

# A Star Pattern Recognition Algorithm for Satellite Attitude Determination

M. D. Pham, K. S. Low, Chen Shoushun and Y. T. Xing

School of Electrical and Electronic Engineering

Nanyang Technological University, Singapore

**Abstract**—To determine the satellite attitude in space, various attitude sensors are used. Among them, the most accurate sensor is the star tracker. In this paper, a novel star pattern recognition algorithm in the “lost in space” mode is presented. The proposed method improves the searching speed by arranging the large star catalogue using a search tree data structure. To further improve the processing speed, the star image is processed at the grid level instead of pixel level. The results from the study demonstrate that the proposed approach has significantly reduced the average run-time by 50% as compared to the conventional methods while still achieve slightly better star recognition accuracy at 95.07%.

**Index terms** – satellite attitude determination; star tracker; image processing and pattern recognition.

## I. INTRODUCTION

To produce the 3-axis attitude information of a satellite, an optical-electronics device known as the star tracker is commonly used. Typically, a satellite employs several attitude determination sensors such as magnetic sensors, sun sensors, earth’s horizon scanner etc. However, the star tracker is still the most accurate solution for spacecraft with a bore sight accuracy of 20-90 arc second [1]. Since the stars’ positions remain relatively fixed on the Earth Centre Inertial frame, their positions can be used as a reference to determine the spacecraft attitude using QUEST or TRIAD methods etc [2].

Over the past decades, a variety of star pattern recognition methods have been proposed [3-17] based on extracting important features of star cluster. These methods vary in complexity, database size, robustness, recognition accuracy and processing time. In general, these algorithms can be categorized into angular distance based method and grid based method. In the angular distance based method, the star pattern is characterized by the angular distances between the reference and the neighbouring stars. The Liebe algorithm [3-4], Pyramid algorithm [5-8], planar triangle algorithm [9] and geometric voting algorithm [10] belong to this category.

In [11], a grid-based method uses the bit pattern to recognize stars. The bit pattern is generated by applying a grid layer on the captured star images. Subsequently, several improved versions of the grid based algorithm have been proposed. Two variation of grid-based method are the polar grid method [12] and the elastic grey grid algorithm [13]. All the star pattern recognition methods focus primarily on the star pattern identification but not on the star pattern catalogue. In this paper, a novel method is proposed to overcome this shortcoming. The results show that this new approach is a fast and reliable method for autonomous star pattern recognition. There are two major innovations of the proposed method: 1)

an optimized database using search tree data structure for fast search, 2) a parallel search feature is employed.

The key advantage of the search tree is to narrow down the search region in every iteration to improve the searching speed. In conventional methods, star pattern catalogue  $D$  is not optimized for size and search speed. The search process is simply scanning through the catalogue  $D$ . The estimated runtime of conventional methods is  $O(N)$ . Instead of scanning through the entire catalogue to complete the process, the proposed search scheme requires only a few iterations. For the proposed star identification method, the worst-case of the run time is estimated around  $O(2\log_2 N)$ , where  $N$  is the length of star catalogue  $D$ . In addition, the multiple feature vectors can be searched in parallel in a search tree. Consequently the search speed is significantly faster. The study shows that the proposed method requires a small memory of 30kB, and the run-time has been reduced by 50% as compared to the conventional methods. The proposed method also in-cooperates grid-based image processing to compensate for the image noise.

This paper is organised as follow. Section II introduces the star pattern database generation. This section describes the search tree construction based on star pattern database. A new star recognition method is developed for the “lost in space” mode. The search processes are also introduced to identify stars. In Section III, a laboratory star tracker prototype is described. The star tracker is capable of determining its three axes attitude. The simulation and experimental results are then presented in section IV. The proposed approach has been benchmarked with three algorithms namely Liebe method [3], pyramid method [5] and geometric voting algorithm [10]. The results validate the desired performance of the star tracker. Finally, section VI concludes this work.

