Hindawi Publishing Corporation Mathematical Problems in Engineering Volume 2015, Article ID 982420, 11 pages http://dx.doi.org/10.1155/2015/982420



Research Article

Lost-in-Space Star Identification Using Planar Triangle Principal Component Analysis Algorithm

Fuqiang Zhou and Tao Ye

Key Laboratory of Precision Opto-Mechatronics Technology, Ministry of Education, Beihang University, Beijing 100191, China

Correspondence should be addressed to Fuqiang Zhou; zfq@buaa.edu.cn

Received 2 March 2015; Revised 6 May 2015; Accepted 25 May 2015

Academic Editor: Giacomo Innocenti

Copyright © 2015 F. Zhou and T. Ye. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is a challenging task for a star sensor to implement star identification and determine the attitude of a spacecraft in the lost-in-space mode. Several algorithms based on triangle method are proposed for star identification in this mode. However, these methods hold great time consumption and large guide star catalog memory size. The star identification performance of these methods requires improvements. To address these problems, a star identification algorithm using planar triangle principal component analysis is presented here. A star pattern is generated based on the planar triangle created by stars within the field of view of a star sensor and the projection of the triangle. Since a projection can determine an index for a unique triangle in the catalog, the adoption of the k-vector range search technique makes this algorithm very fast. In addition, a sharing star validation method is constructed to verify the identification results. Simulation results show that the proposed algorithm is more robust than the planar triangle and P-vector algorithms under the same conditions.

1. Introduction

Accurate attitude information is essential for spacecraft autonomous navigation. This plays a very important role in spacecraft control systems. Many different strategies can be employed to determine the attitude. One of the most widely used attitude measurement devices is a star sensor. Other sensors for determining attitude include a sun sensor and magnetometer [1–3]. A star sensor can provide precision attitude without prior information. It obtains the attitude directly from the stars within its field of view (FOV). As long as at least two stars in the FOV are identified in the predefined onboard star catalog, the attitude of the spacecraft can be determined using the TRIAD algorithm [4].

A classic star sensor works in two basic modes. When it is first activated or after a system failure, it has no prior information regarding the attitude. This is called the "lost-in-space" (LIS) mode. Once the attitude is determined, the star sensor changes to the tracking mode. It employs previously obtained information to predict the current attitude. This paper addresses attitude determination in the LIS mode. The most important function for a star sensor to determine the attitude is star identification. Star identification in the LIS

mode is much more difficult than in the tracking mode [5–8]. As a result, there are many different algorithms for star identification within an FOV (see review paper [9]). We focus our research on triangle algorithms. In this study, we explore a modified triangle algorithm employing a highly compressed feature catalog memory. It provides a significant improvement in terms of star identification time and rate.

The first triangle algorithm was developed by Liebe [10]. Liebe stored guide triangles that are formed by guide stars. These are used for matching observation triangles within the FOV. If the guide triangle matches any observation triangle, then the star identification succeeds. Even though only 1,000 stars were selected from the Smithsonian Astrophysical Observatory (SAO) [11] star catalog to form the guide triangles in this algorithm, there are still 185,000 star triangles. The recognition rate for this algorithm is 94.6% and the all-sky star identification average time is about 10 s.

Based on the traditional triangle algorithm, Mortari et al. proposed a pyramid algorithm [12] that matches four stars by ordering the angular distance between the stars to form an index. The adoption of the k-vector range search technique makes this algorithm fast. This algorithm can estimate the frequency of false matches, but no statistical

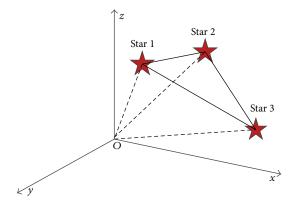


FIGURE 1: Planar triangle from combination of three stars.

analysis is given for the errors in the four stars' angular distance measurements.

Then, Cole and Crassidis made some changes in terms of feature selection for planar triangles method. The triangle is a combination of three stars, as shown in Figure 1. The area and polar moment of the triangles are used to determine what triangle is being examined by the star sensor [13]. Using these two features, the number of similar solutions will be rapidly reduced. Unlike the pyramid algorithm, this algorithm can give precise statistical analysis of the polar moment or area errors and can thus provide an accurate range for the observation triangles matching the catalog triangles. However, there are certain drawbacks to this algorithm. A large memory is required to hold the triangle data. For example, when the star sensor has an 8×8 -degree FOV, the area and polar moment calculated can generate a guide star catalog of 167 MB. In addition, this algorithm is affected by star positional noise in the image.

