An Autonomous Star Recognition Algorithm with Optimized Database

MINH DUC PHAM
KAY-SOON LOW, Senior Member, IEEE
SHOUSHUN CHEN, Member, IEEE
Nanyang Technological University

A novel star pattern recognition algorithm is presented for satellite attitude determination in the "lost in space" mode. The proposed method improves the speed in searching a large star catalogue by arranging it using a search tree data structure. The algorithm also processes the star image at grid level instead of pixel level to further improve the processing time. The experimental results show that the proposed approach significantly reduces the average run-time by 50% as compared with the conventional methods while still achieving slightly better star recognition accuracy at 95.07%.

Manuscript received April 30, 2011; revised February 7 and August 28, 2012; released for publication October 5, 2012.

IEEE Log No. T-AES/49/3/944592.

Refereeing of this contribution was handled by M. R. Akela.

Authors'address: School of Electrical & Electronic Engineering, Nanyang Technological University, Block S2, 50 Nanyang Avenue, Singapore 639798, Singapore, E-mail: (k.s.low@ieee.org).

I. INTRODUCTION

Star tracker is an optical-electronics device for estimating the spacecraft orientation in space. It can produce 3-axis attitude information without prior knowledge. In a satellite, there are other attitude determination sensors such as magnetic sensor, Sun sensor, and Earth's horizon scanner. However, the star tracker remains the most accurate solution for spacecraft with 20–90 arc second bore sight accuracy [1, 8]. As the stars' positions remain relatively fixed on the Earth centre inertial frame, their positions can be used as reference to determine a spacecraft attitude using methods such as QUEST or TRIAD methods, etc. [2].

The autonomous star pattern recognition algorithm is based on extracting important features of star cluster. Different methods have been proposed [3–12] and they vary in complexity, recognition time, database size, recognition accuracy, and robustness. The Liebe algorithm [3–4] utilizes triple star pattern which is characterized by angular distances from the reference star to the two closest neighbouring stars, and a spherical angle between two neighbouring stars. This triple star pattern is compared with the pattern catalogue to determine the correct star identity.

In [5] Padgett and Delgado introduce a grid-based method that uses bit pattern to recognize stars. The bit pattern is generated by applying a grid layer on the captured star images. Several improved versions of grid-based algorithms have also been proposed. In [6] polar grid is used instead of conventional rectangular grid to reduce position error caused by the rotation of the camera. Hence it increases the robustness against rotational position noise. To tolerate position and magnitude noise, the elastic gray grid algorithm is proposed in [7].

In [8]–[11], Mortari, et al. etc propose the pyramid star pattern recognition algorithm. The pyramid pattern consists of angular distances between four neighbouring stars. The k-vector technique is used for fast searching of the large-scale angular distance catalogue. Another approach is the planar triangle algorithm [12]. The planar triangle algorithm characterizes a triple star pattern by its planar triangle area and triangle polar moment. The method is reported to have a database size 10 times larger than the equivalent angle catalogue. In [14] the geometric voting algorithm is introduced. Its main idea is that each member of the catalogue pairs votes for each member of image pairs. The correct identity of an image star is the one that received the most votes. A second voting scheme is also introduced to confirm the recognized star identity.

Neural network based autonomous star identification has also been investigated [16]. The original angle database is divided into 14 subangle databases that correspond to 14 neural logic networks (NLN). Neural networks are trained using laboratory

TABLE I
Star Tracker Configuration

Field of view	$20^\circ \times 20^\circ$
Resolution	512×512 pixels
Pixels size	7×7 um
Angular resolution	0.0391°
Star catalogue	SAO SKY 2000
Visual magnitude threshold	5 unit
Number of stars	1631

TABLE II
Probability of Stars Captured by Camera

Number of Stars	Probability
3	97.15%
4	94.23%
5	90.59%
6	88.09%

computers before being loaded into the spacecraft on-board computer memory. The star identity is recognized with maximum hidden node value. The memory required to store the weight vectors for NLNs is over 3MB with 17,533 triplets and this is the main disadvantage of the scheme.

In this paper a novel method is proposed that leads to fast and reliable autonomous star pattern recognition. The major innovations reside in two areas: 1) an optimized database for fast search, and 2) parallel search capability. The proposed method introduces search tree data structure which leads to an optimized star pattern catalogue. The idea behind search tree is to narrow down the search region in every iteration to improve the search speed. The search requires only a small number of iterations instead of scanning through the entire catalogue. The worst case run-time is estimated as $O(2\log 2N)$, where N is the length of the star catalogue. In addition, multiple feature vectors can be searched in parallel in the search tree. Hence the search speed is significantly improved. The proposed method requires a small memory of 30 kB, and a short average run-time of 4.85 ms. The proposed method also in-cooperates grid-based image processing to reduce the effect of image noise.