## II. AUTONOMOUS STAR PATTERN RECOGNITION

### A. Database generation

The star pattern database is first processed on the ground and loaded subsequently into the star tracker on-board memory. A typical camera’s sensitivity is  $M < 5$ . With this constraint, only those stars that have visual magnitude less than 5 units from the SAO J2000 star catalogue are extracted. Moreover, double, triple, or multiple stars are also discarded to prevent misidentification. The reduced star catalogue  $T$  consists of 1631 stars. Next, a star pattern database  $D$  is generated to store the star patterns. The database generation procedure is as follows:

(1) For each reference star  $S_i$  in catalogue  $T$ , find all the neighbour stars  $S_j$  such that the distance between the neighbour stars to the reference star is less than half the field of view length  $d_f$  of star tracker camera as shown in Figure 1a, i.e.  $\mathbf{V}_i$  and  $\mathbf{V}_j$  are position vectors of star  $S_i$  and  $S_j$  in the Earth centre inertial frame.

$$\mathbf{V}_i = [x_i, y_i, z_i]; \mathbf{V}_j = [x_j, y_j, z_j] \quad (1)$$

$$d_{ij} = |\mathbf{V}_i \cdot \mathbf{V}_j| \leq 0.5d_f \quad (2)$$

The condition (2) is to find all the neighbouring stars  $S_j$  of star  $S_i$  that is visible to the star camera. The star  $S_i$  is situated at the z pointing direction of star camera. Hence, all stars  $S_j$  which is visible to the star camera have angular distance to star  $S_i$   $d_{ij} = |\mathbf{V}_i \cdot \mathbf{V}_j| \leq 0.5d_f$

(2) The camera attitude matrix  $\mathbf{C}$  is configured such that the camera bore sight vector (z axis) is aligned with the reference star.

$$\begin{bmatrix} \mathbf{c}_x \\ \mathbf{c}_y \\ \mathbf{c}_z \end{bmatrix} = \begin{bmatrix} \sqrt{z_i^2/(x_i^2 + y_i^2)} & -(x_i/y_i)\sqrt{z_i^2/(x_i^2 + y_i^2)} & 0 \\ 0 & 0 & 0 \\ x_i & y_i & z_i \end{bmatrix} \quad (3)$$

(3) The reference star attitude  $\mathbf{V}_i^{cam}$  and the neighbour star attitude  $\mathbf{V}_j^{cam}$  in the camera frame are calculated as:

$$\begin{aligned} \mathbf{V}_i^{cam} &= [x_i^{cam}; y_i^{cam}; z_i^{cam}] = \mathbf{C} \cdot \mathbf{V}_i \\ \mathbf{V}_j^{cam} &= [x_j^{cam}; y_j^{cam}; z_j^{cam}] = \mathbf{C} \cdot \mathbf{V}_j \end{aligned} \quad (4)$$

(4) The star attitude  $\mathbf{V}_i^{cam}$  and  $\mathbf{V}_j^{cam}$  are projected onto the CMOS image sensor plane centre as in Figure 1b. The star position  $(Px_i, Py_i)$  and  $(Px_j, Py_j)$  on the image sensor can be determined as follows:

$$\begin{aligned} Px_i &= \frac{f}{\rho} \frac{x_i^{cam}}{z_i^{cam}} + \frac{\mu_v}{2}; Py_i = \frac{f}{\rho} \frac{y_i^{cam}}{z_i^{cam}} + \frac{\mu_h}{2} \\ Px_j &= \frac{f}{\rho} \frac{x_j^{cam}}{z_j^{cam}} + \frac{\mu_v}{2}; Py_j = \frac{f}{\rho} \frac{y_j^{cam}}{z_j^{cam}} + \frac{\mu_h}{2} \end{aligned} \quad (5)$$

where  $f, \rho, m_v$  and  $m_h$  are the optical focal length, the pixel size, vertical and horizontal dimensions of CMOS image sensor.