Yang et al. developed a *P*-vector star identification algorithm [14]. The Quine method [15], which can determine a star with only one angular distance triangle, is assimilated in this algorithm. The three edges of the triangle are used to generate a parameter *P*. Comparison of the value of the parameter *P* of the observation triangle with a triangle in the catalog can identify the star. Considerable time and memory can be saved with this algorithm. However, it also has some drawbacks. Because the feature triangle is a combination of the guide star and the two brightest stars around the guide star, the star identification rate is easily affected by magnitude noise. Furthermore, if the stars within the star sensor FOV cannot form a triangle corresponding to a triangle in the catalog, the algorithm will not work.

As mentioned previously, accurate error analysis, storage, speed, and the ability to manage noise are the main problems with traditional triangle star identification methods. In this study, we develop a star identification algorithm called the planar triangle principal component analysis (PTP) algorithm that performs well against star positional noise and magnitude noise. The algorithm, which incorporates the advantages of the planar triangles and *P*-vector algorithms, can calculate the range of the measurement error precisely and identify a star faster. It creates planar triangles from the

stars in the FOV similar to the planar triangle algorithm and then employs the three sides of the triangles to build vectors projected on the projection line to compute the projection as in the *P*-vector algorithm. The stars selected to construct the planar triangle are satisfied with a specific condition that is more flexible than in the P-vector algorithm. Thus, less memory is needed to store the planar triangles in the catalog database. The number of reference triangles is sufficient to be used for matching the observed triangles in the FOV of the star sensor. The adoption of the k-vector range search technique makes the proposed algorithm very fast. In addition, the proposed sharing star validation method can effectively verify the identification results. In simulations, using the stars brighter than magnitude 6.0 My and 10×10 -degree FOV, our algorithm obtains an identification rate of 96.79% from the statistics of 10,000 random sensor orientations at a positional noise level of 1 pixel and a magnitude level of 0.5 Mv.

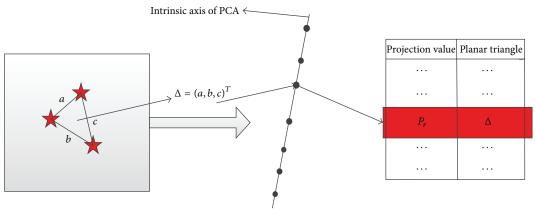
The rest of the paper is organized as follows. In Section 2, the algorithm is described in detail, including guide star catalog construction, matching, star identification, and identification result validation. In Section 3, a simulation and an analysis of the results are provided. These focus on discussing the number of slave stars in guide database construction and comparing the robustness of the different algorithms (our PTP algorithm, *P*-vector algorithm, and planar triangle algorithm) with regard to the algorithm parameters, including positional noise, magnitude noise, and the number of stars in the FOV. We draw our conclusion in Section 4.

2. Algorithm Description

The central concept of our PTP algorithm is that the planar triangles composed of the stellar unit vectors in the starsensor-body coordinate system and the celestial coordinate system are congruent. The concept behind our PTP algorithm is to use the "three sides equal in congruent triangles" principle to recognize the pattern of the stars. Matching the three sides of the triangles simultaneously requires significant computation and results in a redundant match. However, by taking the three sides of the triangle as a three-dimensional vector and using the principal component analysis (PCA) algorithm [16] to compute the projector line, all planar triangles are projected to the projector line to compute the projections. By using the projection of the triangle in the image to find a matched triangle in the guide star database, the computation effort is greatly reduced. The simple diagram of the projection geometry is shown in Figure 2.

Our PTP algorithm has three steps. First, the guide database is constructed using PCA to train the planar triangles. Second, the projection of the planar triangles is computed to search for the best candidate triangle in the guide database. The third step validates the star identification and reports the identification result or failure. The functional steps of our PTP algorithm are shown in Figure 3.

2.1. Guide Database Construction. The guide database consists of two parts: reference catalog and feature pattern database. The basic information (ID, apparent position, and



Star tracker frame

The projection of planar star triangle on the Find a matched triangle in the guide star intrinsic axis of PCA database

Represents the projection value of the planar star triangle

FIGURE 2: The simple diagram of the projection geometry.



FIGURE 3: Functional steps of PTP algorithm.

magnitude) of the stars is stored in the reference catalog. The stars are selected from the standard SAO Star Catalog J2000 [11]. The SAO catalog contains the information of 258,996 stars, including equatorial coordinates (including right ascension and declination), B1950 positions, proper motions, and apparent magnitude of the stars. The catalog is sorted by increasing ID number. For celestial navigation, we focus on the information about the equatorial coordinates and apparent magnitude of the stars. Given the attitude and the optical parameters of a star sensor, it is convenient for us to simulate a star image by using the catalog.

