITION
33:     function PATTERNRECOGNITION(dist, rel_angle, starshortlist)
34:         success = 0
35:         fail_match = 0
36:         for i = 1 to n − 1 do
37:             for j = 1 to size(starshortlist) do
38:                 if dist(i) == radius_SPD(starshortlist((j), k))&&dist(i+1) == radius_SPD(starshortlist(j, l)) then
39:                     sumangleSPD = 0
40:                     for ind = k to l do
41:                         sumangleSPD = sumangleSPD + angle_SPD starshortlist(j), ind)
42:                     end for
43:                 end if
44:                 if rel_angle(i) == sumangleSPD then    ▷ Confirming initial match
45:                     init_match = 1
46:                 end if
47:             end for
48:         end for
49:         if init_match == 1 then
50:             ind_i = i + 2
51:             ind_SPD = l + 1
52:         end if
```

```
53:         for a = ind_i to n − 1 do
54:             if dist(a) == radius_SPD(starshortlist(j), x) then
55:                 sumangleimage = sumangleimage + rel_angle(a)
56:                 sumangleSPD = sumangleSPD+angle_SPD starshortlist(a), ind)
57:             end if
58:             if sumangleimage == sumangleSPD then
59:                 fail_match = 0
60:             else
61:                 fail_match = 1
62:             end if
63:         end for
64:         if fail_match ! = 1 then
65:             success = 1
66:             ref = j
67:         end if
68:         if success ! = 1 then
69:             shortlist(fail = 1, Slab = Slab + 1)
70:         else
71:             return = ref
72:         end if
73:     end function
74: end procedure
```

Algorithm 2 Proposed approach 2

procedure FEATURE EXTRACT ▷ This procedure remains the same as in the first approach
end procedure
procedure SHORTLISTING
    $S = round(log_2(PR))$ ▷ S - Slabs; PR - FOV
    $bin_{LUT} = round(rel_{angle} * N_{LUT}/180)$ ▷ N_{LUT} is the number of bins
    for $i = 0$ to $n - 1$ **do**
        if bin_{LUT} **then**
            $I_{32}(i) = round(dist * (S + 1)/PR)$
        end if
    end for
    if *HD(I_{32},LUT(i))* $<= 16$ **then** ▷ HD - Hamming distance
        $patternrecognition(i)$ ▷ Shortlist the star entry
    end if
end procedure
procedure PATTERNRECOGNITION
    function PATTERNRECOGNITION(k)
        $bin_{SPD} = round(rel_{angle} * N_{SPD}/180)$ ▷ N_{SPD} is the number of bins
        for $i = 0$ to $n - 1$ **do**
            if bin_{SPD} **then**
                $I_{512}(i) = dist$
            end if
        end for
        for $k = 0$ to $length(shortlist)$ **do**
            $corr(k) = SPC(I_{512}, SPD(k))$ ▷ SPC - Spearman-correlation
        end for
        $ref = ind(max(corr))$ ▷ Maximum spearman correlation - star ID
        return $= ref$
    end function
end procedure

Algorithm 3 Proposed approach 3

procedure FEATURE EXTRACT ▷ This procedure remains the same as in the first approach
end procedure
procedure SHORTLISTING
  for $k = 0$ to $n - 1$ **do**
    $Vb_n(k) = [Vb(k)^T V(k); Vb(k)^T V(k+1)]$ ▷ Vb_n - Rotation-invariant vector frame
  end for
  for $k = 0$ to $length(SPD)$ **do**
    $count(k) = match(Vb_n, SPD(k))$ ▷ Calculate the voting between the SPD and the image
  end for
  sort$(top1/3^{rd} count)$
  $patternrecognition(top1/3^{rd}(k))$ ▷ Shortlist the top one-third of the maximum matches
end procedure
procedure PATTERNRECOGNITION
  function PATTERNRECOGNITION(sh)
    for $i = 0$ to $n - 1$ **do**
      sort$(dist)$ ▷ Sort the neighboring star distance
    end for
    for $l = 0$ to 3 **do**
      for $m = 0$ to $length(sh)$ **do**
        if *dist(l) == dist(SPD(sh))* **then**
          $SP = l$
        end if
        calculate $Vb_n(SP)$ ▷ Calculate the rotation-invariant additive vector sequence
        for $z = 0$ to $length(sh)$ **do**
          $count(z) = match(Vb_n(SP), sh(z))$ ▷ Calculate the voting between the shortlisted list and the image
        end for
      end for
    end for
    return $= max(ind(count))$ ▷ Max. match - Star ID identified
  end function
end procedure

Author's Publications

Technology Disclosures

- D.S. Mehta, S. Chen, K.S. Low, "A high accuracy star tracker using running sequential angular match technique", *Singapore provisional patent application number* 10201603223W, Apr 22 2016.