(5) The star visual magnitudes  $M$  are also included in the simulated images as shown in Figure 1c. The image noise is also included based on Gaussian distribution

$$I(Px, Py) = A \cdot \exp \left[ -\frac{(Px - Px_i)^2}{2\sigma_x^2} - \frac{(Py - Py_i)^2}{2\sigma_y^2} \right] \quad (6)$$

where  $I(Px, Py)$  is the pixel intensity at position  $(Px, Py)$ ,  $(Px_i, Py_i)$  is the centroid of the star  $S_i$ ,  $A$  is the maximum intensity of the star and  $(\sigma_x, \sigma_y)$  is the position variance along the vertical and horizontal directions. The maximum pixel intensity  $I_m$  is exponentially proportional to the star visual magnitude  $M$  as  $M = 2.5 \log_{10}(I_m) + C_M$  where  $C_M$  is a predefined constant [18].

(6) The star image  $I$  is applied a square grid  $g \times g$   $g > (\sigma_x, \sigma_y)$  to become image  $G$ , as shown in Figure 1d. In this study,  $g$  is set to 2 pixels. The pixel  $G(m, n)$  is set to 1 if a star projection falls into this pixel, else  $G(m, n)$  is set as 0. If summation of pixel intensity  $I(Px, Py)$  within  $G(m, n)$  is greater than the threshold  $\zeta$ , pixel  $G(m, n)$  is set to 1.

$$G(m, n) = \begin{cases} 0, & \sum_{x=(m-1)^*g+1}^{m^*g} \sum_{y=(n-1)^*g+1}^{n^*g} I(Px, Py) < \zeta \\ 1, & \sum_{x=(m-1)^*g+1}^{m^*g} \sum_{y=(n-1)^*g+1}^{n^*g} I(Px, Py) > \zeta \end{cases} \quad (7)$$

(7) The planar distances  $[D_k, k = 1, \dots, N]$  from the reference star  $S_i$  to all  $N$  neighbour stars  $S_j$  are calculated and sorted in ascending order from the generated star image. The feature vector  $f_{S_i}$  of reference star  $S_i$  is created as  $f_{S_i} = [S_i, N, \{D_k, k = 1, \dots, N\}]$  and appended into the star pattern database  $D$ .

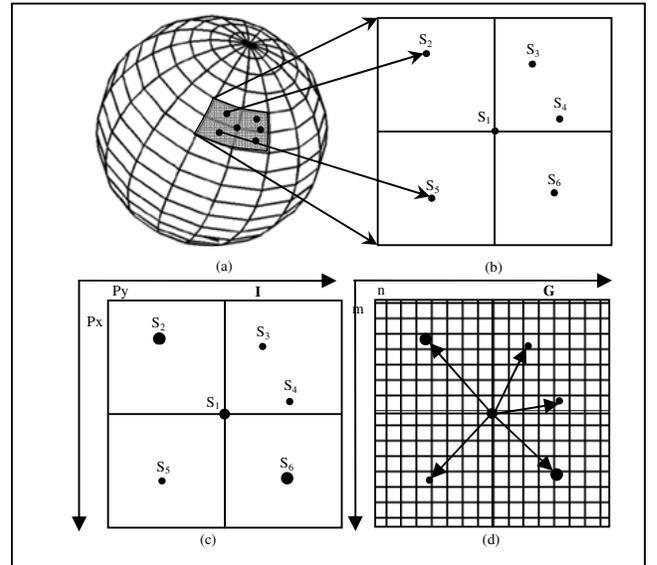


Figure 1. Star pattern database  $D$  generation.

(8) Finally, the star pattern database  $D$  is sorted in ascending order according to the number of neighbouring stars  $N$ . The entries with the smallest  $N$  are placed at the top of the database. Similarly, the first and subsequent columns of planar distances from the reference to neighbouring stars are also sorted in ascending order. Figure 2 illustrates one such star pattern database  $D$ . In Figure 2, the first column is the star identity number of reference star  $S_i$  which is extracted from SAO J2000. It ranges from 1 to 1631. The second column is the number of neighbouring stars  $N$  around the reference star  $S_i$ . The subsequent columns, i.e.  $D_1$  to  $D_5$ , are the distances  $D_k$  from all neighbouring stars  $S_j$  to the reference star  $S_i$ . Based on our experimental study using the SAO J2000 star catalogue, there are 14 distance columns and 14 sorting processes for  $N=14$ . These columns are sorted in ascending order.