Here we use a wide, 10×10 -degree FOV and assume that the star sensor is capable of capturing stars brighter than 6.0 Mv. A total of 5103 stars were picked from the star catalog. The pattern database creation consists of three steps: (1) feature triangle selection; (2) computation of the projection line for all of the feature triangles; (3) pattern database creation.

(1) Feature Triangle Selection. As shown in Figure 4, $\mathbf{r_1}$, $\mathbf{r_2}$, and $\mathbf{r_3}$ are three unit vectors to the stars within the FOV, allowing the sides of the planar triangle to be calculated by

$$a = \|\mathbf{r}_1 - \mathbf{r}_2\|_2,$$

$$b = \|\mathbf{r}_2 - \mathbf{r}_3\|_2,$$

$$c = \|\mathbf{r}_1 - \mathbf{r}_3\|_2,$$
(1)

where $\|\cdot\|_2$ denotes the 2-norm of a vector.

As mentioned previously, 5103 stars brighter than 6.0 Mv were selected from the standard catalog. We take these stars from the reference catalog as the master stars. As shown

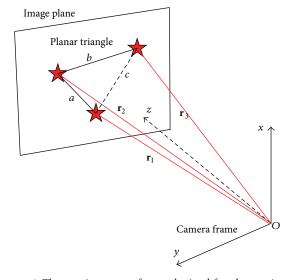


FIGURE 4: Three unit vectors of stars obtained for planar triangle.

in Figure 5, the apparent position of the master star is considered the optical axis of the star sensor that extracts the stars in a 10×10 -degree FOV. We define the extracted stars as slave stars. The distance between the master star and an arbitrary slave star within an FOV can be calculated using (2), denoted by D_{ms} . With the restriction of the circular field of view, the maximum angle between the master star and the slave star is 5° , and the value of D_{ms} is very small. We will migrate the star database to hardware such as DSP6701; however, double-type data occupy 8 bytes and float-type data occupy 4 bytes in DSP memory. Multiplying by 10000

k	p_r	I_1	I_2	I_3	E_1	E_2	E_3
1481	-340.82761	196653	196659	196774	84.69942	302.27382	257.24929
1482	-340.67816	255962	255998	255973	222.12911	289.31676	211.56302
1483	-340.47604	195986	196061	196059	185.17209	235.82384	252.29448
1484	-340.31628	210501	210518	229285	241.14157	387.20251	156.38031
1485	-340.00249	77354	58280	58319	224.34569	377.17599	168.09326
1486	-339.97403	221998	221895	221910	259.94539	336.45580	174.71311
1487	-339.73879	238077	238222	238085	191.77735	192.40493	272.11040
1488	-339.73724	252293	252304	252400	24.56941	303.19337	280.94802
1489	-339.65092	94111	94095	94063	119.22040	269.02036	261.94229

TABLE 1: Portion of the feature star triangle data.

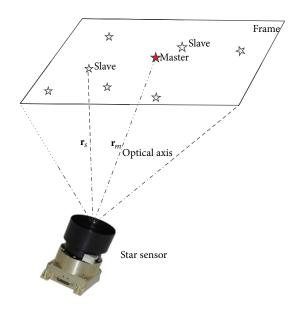


FIGURE 5: Star-sensor-capturing stars.

can facilitate the calculation and we can store D_{ms} in float-type data in the DSP, which can save memory as well. To this end, 10000 can satisfy the requirements of calculation. Incorporating this, when the angle between the master star and the slave star is 5°, the value of D_{ms} is 872.8. For the simulated star sensor with resolution of 1024 \times 1024, we choose 0.05° as a lower limit to avoid double star interference in star identification. Consider

$$D_{ms} = \|\mathbf{r}_{\mathbf{m}} - \mathbf{r}_{\mathbf{s}}\|_{2}, \qquad (2)$$

where ${\bf r_m}$ denotes the master star's unit vector to the stars within the FOV and ${\bf r_s}$ denotes the slave stars' unit vector to the stars within the FOV.