This paper is organised as follow. In Section II the star distribution analysis in the space is presented. Section III introduces the star pattern database generation that is used for the proposed star recognition algorithm. In Section IV a new star recognition method is developed for "lost in space" mode. This section describes the search tree construction based on star pattern database presented in previous section. The single search and parallel search processes are also introduced to identify stars. The simulation and experimental results are then presented in Section V. The proposed approach has been benchmarked with two algorithms, namely, Liebe method [3], geometric voting algorithm [14], and the pyramid method [8]. Finally, Section VI concludes this work.

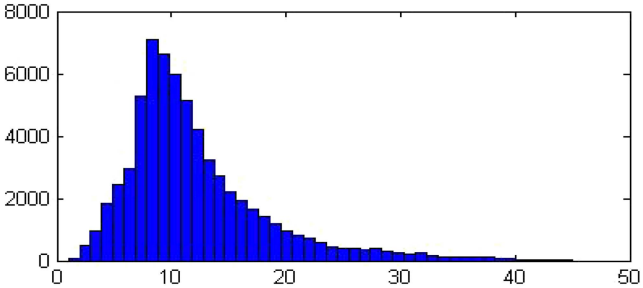


Fig. 1. Star distribution histogram.

II. STAR DISTRIBUTION ANALYSIS

Similar to most star tracking algorithms, the proposed method requires at least 3 stars: a reference star and at least 2 neighbouring stars. To investigate the probability that the camera frame can capture at least 3 stars, a star distribution analysis is presented in this section. The star tracker configuration used in this study is given in Table I. It has a 20° optical field of view (FOV), 7 um square sensor pixel size, and 512×512 pixels image resolution.

To perform the analysis the star camera direction is rotated over the sky. For each star camera direction, the number of stars that is within the camera's FOV is recorded. The SAO J2000 star catalogue is used for star generation. This catalogue consists of 258,996 star entries [1] with visual magnitude ranging from $M = 1$ to 10 stellar magnitude units ($M = 1$ are the brightest stars). A program has been developed to simulate the star images captured by star tracker for a given attitude. The star camera's attitude is rotated from 0 to 360° with reference to the right ascension and varied from −90° to 90° declinations with an angle increment of 2°. This results in 64801 captured star images. Figure 1 shows the star distribution histogram. From the histogram, it is observed that the number of visible stars in the camera's FOV ranges between 0 and 45. The overall average is 11.8475 visible stars per frame. Table II shows the probability derived from the histogram that the camera captures from 3 to 6 stars. From Table II it is concluded that the star tracker has a probability of 97.15% to capture at least 3 stars.

III. DATABASE GENERATION

For a star tracker the star pattern database is first processed on the ground before it is loaded into the star tracker on-board memory. A typical camera's sensitivity is $M < 5$. With this constraint we extract only those stars that have visual magnitude less than 5 units from the SAO J2000 catalogue. Moreover, double and multiple stars are discarded as they appear too close to each other to allow the CMOS image sensors to differentiate them correctly. The reduced

star catalogue **T** now consists of 1631 stars. Next, a star pattern database **D** is generated to store the important features describing each reference star in the star catalogue **T**. The database generation is illustrated in Fig. 2 and its procedure is described as follows.

1) The shaded region in Fig. 2(a) shows the FOV of the star camera. The star images are projected on the image sensor plane as shown in Fig. 2(b). The star near the centre of the FOV is called the reference star S_1, and the rests are the neighbouring stars S_2, S_3, S_4, S_5; S_6. For each reference star S_i in the catalogue **T**, we first find all the neighbour stars S_j such that the distance between the neighbour star and the reference star is less than half the FOV length d_f. Thus given $\mathbf{V}_i$ and $\mathbf{V}_j$ being the position vectors of stars $\mathbf{S}_i$ and $\mathbf{S}_j$ in the Earth centre inertial frame,

$$\mathbf{V}_i = [x_i; y_i; z_i]$$

$$\mathbf{V}_j = [x_j; y_j; z_j]$$

$$d_{ij} = |\mathbf{V}_i \cdot \mathbf{V}_j| \le \frac{d_f}{2}$$

2) The camera attitude matrix **C** is configured such that the camera bore sight vector (z axis) is aligned with the reference star, i.e.

$$\mathbf{C} = \begin{bmatrix} \mathbf{c}_x \\ \mathbf{c}_y \\ \mathbf{c}_z \end{bmatrix}$$