Journals

- D.S. Mehta, S. Chen, K.S. Low, "A robust star identification algorithm with star shortlisting", *Advances in Space Research*, vol. 61(10), pp. 2647-2660, 2018.
- D.S. Mehta, S. Chen, K.S. Low, "A hamming distance and spearman correlation based star identification algorithm", *IEEE Transactions on Aerospace and Electronic Systems*, 2018.
- D.S. Mehta, S. Chen, K.S. Low, "A rotation-invariant additive vector sequence based star pattern recognition", *IEEE Transactions on Aerospace and Electronic Systems*, 2018.

Conferences

- D.S. Mehta, S. Chen, K.S. Low, "A high accuracy star tracker using running sequential angular match technique", In *Proceedings of IEEE Region 10 Conference (TENCON)*, pp. 3691-3694, 2016.
- D.S. Mehta, S. Chen, "A spearman correlation based star pattern recogniton", In *IEEE International Conference on Image Processing (ICIP)*, pp. 4372-4376, 2017.

- D.S. Mehta, W. Yan, S. Chen, "A star shortlisting technique for a Lost-in-space mode star tracker", In *Proceedings of IEEE Aerospace Conference*, pp. 1-8, 2018.
- D.S. Mehta, S. Chen, "A star pattern recognition technique based on the binary pattern formed from the FFT coefficients", In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp.1-5, 2018.

Bibliography

[1] Christine Anderson, C Vanek, H Freeman, D Furlong, A Kirschbaum, R Roy, P Wilhelm, and S Wander, "Lewis spacecraft mission failure investigation board final report," Tech. Rep., 1998.

[2] Christopher D Hall, "When spacecraft won't point," *Advances in the Astronautical Sciences*, vol. 116, pp. 1–7, 2004.

[3] Bong Wie, *Space Vehicle Dynamics and Control*, AIAA, 1998.

[4] Michael D Griffin, *Space Vehicle Design*, AIAA, 2004.

[5] A Davies and A Holt, "Use of autonomous star trackers in modern attitude and orbit control systems," *Spacecraft Guidance, Navigation and Control Systems*, vol. 516, pp. 87, 2003.

[6] Morris M Birnbaum, "Spacecraft attitude control using star field trackers," *Acta Astronautica*, vol. 39, no. 9-12, pp. 763–773, 1996.

[7] James R Wertz, *Spacecraft Attitude Determination and Control*, vol. 73, Springer Science & Business Media, 2012.

[8] Alexandra Witze et al., "Software error doomed Japanese Hitomi spacecraft," *Nature*, vol. 533, no. 7601, pp. 18–19, 2016.

[9] Gunter's Space Page, "Velox c1 satellite," Tech. Rep.

[10] Ludovic Blarre, Julien Ouaknine, Lionel Oddos-Marcel, and Pierre Emmanuel Martinez, "High accuracy sodern star trackers: recent improvements proposed on sed36 and hydra star trackers," vol. 6046, 2006.

[11] Carl C Liebe, Edwin W Dennison, Bruce Hancock, Robert C Stirbl, and Bedabrata Pain, "Active pixel sensor (aps) based star tracker," *IEEE Aerospace Conference Proceedings*, vol. 1, pp. 119–127, 1998.

[12] S Roeser and U Bastian, "A new star catalogue of SAO type," *Astronomy and Astrophysics Supplement Series*, vol. 74, pp. 449–451, 1988.

[13] Dorrit Hoffleit and Carlos Jaschek, "The bright star catalogue," *The Bright Star Catalogue, New Haven: Yale University Observatory*, vol. 4th edition.

[14] Michael AC Perryman, L Lindegren, J Kovalevsky, E Hoeg, U Bastian, PL Bernacca, M Crézé, F Donati, M Grenon, M Grewing, et al., "The HIPPARCOS catalogue," *Astronomy and Astrophysics*, vol. 323, pp. L49–L52, 1997.

[15] Minh Duc Pham, Kay-Soon Low, and Shoushun Chen, "An autonomous star recognition algorithm with optimized database," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 1467–1475, 2013.