The rule for feature planar triangle selection is as follows. First, we pick the slave stars that satisfy $8.728 \le D_{ms} \le 872.8$ from the FOV and sort them by the distance between the master star and the slave stars. Second, the four nearest slave stars from the master star, denoted by s_1, s_2, s_3 , and s_4 , are kept. The first nearest slave star is denoted by s_1 , the second nearest slave star is denoted by s_2 , the same criteria apply for s_3, s_4 , and so on. If there are less than four slave stars,

they will all be kept. The four slave stars compose the planar triangles with the master star (denoted by m) as follows: Δms_1s_2 , Δms_1s_3 , Δms_1s_4 , Δms_2s_3 , Δms_2s_4 , and Δms_3s_4 .

(2) Computation of Projection Line for All Feature Triangles. The arbitrary triangles constructed above can be uniquely identified by a vector denoted by Δ_i , where $\Delta_i = (a_i, b_i, c_i)^T$. a_i, b_i , and c_i denote the three sides of the triangle, and when $1 \le i \le s_n$, i denotes the arbitrary triangles index and s_n denotes the total number of triangles. All the triangles constructed above can then compose a matrix: $\mathbf{M} = [\Delta_1, \Delta_2, \ldots, \Delta_n]$. The size of \mathbf{M} is $(3 \times n)$, where $n = s_n$. Then, the PCA algorithm is applied to \mathbf{M} , with the following procedure.

- (1) Calculate the column mean Δ of the matrix \mathbf{M} , where $\Delta = (1/n) \sum_{i=1}^{n} \Delta_i$ and $n = s_n$.
- (2) Construct the zero mean matrix $\mathbf{Q} = [\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \dots, \boldsymbol{\varphi}_n]$, where $\boldsymbol{\varphi}_i = \boldsymbol{\Delta}_i \boldsymbol{\Delta}$ and $1 \le i \le n$, $n = s_n$.
- (3) Calculate the eigenvalue of the covariance matrix C, where $C = QQ^T$, and then take the eigenvector corresponding to the largest eigenvalue \mathbf{u}_{max} as the main component of the data's direction. This is termed the "projection axis" in this paper.
- (3) Pattern Database Creation. For any arbitrary triangle, the projection corresponding to the projection axis can be found by using

$$p_{ri} = \mathbf{u}_{\max}^T \boldsymbol{\varphi}_i, \tag{3}$$

where p_{ri} denotes the *i*th triangle's projection, $1 \le i \le s_n$.

Then, each triangle can be assigned a unique index value in the projection axis. The projections of the triangles are sorted in order to generate a k-vector to locate a particular triangle in the pattern database. Table 1 presents a portion of this pattern database of the star triangle data matrix sorted in ascending order of minimum projection (p_r). Columns 3–5 indicate the ID of the stars in the celestial sphere. Columns 6–9 represent the three sides of the planar triangle formed by the stars.

A spatial distribution projection of the triangles on the projection axis is shown in Figure 6. The triangle indices corresponding to the projections are shown in Figure 7. In Figure 6, we can see that the projection values of some

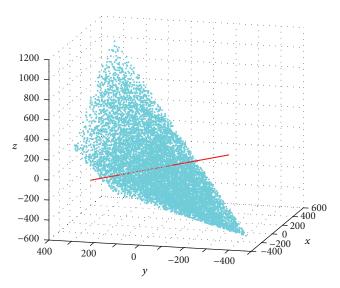


FIGURE 6: Spatial distribution projection of triangles on projection axis.

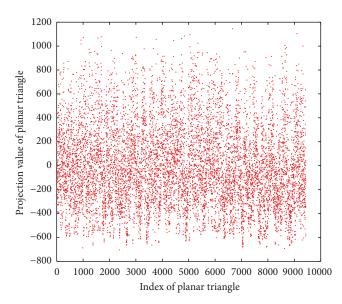


FIGURE 7: Projection corresponding to the index of the triangles.

triangular eigenvectors on the projection axis are very close. Actually, each triangle vector corresponds to a unique projection value. In Figure 7, we can see that some of the projections are very close. These close projections, corresponding to the triangles, will be taken as the candidate triangles in the star identification stage. Moreover, from Figures 6-7, it can be seen that the data points are too close. It means that the PCA algorithm cannot separate the planar star triangles well. However, the main function of PCA in this paper is used to reduce the dimension of the triangle vector. The three dimensions of the triangle vector are used to generate a projection value, which can be used for fast triangle matching combining with the *k*-vector method and the three-side restriction of the planar triangles. Then, the validation

method proposed validates the candidate matched triangles and makes the final identification of the stars.