ID	N	D <sub>1</sub>	D <sub>2</sub>	D <sub>3</sub>	D <sub>4</sub>	D <sub>5</sub>
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 2. Structure of star pattern database  $D$ .

### B. Search tree construction and single search process

Before the search process, the search tree is constructed based on the star pattern catalogue. An example is illustrated in Figure 3. The search tree is a data structure that consists of layers of nodes. Each node is a decision rule that decides the path to the next layer based on the elements of feature vector  $f_{Si}$ . The number of layer is equal to the number of neighbour stars  $N$ . The search tree is constructed from the top to the bottom layer. The first layer's decision value is  $N$ . The second layer is the decision value  $D_1$ , and the subsequent layer's decision values are  $D_2$ ,  $D_3$  etc. The final layer is the star identity numbers. Each combination of decision values from layer 1 to the final layer creates a path, known as the search path. The search path leads to a unique identity number at the end of search tree.

Consider  $f_{Si} = \{N=5, D_1=6, D_2=13, D_3=20, D_4=20, D_5=27\}$ . As shown in Figure 3, the feature vector implies that  $N=5$  is chosen in the first layer. The search path is narrowed down to sub-tree  $N=5$ . This sub-tree consists of 4 decision values  $\{D_1=5, 6, 7, 9\}$ . So the decision value  $D_1=6$  is chosen in the second layer. The decision value  $D_1=6$  leads to a sub-tree in the third layer. The third layer has decision values  $\{D_2=11, 13, 16, 17\}$ . From feature vector,  $D_2=13$  is chosen. The search process is continued until the sixth layer which is  $D_5=27$ . Finally, star identity 1395 is returned.

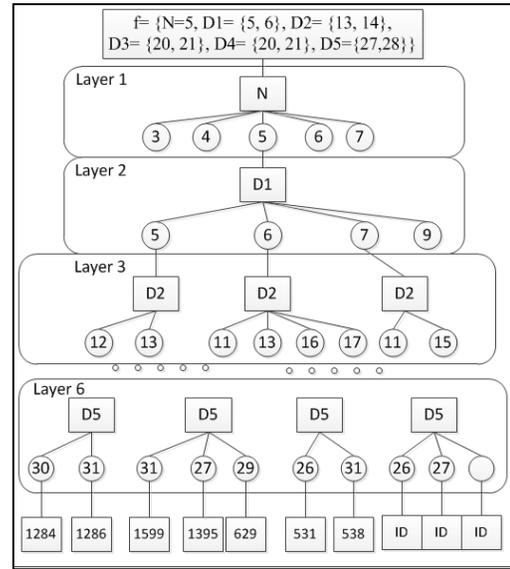


Figure 3. Search tree structure.

### C. Autonomous star pattern recognition

AUTONOMOUS STAR PATTERN RECOGNITION ALGORITHM	
1.	Capture star image $I$ and find $N$ stars in image
2.	Find star centroid $(P_x, P_y)$ of star $S_i$
	$P_x = \frac{\sum I(P_x, P_y) \cdot P_x}{\sum I(P_x, P_y)}, (i=1, \dots, N)$ $P_y = \frac{\sum I(P_x, P_y) \cdot P_y}{\sum I(P_x, P_y)}, (i=1, \dots, N)$
3.	IF $N < 3$
4.	Go to step 1
5.	ELSE
6.	Chose the reference star $S_{ref}$ as the brightest star
7.	Apply a square grid $g \times g$ to image $I$ ,
8.	Create the feature vector $f = \{N, D_1, D_2, D_3, D_4, \dots, D_{n-1}\}$ .
9.	Add noise tolerance $\eta$ , $f = \{N+\eta, D_1+\eta, D_2+\eta, D_3+\eta, D_4, \dots, D_{n-1}+\eta\}$ .
10.	IF ( $S_{ref}$ found in search tree)
11.	$S_{ref}$ is confirmed
12.	Calculate satellite attitude
13.	ELSE
14.	Go to step 6
15.	ENDIF
16.	ENDIF