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

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

$$\mathbf{V}_i^{\text{cam}} = [x_i^{\text{cam}}; y_i^{\text{cam}}; z_i^{\text{cam}}] = \mathbf{C} \cdot \mathbf{V}_i$$

$$\mathbf{V}_j^{\text{cam}} = [x_j^{\text{cam}}; y_j^{\text{cam}}; z_j^{\text{cam}}] = \mathbf{C} \cdot \mathbf{V}_j.$$

4) The star attitudes $\mathbf{V}_i^{\text{cam}}$ and $\mathbf{V}_j^{\text{cam}}$ are projected onto the centre of the CMOS image sensor plane as in Fig. 2(b). The star position (Px_i, Py_i) and (Px_j, Py_j) on the image sensor can be determined as follows:

$$Px_i = \frac{f}{\rho}\frac{x_i^{\text{cam}}}{z_i^{\text{cam}}} + \frac{\mu_v}{2}, \qquad Py_i = \frac{f}{\rho}\frac{y_i^{\text{cam}}}{z_i^{\text{cam}}} + \frac{\mu_h}{2}$$

$$Px_j = \frac{f}{\rho}\frac{x_j^{\text{cam}}}{z_j^{\text{cam}}} + \frac{\mu_v}{2}, \qquad Py_j = \frac{f}{\rho}\frac{y_j^{\text{cam}}}{z_j^{\text{cam}}} + \frac{\mu_h}{2}$$

where f, ρ, μ_v, and μ_h are the optical focal length, the pixel size, and the vertical and horizontal dimensions of the CMOS image sensor.

5) The star visual magnitudes M are also included in the simulated images as shown in Fig. 2(c). In this example, the stars S_1, S_2, S_6 are brighter. Hence their projections are larger. The image noise is also included based on the following Gaussian distribution

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

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 (σ_x,σ_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 [19].

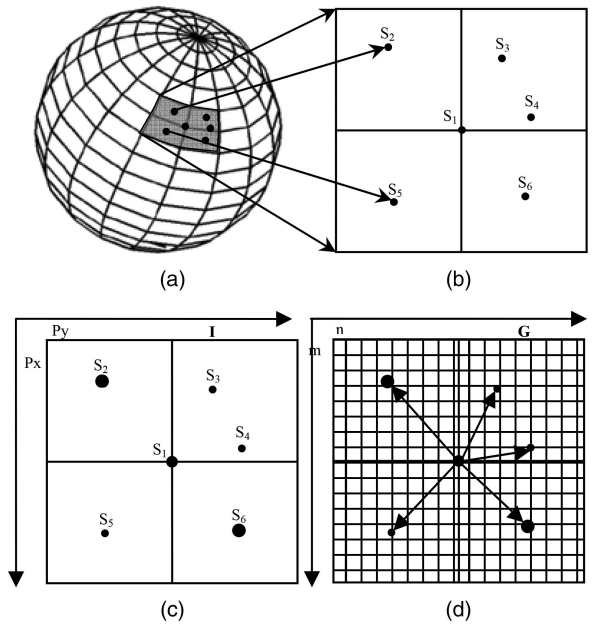


Fig. 2. Star pattern database generation. (a) Shaded rectangle shows star camera's FOV. (b) Star images are projected on image sensor plane. (c) Effect of star magnitude is reflected on projection. (d) Square grids are applied to star images.

6) A square grid $g \times g$ ($g > (\sigma_x,\sigma_y)$ maximum position variance) is applied to the star image. This is shown in Fig. 2(d). In this study g is set to 2 pixels. The pixel $G(m,n)$ is set to 1 if a star projection falls onto this pixel with a total pixel intensity $I(Px,Py)$ greater than the threshold ζ, otherwise $G(m,n)$ is set as 0 as follows:

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

7) The planar distances $\{D_k,\ k = 1,\ldots,N\}$ 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_{Si} of reference star S_i is created as $f_{Si} = \{Si, N, \{D_k,\ k =$

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

Fig. 3. Structure of star pattern database D.