- 2.2. Star Identification. Star identification consists of two steps. Step 1 focuses on obtaining the projection of the observed triangle and calculating an accurate projection deviation. Step 2 describes a quick solution to the triangle matching and star identification processes using the projection and projection deviation.
- (1) Projection of Observed Triangle. Using the guide database constructing process, we can obtain the mean vector $\mathbf{\Delta} = (a_v, b_v, c_v)^T$ of all the corresponding triangle vectors in the database as well as the direction vector $\mathbf{u}_{\text{max}} = (w_1, w_2, w_3)^T$ of the projection line. Given the arbitrary three star unit direction vectors within the FOV corresponding to the star sensor coordinates, denoted by $\mathbf{t_1}$, $\mathbf{t_2}$, and $\mathbf{t_3}$, we can calculate the three sides of a triangle that is a combination of these three stars as shown in

$$a = \|\mathbf{t}_1 - \mathbf{t}_2\|_2,$$

$$b = \|\mathbf{t}_2 - \mathbf{t}_3\|_2,$$

$$c = \|\mathbf{t}_1 - \mathbf{t}_3\|_2.$$
(4)

Then, we can assume that the vector corresponding to the triangle is denoted by $\widetilde{\Delta} = (a, b, c)^T$, and the projection of the triangle on the projection line can be found using

$$p_r = \mathbf{u}_{\text{max}}^T \left(\widetilde{\Delta} - \Delta \right)$$

$$= aw_1 + bw_2 + cw_3 - a_v w_1 - b_v w_2 - c_v w_3.$$
(5)

The equation given is for the star sensor frame. It can be also used, however, in the Earth-centered inertial frame. To obtain a boundary for the measurement error, the standard deviation of the calculated projection can be calculated.

Since the projection is a nonlinear function of \mathbf{t}_1 , \mathbf{t}_2 , and \mathbf{t}_3 , a linearization approach can be used to determine its variance. To compute the variance, the following 1×9 partial derivative matrix is evaluated by

$$\mathbf{T} = \begin{bmatrix} \mathbf{T}_1^T & \mathbf{T}_2^T & \mathbf{T}_3^T \end{bmatrix} = \begin{bmatrix} \frac{\partial p_r}{\partial \mathbf{t}_1} & \frac{\partial p_r}{\partial \mathbf{t}_2} & \frac{\partial p_r}{\partial \mathbf{t}_3} \end{bmatrix}, \tag{6}$$

where

$$\mathbf{T}_{1}^{T} = \frac{\partial p_{r}}{\partial a} \frac{\partial a}{\partial \mathbf{t}_{1}} + \frac{\partial p_{r}}{\partial c} \frac{\partial c}{\partial \mathbf{t}_{1}},$$

$$\mathbf{T}_{2}^{T} = \frac{\partial p_{r}}{\partial a} \frac{\partial a}{\partial \mathbf{t}_{2}} + \frac{\partial p_{r}}{\partial b} \frac{\partial b}{\partial \mathbf{t}_{2}},$$

$$\mathbf{T}_{3}^{T} = \frac{\partial p_{r}}{\partial b} \frac{\partial b}{\partial \mathbf{t}_{3}} + \frac{\partial p_{r}}{\partial c} \frac{\partial c}{\partial \mathbf{t}_{3}}.$$

$$(7)$$

The partials with respect to a, b, and c are given by

$$\frac{\partial p_r}{\partial a} = w_1,$$

$$\frac{\partial p_r}{\partial b} = w_2,$$

$$\frac{\partial p_r}{\partial c} = w_3.$$
(8)

The partials with respect to t_1 , t_2 , and t_3 are given by

$$\frac{\partial a}{\partial \mathbf{t}_{1}} = \frac{(\mathbf{t}_{1} - \mathbf{t}_{2})^{T}}{a},$$

$$\frac{\partial a}{\partial \mathbf{t}_{2}} = -\frac{\partial a}{\partial \mathbf{t}_{1}},$$

$$\frac{\partial b}{\partial \mathbf{t}_{2}} = \frac{(\mathbf{t}_{2} - \mathbf{t}_{3})^{T}}{b},$$

$$\frac{\partial b}{\partial \mathbf{t}_{3}} = -\frac{\partial b}{\partial \mathbf{t}_{2}},$$

$$\frac{\partial c}{\partial \mathbf{t}_{1}} = \frac{(\mathbf{t}_{1} - \mathbf{t}_{3})^{T}}{c},$$

$$\frac{\partial c}{\partial \mathbf{t}_{3}} = -\frac{\partial c}{\partial \mathbf{t}_{1}}.$$
(9)