[16] Michael D'Angelo and Richard Linares, "Attitude determination using a photon counting star tracker," *Image Processing Workshop (WNYIPW)*, pp. 1–4, 2011.

[17] Steve B Howell, *Handbook of CCD astronomy*, vol. 5, Cambridge University Press, 2006.

[18] Harold D Black, "A passive system for determining the attitude of a satellite," *AIAA journal*, vol. 2, no. 7, pp. 1350–1351, 1964.

[19] Malcolm David Shuster and S D_ Oh, "Three-axis attitude determination from vector observations," *Journal of Guidance, Control, and Dynamics*, vol. 4, no. 1, pp. 70–77, 1981.

[20] Carl Christian Liebe, AR Eisenman, BR Hancock, JA Mellstrom, JM Ratliff, GM Swift, JW Alexander, M Wadsworth, and WJ Walker, "Star tracker design considerations for the europa orbiter mission," *IEEE Aerospace Conference Proceedings*, vol. 2, pp. 67–81, 1999.

[21] Carl Christian Liebe, "Pattern recognition of star constellations for spacecraft applications," *IEEE Aerospace and Electronic Systems Magazine*, vol. 7, no. 6, pp. 34–41, 1992.

[22] Carl Christian Liebe, Leon Alkalai, George Domingo, Bruce Hancock, Don Hunter, Jeff Mellstrom, I Ruiz, Cesar Sepulveda, and Bedabrata Pain, "Micro aps based star tracker," *IEEE Aerospace Conference Proceedings*, vol. 5, pp. 5–5, 2002.

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

[24] Carl Ch Liebe, "Star trackers for attitude determination," *IEEE Aerospace and Electronic Systems Magazine*, vol. 10, no. 6, pp. 10–16, 1995.

[25] Allan Read Eisenman, Carl Christian Liebe, and John Leif Jorgensen, "The new generation of autonomous star trackers," 1997.

[26] Daniele Mortari, John L Junkins, and M Samaan, "Lost-in-space pyramid algorithm for robust star pattern recognition," *Guidance and Control*, pp. 49–68, 2001.

[27] Daniele Mortari and Beny Neta, "K-vector range searching techniques," 2014.

[28] Daniele Mortari, Malak A Samaan, Christian Bruccoleri, and John L Junkins, "The pyramid star identification technique," *Navigation*, vol. 51, no. 3, pp. 171–183, 2004.

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

[30] Craig L Cole and John L Crassidis, "Fast star-pattern recognition using planar triangles," *Journal of Guidance Control and Dynamics*, vol. 29, no. 1, pp. 64–71, 2006.

[31] G Lamy Au Rousseau, J Bostel, and B Mazari, "Star recognition algorithm for aps star tracker: oriented triangles," *IEEE Aerospace and Electronic Systems Magazine*, vol. 20, no. 2, pp. 27–31, 2005.

[32] Michael Kolomenkin, Sharon Pollak, Ilan Shimshoni, and Michael Lindenbaum, "Geometric voting algorithm for star trackers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, 2008.

[33] MD Pham, KS Low, Chen Shoushun, and YT Xing, "A star pattern recognition algorithm for satellite attitude determination," *IEEE Symposium on Industrial Electronics and Applications (ISIEA)*, pp. 236–241, 2012.

[34] Phillip Alvelda and A Miguel San Martin, "Neural network star pattern recognition for spacecraft attitude determination and control," *Advances in Neural Information Processing Systems*, pp. 314–322, 1989.

[35] Chunyan Li, Ke Li, Longyun Zhang, Shengzhen Jin, and Jifeng Zu, "Star pattern recognition method based on neural network," *Chinese Science Bulletin*, vol. 48, no. 18, pp. 1927–1930, 2003.

[36] Lalitha Paladugu, Marco P Schoen, and Brian G Williams, "Intelligent techniques for star-pattern recognition," *Proceedings of ASME, IMECE*, pp. 16–21, 2003.

[37] Ai Xiang He, Chen Chen Wang, and Hai Ning Zhang, "A star map recognition method based on multi-layers sofm network," *Trans Tech Applied Mechanics and Materials*, vol. 411, pp. 1011–1014, 2013.

[38] Yang Jing and Wang Liang, "An improved star identification method based on neural network," *10th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 118–123, 2012.