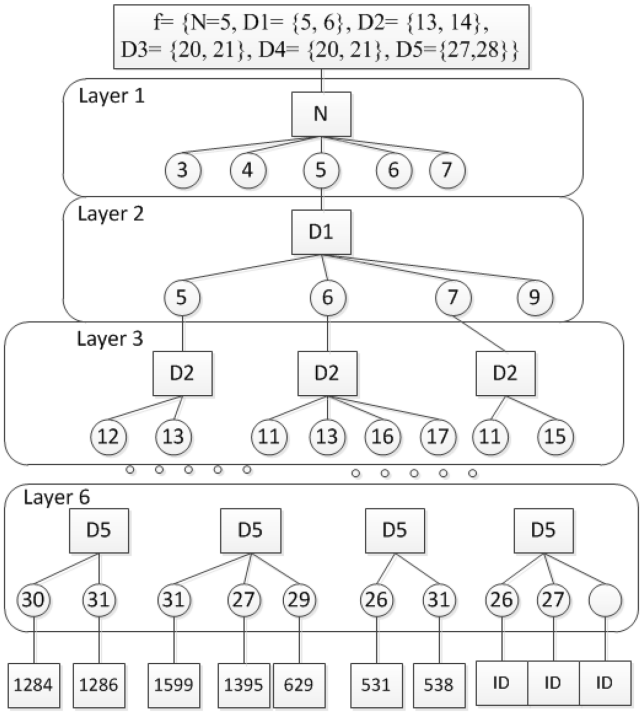


Fig. 4. Example of search tree structure.

$1,\ldots,N\}\}$ and appended into the star pattern database **D**.

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 3 illustrates one such star pattern database **D**. In Fig. 3 the first column is the 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 D_1, D_2, D_3, D_4, and 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 $N = 14$. Consequently, a longer sorting time is expected if there are a large number of neighbour stars. However the long sorting time does not affect the star recognition performance as it is done offline on the ground.

IV. STAR PATTERN RECOGNITION METHOD

A. Search Tree Construction and Single Search Process

Before the search process the search tree is constructed based on the star pattern catalogue. As illustrated in Fig. 4, 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 layers 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 number. 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 the 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 Fig. 4 the feature vector implies that $N = 5$ is chosen in the first layer. The search path is narrowed down to subtree $N = 5$. This subtree 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 subtree in the third layer. The third layer has decision values $\{D_2 = 11,13,16,17\}$. From the feature vector, $D_2 = 13$ is chosen. The search process is continued until the sixth layer which is $D_5 = 27$. Finally, the star identity $S_i = 1395$ is returned.

B. Parallel Search

Due to the noise in image acquisition process, the feature vector is contaminated with position noise. To take into account the position noise, a tolerance level η is added into the feature vector

$$f_{Si} = \{N, \{D_k + \eta,\ k = 1,\ldots,N\}\}.$$

Thus the star pattern search becomes a searching of multiple feature vectors in the search tree. Higher

tolerance level (high noise) η implies more search paths. Each feature vector is searched in a copy of star catalogue D. The search processes can be executed in parallel in the search trees. Hence the total run-time is similar to the single search process. Figure 4 illustrates an example with $\eta = +1$ and a feature vector $f_{Si} = \{N = 5,\ D_1 = \{5,6\},\ D_2 = \{13,14\},\ D_3 = \{20,21\},\ D_4 = \{20,21\}, D_5 = \{27,28\}\}$. The combinations of feature vectors are

$$f_1 = \{N = 5,\ D_1 = 5,\ D_2 = 13,\ D_3 = 20\}$$
$$f_2 = \{N = 5,\ D_1 = 5,\ D_2 = 13,\ D_3 = 21\}$$
$$f_3 = \{N = 5,\ D_1 = 5,\ D_2 = 14,\ D_3 = 20\}$$
$$f_4 = \{N = 5,\ D_1 = 5,\ D_2 = 14,\ D_3 = 21\}$$
$$f_5 = \{N = 5,\ D_1 = 6,\ D_2 = 13,\ D_3 = 20\}$$
$$f_6 = \{N = 5,\ D_1 = 6,\ D_2 = 13,\ D_3 = 21\}$$
$$f_7 = \{N = 5,\ D_1 = 6,\ D_2 = 14,\ D_3 = 20\}$$
$$f_8 = \{N = 5,\ D_1 = 6,\ D_2 = 14,\ D_3 = 21\}.$$

The feature vectors f_3, f_4, f_7, f_8 do not return valid star identity in the final layer, because there is no decision value for $D_2 = 14$ in layer 3. Similarly feature vectors f_1, f_2, f_6 do not return valid star identity in the final layer, because there is no decision value for D_3 in layer 4. This happens when suitable decision nodes do not exist. Finally, only feature vector f_5 could return a valid star identity $S_i = 1395$.

In order to implement the parallel search, the hardware platform must be able to support multi-thread and parallel memory access. With the help of operating system and special hardware platform with parallel memory accesses, this can be easily implemented using a device such as field programmable gate array (FPGA).

C. Autonomous Star Pattern Recognition

When the star tracker is activated, it enters into the "lost-in-space" operation mode. The camera first captures star images and then it performs the following image processing tasks.

1) Image preprocessing is performed to improve image quality. The process includes noise filtering, image threshold, and star labeling to find the number of stars N in the image.