The unit direction vector of any star t_i , corresponding to the star-sensor-body coordinate system, can be found using the standard coordinate transformation equation: $t_i = Ar_i$, where r_i is the unit direction vector of the star in the Earth-centered inertial coordinate system and A is the direction cosine matrix. When measurement errors (a typical star sensor follows a nearly zero mean Gaussian white-noise distribution [17]) are removed, the probability of errors is concentrated on the direction of Ar_i [18]. Then, the sphere containing that point can be approximated by a tangent plane [13], characterized by

$$\tilde{\mathbf{t}}_{i} = \mathbf{A}\mathbf{r}_{i} + \mathbf{v}_{i},$$

$$\mathbf{v}_{i}^{T} \mathbf{A}\mathbf{r}_{i} = 0,$$
(10)

where $\tilde{\mathbf{t}}_{i}$ denotes the *i*th measurement and the sensor error \mathbf{v}_{i} is approximately Gaussian, which satisfies

$$E\left\{\mathbf{v_{i}}\right\} = 0,$$

$$\mathbf{R_{i}} \equiv E\left\{\mathbf{v_{i}}\mathbf{v_{i}}^{T}\right\} = \sigma^{2}\left[\mathbf{I} - (\mathbf{A}\mathbf{r_{i}})(\mathbf{A}\mathbf{r_{i}})^{T}\right],$$
(11)

where σ^2 denotes the variance of the star sensor noise and E denotes the expected sensor error. Additional details of this model are given in [19]. Since the direction cosine matrix $\bf A$ is unknown, we can substitute $\tilde{\bf t}_i$ for $\bf Ar_i$ [18], and then (11) can be reconstructed as

$$\mathbf{R_i} \equiv E\left\{\mathbf{v_i}\mathbf{v}_i^T\right\} = \sigma^2 \left[\mathbf{I} - \widetilde{\mathbf{t}_i}\widetilde{\mathbf{t}_i}^T\right]. \tag{12}$$

According to (4) to (12), the variance of the projection can be calculated by

$$\sigma_p^2 = \mathbf{TRT}^T, \tag{13}$$

where

$$\mathbf{R} \equiv \begin{bmatrix} \mathbf{R}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_3 \end{bmatrix}, \tag{14}$$

where **0** denotes a 3×3 matrix of zeros and $\mathbf{R_1}$, $\mathbf{R_2}$, and $\mathbf{R_3}$ are given by (12). Note that the matrices **T** and **R** cannot be evaluated at their respective true values; however, the errors that are generated by replacing the true values with the measured ones can be negligible compared with typical star sensor noise levels. Since the standard deviation σ_p is derived analytically, the range of the true projection can be determined precisely.

- (2) Star Identification Procedure. In the star identification phase, we number the stars to be identified in an image. Then, planar triangles are constructed for the numbered stars. We take each numbered star as a master star and select stars that satisfy the feature triangle construction criteria from the remaining stars as slave stars. After constructing the planar triangles, we can calculate the three sides of the triangle by using (1) and can also calculate the projection p_r of the triangle using (3). Owing to the influence of noise, there must be a projection deviation. The standard deviation of the projection, σ_p , can be found using (13). Then, we can confirm that the boundaries of the true value of projection are in the range $[p_r - \xi, p_r + \xi]$, where $\xi = 3\sigma_p$. As this range is chosen to be the measurement projection ±3 times the standard deviation of the measurement noise, $3\sigma_p$, the true measurement is expected to be within this range 99.7% of the time. After the range of error is determined, we can use the k-vector method to quickly search and match in the pattern database. The *k*-vector method works as follows.
 - (1) Construct a linear equation z = mx + q, where q is the k-vector constant, m is the slope of the k-vector, z is the projection, and x is a function for the error. Each projection $p_r(k)$ corresponds to a triangle in Table 1 where the maximum projection p_{\max} is denoted by $p_r(s_n)$ and the minimum projection p_{\min} is denoted by $p_r(1)$. Then, the step of the length d can be found using

$$d = \frac{(p_{\text{max}} - p_{\text{min}})}{n} \tag{15}$$

and q and m can be given by

$$q = p_{\min} - m - d,$$

$$m = \frac{(p_{\max} - p_{\min} + 2d)}{n}.$$
(16)

TABLE 2: Portion of *k*-vector value data.

Limit index	k-vector value
1909	1574
1910	1579
1911	1581
1912	1583
1913	1584
1914	1585

(2) Compute the projection of the triangle p_r in the star image, and then, the true value p_{true} is assumed to satisfy the boundaries $p_{\text{true}} \in [p_r - \xi, p_r + \xi]$. We can then obtain the upper and lower limits of x using

$$x_{l} = \frac{\left(p_{r} - \xi - q\right)}{m},$$

$$x_{h} = \frac{\left(p_{r} + \xi - q\right)}{m},$$
(17)

where x_l denotes the lower limit of x, x_h denotes its upper limit, and $k_l \le x \le k_h$.