[39] Ye Wang and Haopeng Zhang, "Star recognition based on mixed star pattern and multilayer som neural network," *IEEE Aerospace Conference Proceedings*, pp. 1–6, 2017.

[40] Li Sun, Jie Jiang, Guangjun Zhang, and Xinguo Wei, "A discrete hmm-based feature sequence model approach for star identification," *IEEE Sensors Journal*, vol. 16, no. 4, pp. 931–940, 2016.

[41] Vincenzo Schiattarella, Dario Spiller, and Fabio Curti, "A novel star identification technique robust to high presence of false objects: The multi-poles algorithm," *Advances in Space Research*, vol. 59, no. 8, pp. 2133–2147, 2017.

[42] Xinguo Wei, Guangjun Zhang, and Jie Jiang, "Star identification algorithm based on log-polar transform," *Star*, vol. 6, 2009.

[43] Tjorven Delabie, Thomas Durt, and Jeroen Vandersteen, "Highly robust lost-in-space algorithm based on the shortest distance transform," *Journal of Guidance, Control, and Dynamics*, 2013.

[44] Jian Li, Xinguo Wei, and Guangjun Zhang, "Iterative algorithm for autonomous star identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 1, pp. 536–547, 2015.

[45] Guangjun Zhang, Xinguo Wei, and Jie Jiang, "Full-sky autonomous star identification based on radial and cyclic features of star pattern," *Image and Vision computing*, vol. 26, no. 7, pp. 891–897, 2008.

[46] Xin-guo Wei, Guang-jun Zhang, and Jie Jiang, "A star map identification algorithm using radial and cyclic features," *Opto-electronic Engineering*, vol. 8, pp. 001, 2004.

[47] Jafar Roshanian, Shabnam Yazdani, and Masoud Ebrahimi, "Star identification based on euclidean distance transform, voronoi tessellation, and k-nearest neighbor classification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 6, pp. 2940–2949, 2016.

[48] Edward J Groth, "A pattern-matching algorithm for two-dimensional coordinate lists," *The Astronomical Journal*, vol. 91, pp. 1244–1248, 1986.

[49] D Baldini, M Barni, A Foggi, G Benelli, and A Mecocci, "A new star-constellation matching algorithm for satellite attitude determination," *ESA journal*, vol. 17, pp. 185–198, 1993.

[50] MS Scholl, "Star-field identification for autonomous attitude determination," *Journal of Guidance, Control, and Dynamics*, vol. 18, no. 1, pp. 61–65, 1995.

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

[52] Xin-guo, Wei Zhang Guang-jun, and Jie Jiang, "Star map identification based on a modified triangle algorithm," *Acta Aeronautica Et Astronautica Sinica*, vol. 6, pp. 031, 2006.

[53] Jer-Nan Juang, Hye-Young Kim, and John L Junkins, "An efficient and robust singular value method for star pattern recognition and attitude determination," 2003.

[54] E Antonio Hernández, Miguel A Alonso, Edgar Chávez, David H Covarrubias, and Roberto Conte, "Robust polygon recognition method with similarity invariants applied to star identification," *Advances in Space Research*, vol. 59, no. 4, pp. 1095–1111, 2017.

[55] M Shayan Arani, A Toloei, and Z Eghbaleh, "A geometric star identification algorithm based on triple triangle pattern," *7th IEEE International Conference on Recent Advances in Space Technologies (RAST)*, pp. 81–85, 2015.

[56] Meng Na, Danian Zheng, and Peifa Jia, "Modified grid algorithm for noisy all-sky autonomous star identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 45, no. 2, 2009.

[57] Mahdi Aghaei and Hamid Abrishami Moghaddam, "Grid star identification improvement using optimization approaches," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 5, pp. 2080–2090, 2016.

[58] Enrico Silani and Marco Lovera, "Star identification algorithms: Novel approach & comparison study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 4, 2006.

[59] Youngwoo Yoon, "Autonomous star identification using pattern code," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 2065–2072, 2013.

[60] Suraphol Udomkesmalee, James W Alexander, and Aurelio F Tolivar, "Stochastic star identification," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 6, pp. 1283–1286, 1994.

[61] LH Li, FE Zhang, and T Lin, "An all-sky autonomous star map identification algorithm based on genetic algorithm," *Opto-Electron Eng*, vol. 27, no. 5, pp. 15–18, 2000.

[62] Pablo A Servidia and RS Sanchez Pena, "New robust star identification algorithm," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 3, 2006.