The algorithm above illustrates the autonomous star pattern recognition method using the search tree data structure. The camera first captures the star image and then it performs the following tasks: The image pre-processing is first performed to improve the image quality. The process includes noise filtering, image thresholding and star labelling to find the number of stars  $N$  in the image. The star centroid is then calculated based on the weighted pixel values to determine the star position on image plane. The star pattern vector is extracted from the captured star image and searched in the search tree. If a unique star candidate is returned, the proposed method is repeated to the next brightest star to confirm the result. If the two identified stars are matched, a SUCCESS code is returned. If multiple candidates are returned, the

proposed method is applied to the subsequent brightest stars. The results are used to confirm until a unique identity is returned. Once a star identity  $S_i$  is confirmed, several attitude determination methods can be used to estimate the satellite attitude, for examples QUEST or TRIAD methods [2]. This attitude is then used in the tracking mode of star tracker.

### III. STAR TRACKER HARDWARE AND STAR SIMULATOR

A laboratory star tracker's prototype has been developed as shown in Figure 4. It is capable of determining the satellite's attitude about all three axes. The hardware consists of an image sensor board, a FPGA board and a single board computer. As shown in Figure 4, the upper board is the image sensor board with optical lens. The bottom board is a single board computer with FPGA for image acquisition. The image sensor board is equipped with an APTINA CMOS image sensor (i.e. MT9V034). Its resolution is 512x512 pixels with a pixel dimension of 7x7  $\mu\text{m}$ . The sensor provides fast capture speed at 60 frames per second. The image sensor has a 20 degree field of view. An on-chip analog-to-digital converter provides a resolution of 10 bits per pixel. The data output from MT9V034 is synchronized with the pixel clock running at 25MHz. The pixel data from the image sensor is acquired by the FPGA board and forward to the single board computer for real-time image processing. The system has a dimension of 10x10x5cm<sup>3</sup>. The PC-104 based embeddable single board computer has an AMD Geode™ LX800 processor operating at 500 MHz. The single board has a memory size of 250MB RAM, and consumes 5W of power. It has a wide operating temperature range from -40 to 85 degree.



Figure 4. Star tracker prototype

The star tracker software consists of five functional modules: camera control access, star image processing, star pattern recognition, attitude estimation, and telemetry interface. In addition, the star catalogue and mission catalogue required for the star pattern recognition is also incorporated into the software. The camera control module is used to control image sensor exposure time and ADC gain to produce high quality star image under low illumination condition. The star image is then processed to extract the star information. The proposed autonomous star pattern recognition method as presented in section II is used to determine the star identity. The TRIAD or QUEST method is then used to determine the satellite attitude based on the identified stars. The star tracker also provides telemetry interface for downlink housekeeping data back to the ground station such as the satellite attitude and star images.

## IV. RESULTS

### A. Star image tests

A set of star images have been chosen to verify the performance of the proposed method. The TRIAD method is utilized to determine the final optimum attitude estimate. Each attitude is compared with the true attitude given from the original images. Table 1 shows a typical star identification result with the positions of measured stars and catalogued stars. Figure 5 and 6 show two examples of the star identification results. In these figures, the symbols \* indicate the original star positions and the squares symbols  $\square$  indicate the identified star positions. Figure 5 shows the Albireo image and Figure 6 shows the Antares image. For the Albireo image, the star at position (256,256) is chosen as the guide star. The neighbour stars at positions (220, 214), (303, 200) and (340, 244) are calculated from the star image. The distances between the guide star and neighbour stars are calculated and searched in star pattern catalogue. The star pattern identification returns the star identity as 1268, 1226 and 1241. The position vectors of the identified stars are calculated in camera body frame  $\mathbf{V}_i$  and inertial frame  $\mathbf{W}_i$ :

$$\begin{aligned} \mathbf{V}_1 &= [0.446; -0.764; 0.465]; & \mathbf{W}_1 &= [0; 0.1169; 0.9931] \\ \mathbf{V}_2 &= [0.311; -0.811; 0.492]; & \mathbf{W}_2 &= [-0.029; -0.02; 0.999] \\ \mathbf{V}_3 &= [0.363; -0.784; 0.502]; & \mathbf{W}_3 &= [-0.038; 0.032; 0.998] \end{aligned}$$