- (3) Substituting x_l and x_h into the equation z = mx + q gives z_l and z_h .
- (4) The range is delimited by a lower index k_l and a higher index k_h :

$$p_{r}(p) \leq z_{l} < p_{r}(p+1),$$

$$p_{r}(q) \leq z_{h} < p_{r}(q+1),$$
(18)

where p and q denote the index in Table 1. Consider

$$k_l = p + 1,$$

$$k_b = q + 1.$$
(19)

Obviously, all of the k-vector values should be computed offline. One lower or higher limit of x that is corresponding to a k-vector value can be computed offline. We can generate a k-vector search database. Table 2 shows a port of k-vector search data. Column 1 indicates the limit index that is obtained by (17). Column 2 represents the k-vector value corresponding to the limit index.

When the searching range is determined, we can compare the three sides of the observed triangle with the reference triangle(s) within the boundary. If the three side errors are within a small range, only one triangle will meet the criteria, and so the observed triangle will match the reference triangle, and the stars corresponding to the triangle will also match. If there is more than one triangle that meets the criteria, we store the candidate triangles and initiate the validation procedure.

2.3. Validation. With respect to all the candidate matched triangles, we store the IDs of the stars in the star image corresponding to the catalog as follows: $P_i - I_i$, where P_i

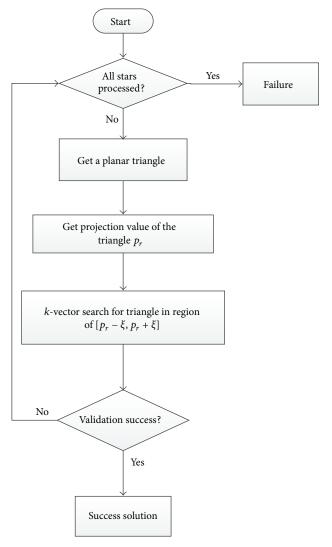


FIGURE 8: Flow chart of star identification and validation algorithm.

denotes the IDs of the star in the star image, I_i denotes IDs of the star in the catalog, and $1 \le i \le n$, where n denotes the total number of possible identified star IDs parts. Since there are redundant triangles existing in some cases, n is more than equal to the total number of stars in the star image. Then, we verify the identification result in the validation stage. The flow chart of the star identification and the validation is shown in Figure 8. From the feature planar triangles construction procedure, it can be seen that different planar triangles share stars, which indicates that a star can exist in different feature triangles within the star sensor's FOV. Based on this, a validation method was constructed. As shown in Figure 9, $P_i(k)$ and $I_i(k)$ denote the star IDs in the star image and in the catalog of the ith star in the kth feature triangle, respectively. Δ_k and Δ_t represent the kth and tth feature planar triangles, respectively. If a star in different triangles has the same IDs in the star image and catalog, our PTP algorithm will successfully identify the star and the triangles with the star; otherwise, it will be determined as a failure. The following short example, whose results are shown in Table 3,

Triangle		1			2			3			4	
Star number	6	11	3	8	7	10	12	11	10	5	7	4
IDs	230523	230575	230492	230644	230667	230596	212793	230575	230596	230726	230667	230692

TABLE 3: The validation results of the identified stars.

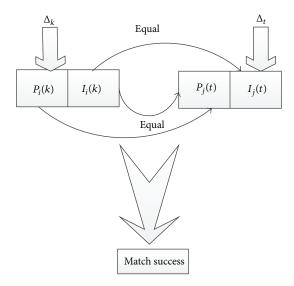


FIGURE 9: The diagram of star identification validation.

demonstrates how the validation method works. The input image was generated by star image simulation procedure from the star catalog, the star sensor's optical parameters, and its attitude. The resulting image contains 12 stars. Only 9 stars are extracted. The rest are too faint to be detected. Only identified stars participate in the validation stage. The first row of Table 3 shows the identified triangles. The identities of the stars in image corresponding to star catalog are shown in the second and the third rows of Table 3. It can be seen that the identified stars in different triangles share common identities in star image and catalog. The validation method determines that stars 3, 4, 5, 6, 7, 8, 10, 11, and 12 are correctly identified. Then, the attitude of the star sensor can be determined by the correctly identified stars' coordinates in image and their matching positions in the catalogue.