[63] Jian Hong and Julie A Dickerson, "Neural-network-based autonomous star identification algorithm," *Journal of Guidance Control and Dynamics*, vol. 23, no. 4, pp. 728–735, 2000.

[64] Clark S Lindsey, Thomas Lindblad, and Age Eide, "A method for star identification using neural networks," *Proceedings SPIE International Society for Optical Engineering*, pp. 471–478, 1997.

[65] Greg Bardwell, "On-board artificial neural network multi-star identification system for 3-axis attitude determination," *Acta Astronautica*, vol. 35, pp. 753–761, 1995.

[66] Daniele Mortari, "Search-less algorithm for star pattern recognition," *Astronaut Sci*, vol. 45, no. 2, pp. 179–194, 1997.

[67] Jian Yang, Guang-jun Zhang, and Jie Jiang, "Fast star identification algorithm using p vector," *Acta Aeronautica Et Astronautica Sinica*, vol. 28, no. 4, pp. 897, 2007.

[68] David D Needelman, James P Alstad, Peter C Lai, and Haytham M Elmasri, "Fast access and low memory star pair catalog for star pattern identification," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 5, pp. 1396, 2010.

[69] Malak A Samaan, Daniele Mortari, and John L Junkins, "Nondimensional star identification for uncalibrated star cameras," *The Journal of the Astronautical Sciences*, vol. 54, no. 1, pp. 95–111, 2006.

[70] Toshiro Sasaki and Michitaka Kosaka, "A star identification method for satellite attitude determination using star sensors," *15th International Symposium on Space Technology and Science*, pp. 1125–1130, 1986.

[71] Eleanor A Ketchum and Robert H Tolson, "Onboard star identification without a priori attitude information," *Journal of Guidance Control and Dynamics*, vol. 18, no. 2, pp. 242–246, 1995.

[72] BM Quine and Hugh F Durrant-Whyte, "A fast autonomous star-acquisition algorithm for spacecraft," *Control Engineering Practice*, vol. 4, no. 12, pp. 1735–1740, 1996.

[73] Malak A Samaan, Christian Bruccoleri, Daniele Mortari, and John L Junkins, "Novel techniques for the creation of a uniform star catalog," *Advances in the Astronautical Sciences*, vol. 116, pp. 1691–1703, 2003.

[74] BM Quine and Hugh F Durrant-Whyte, "Rapid star pattern identification," *SPIE*, vol. 2739, pp. 351–360, 1996.

[75] Qi-Shen Li, Chang-Ming Zhu, and Jun Guan, "A fast star pattern recognition algorithm based on feature vector," *IEEE International Conference on Computer Design and Applications (ICCDA)*, vol. 1, pp. V1–256, 2010.

[76] Daniele Mortari and Michela Angelucci, "Star pattern recognition and mirror assembly misalignment for digistar ii and iii multiple fovs star sensors," *Adv. Astronaut. Sci*, vol. 102, pp. 1175–1184, 1999.

[77] D Mortari and A Romoli, "Navstar iii: A three fields of view star tracker," *IEEE Aerospace Conference Proceedings*, vol. 33, 2002.

[78] Roberto Opromolla, Giancarmine Fasano, Giancarlo Rufino, Michele Grassi, Claudio Pernechele, and Cesare Dionisio, "A new star tracker concept for satellite attitude determination based on a multi-purpose panoramic camera," *Acta Astronautica*, vol. 140, pp. 166–175, 2017.

[79] SH Sohrabi and AA Beheshti Shirazi, "A novel, smart and fast searching method for star pattern recognition using star magnitudes," *25th IEEE International Conference of Image and Vision Computing New Zealand (IVCNZ)*, pp. 1–6, 2010.

[80] Junfeng Xie and Xiao Wang, "A robust autonomous star identification algorithm for zy3 satellite," *First IEEE International Conference onAgro-Geoinformatics (Agro-Geoinformatics)*, pp. 1–4, 2012.

[81] Curtis Padgett, Kenneth Kreutz-Delgado, and Suraphol Udomkesmalee, "Evaluation of star identification techniques," *Journal of Guidance Control and Dynamics*, vol. 20, no. 2, pp. 259–267, 1997.

[82] DOMENICO Accardo and GIANCARLO Rufino, "Brightness-independent start-up routine for star trackers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 3, pp. 813–823, 2002.