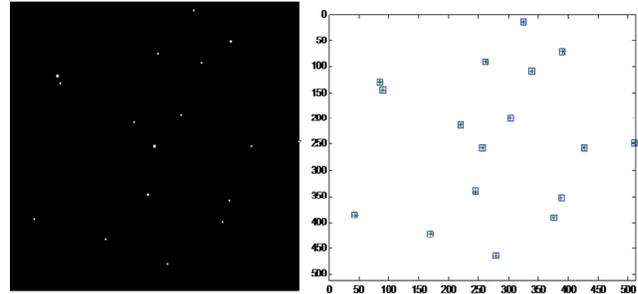


Figure 5. Albireo star image

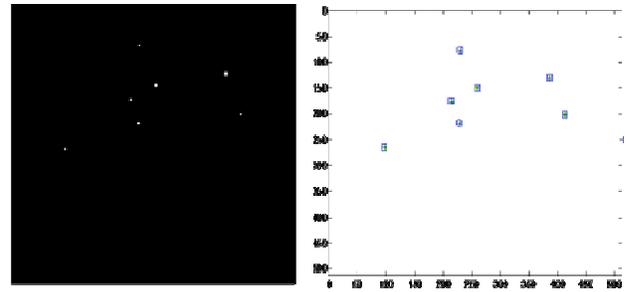


Figure 6. Antares star image

The satellite attitude is then calculated and compared with the true satellite attitude. The pointing accuracy on the bore-sight direction is 0.0259 degree, and in other directions are 0.9897 degree and 0.9900 degree. In the case of Albireo image, the guide star is returned as multiple candidates 1268 and 427. For each star candidate, further verification is calculated to confirm the true identity. For example, the

pattern of star identity 427 is checked from star pattern catalogue and compare with the pattern in Albireo image. The star identity 427 is rejected as the pattern from catalogue is different from pattern from captured image. When there is no returned star candidate, star identification is applied on another guide star or another star image. TABLE 1 shows the star identification test results with the number of observed stars, true attitude and estimated errors. The star position is specified by its right ascension (RA) and declination (DEC). They are the horizontal and vertical positions of star in the sky sphere respectively. In TABLE 1, the star tracker is pointing to the specific stars with specific RA and DEC. The star tracker performs star pattern recognition and attitude determination to determine its attitude. The calculated attitude is compared with the original RA and DEC. The x, y and z accuracy is shown in TABLE 1.

TABLE 1. STAR IDENTIFICATION TEST RESULTS

Name	No of observed star	True Attitude		Estimated error		
		RA	DEC	x	y	z
Albireo	18	292.679	27.9671	0.9897	0.9900	0.0259
Alder	14	322.288	62.6406	0.2032	0.2036	0.0191
Antares	7	208.521	14.2669	0.9395	0.9417	0.0647
Cygnus	11	321.246	38.7083	0.3831	0.3829	0.0269
Delph	10	310.483	14.4919	1.2506	1.2501	0.0373
Dipper	6	185.171	55.6542	0.7716	0.7715	0.0288
Hercules	14	269.442	29.2525	0.0727	0.0719	0.0426
Lyra	17	288.038	35.6058	0.3455	0.3485	0.0476
Taurus	24	68.154	15.8669	0.3845	0.3847	0.0116
Vega	13	283.629	36.9017	0.7956	0.7957	0.0459

### B. Star identification accuracy

The star identification accuracy of the proposed method has been compared with three algorithms, namely Liebe method [3], Geometric voting method [16] and Pyramid method [8]. All these methods have been implemented on the same computer running at 2.99GHz clock rate. The camera attitude is set at right ascensions from 0 to 360 degrees, declinations from -90 to 90 degrees with a 2.5 degrees step. The total number of simulations for each method is 10368. The test results are summarized in TABLE 2.