2) The star centroid (Px_i, Py_i) of star S_i is calculated based on weighted pixel intensity, with $I(Px,Py)$ being the intensity of pixel (Px,Py) in the star image as follows:

$$Px_i = \frac{\sum I(Px,Py)\cdot Px}{\sum I(Px,Py)}, \qquad (i = 1,\ldots,N)$$

$$Py_i = \frac{\sum I(Px,Py)\cdot Py}{\sum I(Px,Py)}, \qquad (i = 1,\ldots,N).$$

3) If $N < 3$, there are an insufficient number of stars for the star recognition. The star tracker will continue to perform image acquisition steps until the acquired number of stars is enough for star recognition. If $N \geq 3$, the brightest star that is nearest to the image centre $(\mu_v/2, \mu_h/2)$ is chosen as the reference star S_{ref}. Its coordinate $(Px_{\text{ref}}, Py_{\text{ref}})$ should satisfy the following condition:

$$\left|(Px_{\text{ref}}, Py_{\text{ref}}) - \left(\frac{\mu_v}{2}, \frac{\mu_h}{2}\right)\right| = \min_{i=1\ldots N}\left|(Px_i, Py_i) - \left(\frac{\mu_v}{2}, \frac{\mu_h}{2}\right)\right|.$$

4) A square grid $g \times g$ is applied to the image as shown in Fig. 2(d). In this study g is set to 2 pixels. The cell $G(m,n)$ is set to 1 if a star projection falls into this cell, otherwise $G(m,n)$ is set as 0. This step is similar to Section III (6).

5) Calculate the planar distances between the reference star and the neighbour stars $D_j = |(Px_j, Py_j) - (Px_{\text{ref}}, Py_{\text{ref}})|$ and sort those distances in ascending order and form the feature vector $f = \{N, D_1, D_2, D_3, D_4, \ldots, D_{n-1}\}$.

6) Noise tolerance η is added into the feature vector to take into account the image noises as $f = \{N + \eta, D_1 + \eta, D_2 + \eta, D_3 + \eta, D_4, \ldots, D_{n-1} + \eta\}$.

7) These feature vectors are searched in search tree. The search tree returns the star candidates for reference star S_{ref}. If no star candidates are returned, an error message is returned. The star recognition method is then applied to the next captured image frame.

8) If a unique star candidate is returned, the proposed method is repeated on the next captured image to confirm the result. If the two identified stars are matched, a SUCCESS message is returned. If multiple candidates are returned, the proposed method is applied to the subsequent brightest stars. The results are used to confirm a unique identity.

9) Once a star identity S_{ref} is confirmed, several attitude determination methods can be used to estimate the satellite attitude, for examples QUEST or TRIAD methods [2]. This attitude information is then used in tracking mode of the star tracker.

V. RESULTS

To benchmark the performance of the proposed method with the available methods, several approaches have been implemented and executed on the same computer platform. The test platform in our laboratory is capable of simulating the star images with given camera attitude parameters (quaternion, rotation matrix, Euler angles), and camera configurations (FOV, pixels size, focal length).

In the following study a standard camera configuration as given in Table I is used for comparing the star recognition algorithms.

TABLE III
Benchmarking of Star Recognition Methods

	Liebe [3]	Geometric Voting [14]	Pyramid Method [8]	Proposed Method
Number of stars entries	1631	45692	45692	1631
Catalogue size	55.2 kB	2120 kB	2120 kB	30 kB
N	3	3	3	3
Average run-time	13.9 ms	73.33 ms	24.41 ms	4.85 ms
Minimum run-time	0.56 ms	0.57 ms	23.33 ms	0.59 ms
Maximum run-time	23.56 ms	931.7 ms	43.21 ms	13.23 ms
Variance run-time	7.02 ms	73.33 ms	2.1 ms	7.33 ms
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%

A. Performance Evaluation

The objective of this study is to evaluate the catalogue size, accuracy, and run-time of the star recognition approaches. The performance of the proposed method has been compared with three available algorithms, namely:

1) Liebe method [3],
2) geometric voting method [14],
3) pyramid method [8].

All these methods have been implemented on the same computer running at 2.99 GHz. The camera attitude is set at right ascensions and varied from 0 to 360 deg, declinations from −90 to 90 deg with a 2.5 deg step. The total number of simulations for each method is 10368. The test results are summarized in Table III. The performance of each algorithm is evaluated based on the following parameters:

1) the number of stars entries in the catalogue,
2) the catalogue size,
3) run-time of algorithm (minimum, maximum, and variance),
4) star identification accuracy,
5) robustness of star recognition with respect to position noise and false stars.