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

[84] David M Gottlieb, "Star identification techniques," 1978.

[85] K Ho, "A survey of algorithms for star identification with low-cost star trackers," *Acta Astronautica*, vol. 73, pp. 156–163, 2012.

[86] Guangjun Zhang, *Star Identification: Methods, Techniques and Algorithms*, Springer, 2016.

[87] Roelof W H Van Bezooijen, "Automated star pattern recognition," 1990.

[88] David Stephen Anderson, "Autonomous star sensing and pattern recognition for spacecraft attitude determination," 1991.

[89] X Wei, *A research on star identification methods and relevant technologies in star sensor*, Ph.D. thesis, Doctoral Thesis of Beijing University Aeronautics and Astronautics, Beijing, 2004.

[90] J Yang, *A research on star identification algorithm and RISC technology application*, Ph.D. thesis, Doctoral Thesis of Beijing University Aeronautics and Astronautics, Beijing, 2007.

[91] Jack A Tappe, *Development of star tracker system for accurate estimation of spacecraft attitude*, Ph.D. thesis, Monterey, California. Naval Postgraduate School, 2009.

[92] Jorge G Padro, "Development of a star tracker-based reference system for accurate attitude determination of a simulated spacecraft," Tech. Rep., 2012.

[93] Richard W Hamming, "Error detecting and error correcting codes," *Bell Labs Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[94] Mohammad Norouzi, David J Fleet, and Ruslan R Salakhutdinov, "Hamming distance metric learning," *Advances in Neural Information Processing Systems*, pp. 1061–1069, 2012.

[95] John B Carroll, "The nature of the data, or how to choose a correlation coefficient," *Psychometrika*, vol. 26, no. 4, pp. 347–372, 1961.

[96] Jan Hauke and Tomasz Kossowski, "Comparison of values of pearson's and spearman's correlation coefficients on the same sets of data," *Quaestiones geographicae*, vol. 30, no. 2, pp. 87, 2011.

[97] Giancarlo Rufino, Domenico Accardo, Michele Grassi, Giancarmine Fasano, Alfredo Renga, and Urbano Tancredi, "Real-time hardware-in-the-loop tests of star tracker algorithms," *International Journal of Aerospace Engineering*, vol. 2013, 2013.

[98] Xin-guo Wei, Guang-jun Zhang, Qiao-yun Fan, and Jie Jiang, "Ground function test method of star sensor using simulated sky image," *Infrared and Laser Engineering*, vol. 6, pp. 040, 2008.

[99] XILINX, "Zynq-7000 all programmable soc overview, advance product specification-ds190," Tech. Rep.

[100] Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf, "Support vector machines," *IEEE Intelligent Systems and their applications*, vol. 13, no. 4, pp. 18–28, 1998.

[101] Daniel T Larose, "k-nearest neighbor algorithm," *Discovering knowledge in data: An introduction to data mining*, pp. 90–106, 2005.

[102] K Krishna and M Narasimha Murty, "Genetic k-means algorithm," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 29, no. 3, pp. 433–439, 1999.

[103] William HE Day and Herbert Edelsbrunner, "Efficient algorithms for agglomerative hierarchical clustering methods," *Journal of Classification*, vol. 1, no. 1, pp. 7–24, 1984.

[104] Mahesh Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, pp. 217–222, 2005.

[105] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 2012.

[106] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen, "Convolutional neural network architectures for matching natural language sentences," *Advances in Neural Information Processing Systems*, pp. 2042–2050, 2014.

[107] LR Medsker and LC Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, 2001.

[108] Felix A Gers, Jürgen Schmidhuber, and Fred Cummins, "Learning to forget: Continual prediction with lstm," *IET*, pp. 850–855, 1999.

[109] Giancarlo Rufino and Domenico Accardo, "Enhancement of the centroiding algorithm for star tracker measure refinement," *Acta Astronautica*, vol. 53, no. 2, pp. 135–147, 2003.

[110] Hao Xuetao Jiang Jie Zhang Guangjun, "Cmos star sensor image acquisition and real-time star centroiding algorithm," *Journal of Beijing University of Aeronautics and Astronautics*, vol. 4, pp. 001, 2005.

[111] Wei Xinguo, Zhang Guangjun, and Jiang Jie, "Subdivided locating method of star image for star sensor," *Journal of Beijing university of Aeronautics and Astronautics*, vol. 29, no. 9, pp. 812–815, 2003.