From TABLE 2, it is observed that the proposed algorithm has a small catalogue. It stores 1631 star entries in its database. In this case, each star entry consists of the number of neighbouring star to the reference star, and the distance between reference and neighbouring stars in ascending order. The database of Liebe method also consists of 1631 entries corresponding to 1631 star patterns. For the Liebe method, each entry consists of three features namely the distances from reference star to two nearest stars, and the angular distance between them. The geometric voting method has the largest database as it stores 45692 combinations of angular separation

between star pairs that are less than the field of view of camera. The pyramid method has similar database to the geometric voting method.

TABLE 2. BENCHMARKING OF STAR RECOGNITION METHODS

	Liebe [3]	Geometric Voting [12]	PyramidMethod[8]	Proposed Method
Number of stars entries	1631	45692	45692	1631
Catalogue size	55.2kB	2120kB	2120kB	30kB
Accuracy	93.05%	86.11%	94.69	95.07%
Average number of returned star	77	11	10	7
Unique star identity	2.82%	1.04%	2.3%	6.2%

The star recognition accuracy is measured by the number of correct star recognition sample over the total number of tests (i.e. 10368). As illustrated in the TABLE 2, the proposed method has 95.07% average accuracy, which is 2% to 9% better than the others.

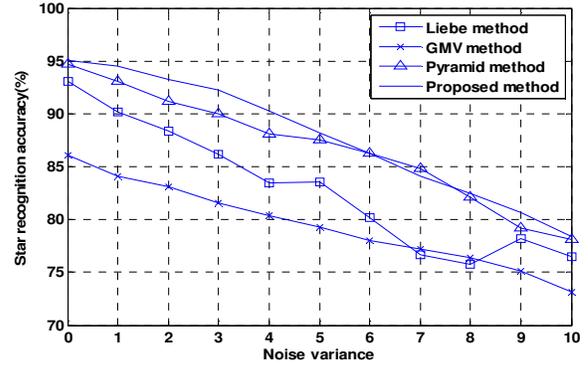


Figure 7. Star pattern recognition accuracy with effects of position noise

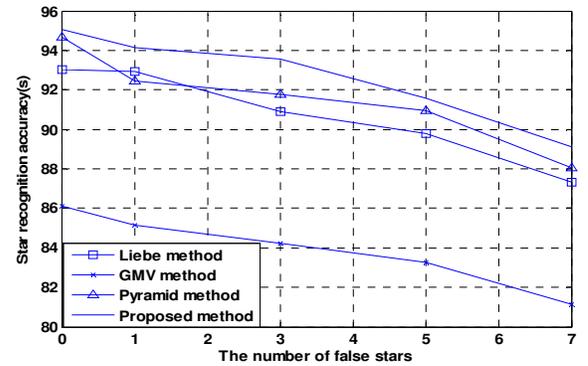


Figure 8. Star pattern recognition accuracy versus number of false stars

In this experiment, we evaluate the robustness of star tracker with respect to star position noise and false star noise. The main source of position noise is thermal noise, dark current noise of CMOS image sensors and alignment error of

image sensor. To study the robustness, Gaussian noise with a variance ranging from 0 to 10 was added into the star images. Figure 7 shows the recognition accuracy with respect to the noise variance. The proposed method has an accuracy of 95.07% under the ideal condition. It decreases about 1% with each unit increase in the noise variance. Overall, the accuracy decreases as noise increases. The proposed algorithm consistently identifies stars at greater than 90% up to a noise variance level of 5 units. Its accuracy is 4.86% better than the Liebe algorithm, 8.27% higher than the geometric voting algorithm and 1% better than pyramid method. Figure 8 shows the recognition accuracy with respect to the number of false stars. The number of false stars is injected into images from 1 to 7 stars. The results also show better accuracy (i.e. 2% to 9%) of the proposed method. The reason is because of grid-based image processing.

## V. CONCLUSION

In this paper, a new star pattern recognition approach has been proposed and developed for a star tracker. It overcomes the bottleneck in searching large star catalogue by using a search tree scheme. The experimental results show that it outperforms Liebe and Geometric voting methods in terms of execution speed, accuracy and robustness. The most important contribution is the run-time reduction by 50% as compared to the conventional methods. In addition, its accuracy is 2% better than the Liebe algorithm and 9% higher than the geometric voting algorithm. The proposed method archives 95.07% accuracy with a noise variance level of 5 units, and injected false stars. The calculated satellite attitude will be used for model predictive control [19-20] to control the satellite attitude in the attitude control system.