From Table III 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 stars to the reference star, and the distance between them in ascending order. The database of the 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 the reference star to the two nearest stars, and the angular distance between them. The geometric voting method has the largest database as it stores 45692 combinations of angular separations between star pairs that are within the FOV of the camera. The pyramid method has a similar database of 45692 entries.

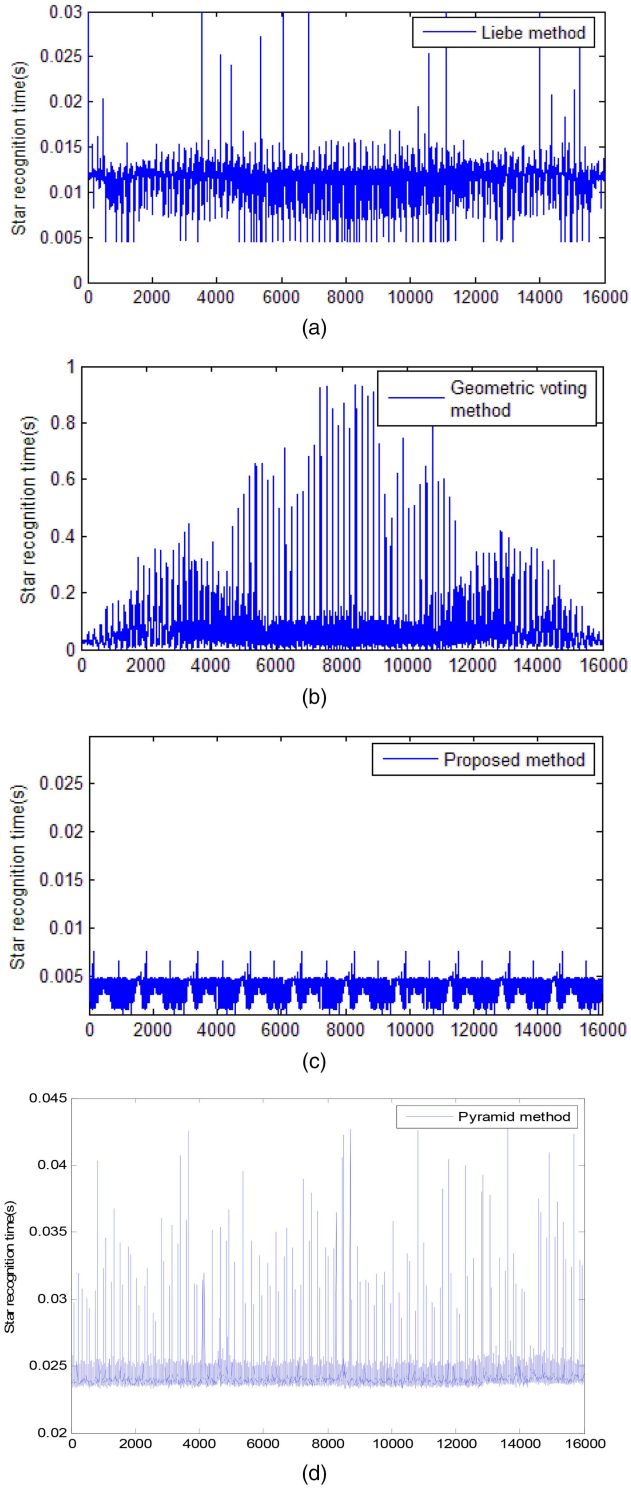


Fig. 5. Run-time for star recognition versus star index. (a) Liebe method. (b) Geometric voting method. (c) Proposed method. (d) Pyramid method.

The run-times of the algorithms are measured from the instant that the star centroids pass into the star recognition program until the star identities are returned. It excludes image preprocessing time and the computation of the satellite's attitude. The star pattern recognition times are shown in Fig. 5. It is clear that the proposed method is the fastest with an

average run-time of 4.85 ms. This is the consequence of utilizing search tree to locate the star identity. In the Liebe method the search process is executed by scanning from top to bottom of the database to find the correct star identity. In the worst case the whole database may be scanned until the last entry. Hence its average run-time is longer at 13.9 ms. The geometric voting method uses the k-vector search scheme [8] to locate the star candidates for each angular distance. The k-vector search method is able to locate star candidates in the database instantly. However, the voting table construction process accounts for a long run-time of 73.33 ms. The pyramid method also uses the k-vector search to locate lists of star candidates. Its run-time is determined by finding a unique match from the star lists which account for 24.41 ms. In summary this experiment demonstrates that the run-time can be reduced by 50% or more with the proposed scheme.

There are two reasons for this speed improvement: 1) an optimized database that narrows down the search region after every iteration (due to the small catalogue size, the run-time is significantly reduced), and 2) parallel search for multiple feature vectors in the catalogue.