## REFERENCES

- [1] J. R. Wertz, *Spacecraft attitude determination and control*, Computer Sciences Corporation, Attitude Systems Operation, Springer, 1978.
- [2] M. D. Shuster, and S. D. Oh, "Three axis attitude determination from vector observation", *J. of guidance and control*, vol.4, no.1, 1981.
- [3] C. C. Liebe, "Pattern recognition of star constellations for spacecraft applications," *IEEE Aerosp. Electron. Syst. Mag.*, vol.8, no.1, pp.31-39, Jan. 1993.
- [4] C. C. Liebe, "Star trackers for attitude determination," *IEEE Trans. Aerosp. Electron. Syst.*, vol.10, no.6, pp.10-16, Jun. 1995.
- [5] D. Mortari, J. L. Junkins, and M. A. Samaan, "Lost-in-space pyramid algorithm for robust star pattern recognition," *Advances in the Astronautical Sciences, Guidance, and Control*, vol.107, pp.49-68, Jan. 2001.
- [6] D. Moratri, "k-vector searching techniques", *Advances in the Astronautical Sciences*, vol. 105, pp. 449-464, Jan. 2000.
- [7] B. B. Spratling, and D. Mortari, "A survey of star identification algorithms", *Algorithms*, vol. 2, pp. 93-107, Jun. 2009.
- [8] M. A. Samaan, D. Mortari, and J. L. Junkins, "Recursive mode star identification algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol.41, no.4, pp. 1246-1254, Oct. 2005.
- [9] C. L. Cole, "Fast star pattern recognition using planar triangle", *J. of Guidance, Control, and Dynamics*, vol.29, no.1, Jan. 2006.
- [10] M. Kolomenkin, S. Polak, I. Shimshoni, and M. Lindenbaum, "A geometric voting algorithm for star trackers", *IEEE Trans. Aerosp. Electron. Syst.*, vol.44, no.2, Jul. 2008.
- [11] C. Padgett, and K. K. Delgado, "A grid algorithm for autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol.33, no.1, pp.202-213, Jan. 1997.
- [12] H. Lee, and H. Bang, "Star pattern identification techniques by modified grid algorithm", *IEEE Trans. Aerosp. Electron. Syst.*, vol.43, no.3, Jul. 2007.
- [13] M. Na, D. Zheng, and P. Jia, "Modified grid algorithm for noisy all-sky autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol.45, no.2, pp.516-522, Apr. 2009.
- [14] G. L. Rousseau, J. Bostel, and B. Mazari, "Star recognition algorithm for APS star tracker: oriented triangles", *IEEE Trans. Aerosp. Electron. Syst.*, vol.20, no.2, Feb. 2005.
- [15] D. Baldini, M. Barni, A. Foggi, G. Benelli, and A. Mecocci, "Star-configuration searching for satellite attitude computation", *IEEE Aerosp. Electron. Syst. Mag.*, vol.31, no.2, Apr. 1995.
- [16] J. Hong, and J. A. Dickerson, "Neural network based autonomous star identification algorithm", *J. of Guidance, Control, and Dynamics*, vol.23, no.4, 2000.
- [17] C. Liebe, K. Gromov, and D. Meller, "Toward a Stellar Gyroscope for Spacecraft Attitude Determination" *J. of Guidance, Control, and Dynamics*, vol.27, no.1, 2004.
- [18] S. Howell, *Handbook of CCD astronomy*, Cambridge University Press
- [19] R. Cao and K. S. Low, "A Repetitive Model Predictive Control Approach for Precision Tracking of a Linear Motion System", *IEEE Trans. Industrial Electron.*, vol. 56, no. 6, pp. 1955-1962, June 2009.
- [20] K. S. Low and R. Cao, "Model predictive control of parallel connected inverters for uninterruptible power supplies", *IEEE Trans. Industrial Electron.*, vol. 55, no. 8, pp. 2884-2893, Aug. 2008.