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

Another important evaluation is the number of returned star candidates after the first run. Normally, the star recognition method will return multiple star candidates. A higher returned number means lower probability of correct recognized star. Moreover, a smaller returned number reduces the total run-time. Table III shows the average number of returned stars for 16000 simulation cases. The proposed method has the lowest number of 7 star candidates. The Liebe method returns 77 while the geometric voting method returns 11 and the pyramid method returns 10. Furthermore, the proposed method returns a unique star identity in 6.2% of simulations as compared to 2.82% for the Liebe method and 1.04% for the geometric voting method, 2.3% for the pyramid method. When there are multiple star candidates, the star recognition procedure is repeated until a unique star identity is returned.

B. Robustness Study

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

To study the robustness Gaussian noise with a variance of 0 to 10 and a zero mean was added into the star images. Figure 6 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 the 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 the pyramid method. Figure 7 shows the recognition accuracy with respect to the number of false stars. The number of false stars is injected into images at random locations from 1 to 7 stars. The results also show better accuracy (2% to 9%) of the proposed method. The reason is because of the grid-based image processing. The star projection on the image plane is very sensitive to image noise. Hence pixel-based image processing is less tolerant to image noise.

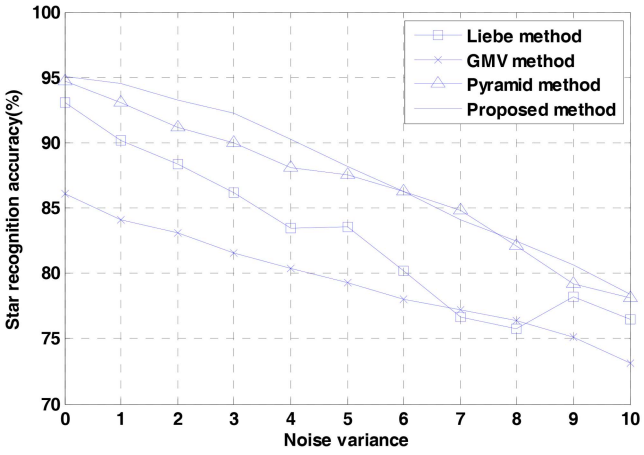


Fig. 6. Star pattern recogniton accuracy with effects of position noise.

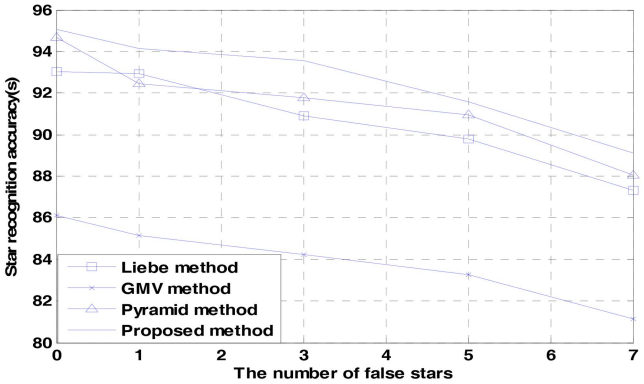


Fig. 7. Star pattern recogniton accuracy versus number of false stars.

VI. CONCLUSION

In this paper a new star recognition approach has been proposed and developed for a star tracker. The proposed approach overcomes the bottleneck

in searching large star catalogues by using a search tree scheme. The experimental results show that it outperforms Liebe, geometric voting, and pyramid methods in terms of execution speed, accuracy, and robustness. The most important contribution is the run-time reduction by 50% as compared with 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. It achieves the objective of fast and accurate search at the expense of using a more specialize hardware such as the FPGA. Moreover, there is a slight increase in memory requirements if noise level is high.

REFERENCES

[1] Wertz, J. R. (Ed.)
Spacecraft Attitude Determination and Control.
New York: Springer, 1978.

[2] Shuster, M. D. and Oh, S. D.
Three-axis attitude determination from vector observation.
Journal of Guidance and Control, **4**, 1 (1981), 70–77.

[3] Liebe, C. C.
Pattern recognition of star constellations for spacecraft applications.
IEEE Aerospace and Electronic Systems Magazine, **8**, 1 (1993), 31–39.

[4] Liebe, C. C.
Star trackers for attitude determination.
IEEE Aerospace and Electronic Systems Magazine, **10**, 6 (1995), 10–16.

[5] Padgett, C. and Delgado, K. K.
A grid algorithm for autonomous star identification.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 1 (1997), 202–213.

[6] Lee, H. and Bang, H.
Star pattern identification technique by modified grid algorithm.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 3 (2007), 1112–1116.

[7] Na, M., Zheng, D., and Jia, P.
Modified grid algorithm for noisy all-sky autonomous star identification.
IEEE Transactions on Aerospace and Electronic Systems, **45**, 2 (2009), 516–522.

[8] Mortari, D., Junkins, J. L., and Samaan, M. A.
Lost-in-space pyramid algorithm for robust star pattern recognition.
Advances in the Astronautical Sciences, Guidance, and Control, **107** (2001), 49–68.

[9] Mortari, D. and Neta, B.
k-vector searching techniques.
Advances in the Astronautical Sciences, **105**, 1 (2000), 449–464.

[10] Spratling, B. B. and Mortari, D.
A survey of star identification algorithms.
Algorithms, **2**, 1 (2009), 93–107.

[11] Samaan, M. A., Mortari, D., and Junkins, J. L.
Recursive mode star identification algorithms.
IEEE Transactions on Aerospace and Electronic Systems, **41**, 4 (2005), 1246–1254.

[12] Cole, C. L.
Fast star pattern recognition using planar triangle.
Journal of Guidance, Control, and Dynamics, **29**, 1 (2006), 64–71.

[13] Lamy Au Rousseau, G., Bostel, J., and Mazari, B.
Star recognition algorithm for APS star tracker: Oriented triangles.
IEEE Aerospace and Electronic Systems Magazine, **20**, 2 (2005), 27–31.

[14] Kolomenkin, M., et al.
A geometric voting algorithm for star trackers.
IEEE Transactions on Aerospace and Electronic Systems, **44**, 2 (2008), 441–456.

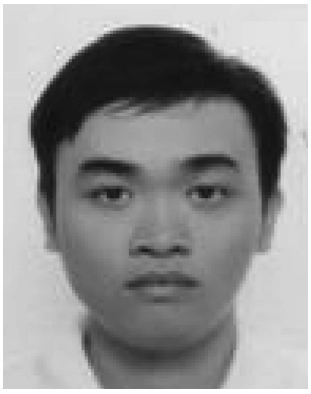
[15] Baldini, D., et al.
Star-configuration searching for satellite attitude computation.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 2 (1995), 768–777.

[16] Hong, J. and Dickerson, J. A.
Neural network based autonomous star identification algorithm.
Journal of Guidance, Control, and Dynamics, **23**, 4 (2000), 728–735.

[17] Liebe, C., Gromov, K., and Meller, D.
Toward a stellar gyroscope for spacecraft attitude determination.
Journal of Guidance, Control, and Dynamics, **27**, 1 (2004), 91–99.

[18] Wei, B.
Space Vehicle Dynamics and Control (AIAA Education Series).
Reston, VA: AIAA, 1998.

[19] Howell, S.
Handbook of CCD Astronomy.
New York: Cambridge University Press, 2006.



Minh Duc Pham received his B.Eng. degree in electrical and electronic engineering from Nanyang Technological University, Singapore in 2010. He has submitted his thesis and is expecting to receive his Master of Engineering degree from the same university.

He is currently a research engineer with the Satellite Research Centre, Nanyang Technological University. His main research interests include attitude determination sensing system and image processing.

Kay-Soon Low (M'88—SM'00) received his B.Eng. degree in electrical engineering from the National University of Singapore, Singapore, and his Ph.D. degree in electrical engineering from the University of New South Wales, Sydney, Australia.

He joined the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 1994, as a lecturer and subsequently became an associate professor. His funded projects are in the area of satellite system, wireless sensor network and clean energy system. He has served as a consultant to a number of companies. Presently, he is the Centre Director of the Satellite Research Centre.

Dr. Low has 16 patents.

Shoushun Chen (M'05) received his B.S. degree from Peking University, M.E. degree from Chinese Academy of Sciences and Ph.D. degree from Hong Kong University of Science and Technology in 2000, 2003, and 2007, respectively.

He held a post-doctoral research fellowship in the Department of Electronic & Computer Engineering, Hong Kong University of Science and Technology for one year after graduation. From February 2008 to May 2009 he was a post-doctoral research associate within the Department of Electrical Engineering, Yale University. In July 2009, he joined Nanyang Technological University as an assistant professor.

Dr. Chen serves as a technical committee member of Sensory Systems, IEEE Circuits and Systems Society (CASS); Associate Editor of *IEEE Sensors Journal*; Associate Editor of *Journal of Low Power Electronics and Applications*; Program Director (Smart Sensors) of VIRTUS, IC Design Centre of Excellence; regular reviewer for a number of international conferences and journals such as TVLSI, TCAS-I/II, TBioCAS, TPAMI, Sensors, TCSVT, etc.