3. Simulation and Analysis

In this section, we evaluate the star identification algorithm presented in this paper and compare it with the *P*-vector and planar triangle algorithms. We show the results for the simulated images. We tested the algorithm using MATLAB in a Microsoft Windows 7 environment on an i5-2400 3.10 GHz PC. The simulations focused on discussing the number of slave stars in guide database construction and the robustness of the key parameters of the algorithm, the number of stars in the FOV, the noise on the star position, and the noise on the star magnitude. For star position noise, it mainly comes from two sources: one comes from the precision of the star centroid algorithms [20, 21] and the other comes from lens distortion

TABLE 4: Star sensor characteristics used in the star image simulation.

Parameter	Value
Pixel resolution	1024 × 1024
Pixel size	$12 \mu\mathrm{m} \times 12 \mu\mathrm{m}$
Image center	(512, 512)
FOV	$10^{\circ} \times 10^{\circ}$
Focal length	70.2262 mm
Sensitivity of star magnitude	6.0 Mv
Point spread function	2 pixels

that brings lateral distortion and radial distortion to image. In addition, the star sensor's sensitivity calibration error may bring in star magnitude noise. Most algorithms [6, 22–26] test their robustness by adding pixel and magnitude noise on the star image. In this paper, we still adopt this way to test the robustness of our algorithm.

3.1. Simulation Platform. For these experiments, a platform in a VC++ environment was designed to simulate the pictures taken by an onboard star sensor. The configuration of the star sensor is shown in Table 4. Each test was performed using 10,000 uniform sensor orientations. The orientations were generated using a Monte Carlo method. For each orientation, stars within a $10^{\circ} \times 10^{\circ}$ FOV were picked from the star catalog.

3.2. Discuss the Number of Slave Stars in Guide Database Construction. The number of the slave stars is very crucial for constructing the guide database. To obtain the proper number of slave stars for constructing our database, we set the limit number of the closet slave stars to 3, 4, 5, and 6 for testing, respectively. Then, we compare the memory size of these guide databases constructed by these limits. Also, 10,000 uniform sensor orientations are generated to test the star identification rate and average identification time with the different guide databases. The result is given with more than 5 stars in the FOV of star sensor, which is shown in Table 5. Column 2 shows the memory size of the guide databases constructed under a different number of slave stars. The guide databases store the feature star triangle data, which are constructed as Section 2.1 demonstrates. Obviously, the larger the memory size of the guide database is, the more time the star identification will take. In turn, when the number of closet slave stars is 3, it obtains the smallest size memory of the guide database and achieves the lowest star identification rate, because the database is less sufficient than the other three databases. From the result, it can be seen that a good balance between memory size and average identification time can be achieved when the number of closet slave stars is set to 4.

TABLE 5: The result with a different number of slave stars.

The number of slave stars	Memory size (MB)	Star identification rate (%)	Average identification time (ms)		
3	0.436	97.92%	2.52		
4	0.784	99.3%	3.11		
5	1.128	99.32%	3.86		
6	1.643	99.34%	4.94		

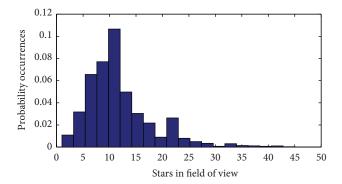


FIGURE 10: Probability occurrences of number of stars in the FOV.

3.3. Robustness against Number of Stars in FOV. The number of stars in the FOV has an impact on the identification rate. Figure 10 shows the probability occurrences of the number of stars in the FOV for 10,000 apparent positions. The minimum number of stars in the FOV was 2, and the maximum was 45. The average number of stars was 14.18. In the FOV, the probability of more than eight stars was 96.47%.

Generally, the more stars there are in the FOV, the higher the identification rate is. Figure 11 shows the changes in the identification rate with the number of stars in the FOV for the three algorithms. From Figure 10, we can see that, in the case of fewer stars in the FOV, the planar triangle algorithm achieved a higher identification rate than the PTP and P-vector algorithms. Our PTP algorithm achieved a higher identification rate than the P-vector algorithm. When the number of stars in the FOV was more than seven, the PTP and P-vector algorithms achieved a higher identification rate, about 100%. The identification rate of the planar triangle algorithm tended toward 100% with some fluctuation. Figure 11 also shows that neither our PTP nor the P-vector algorithm ensured a high identification rate with less than five stars in the FOV. The reason is that our PTP and the *P*-vector algorithms could not cover all the feature triangles. However, we can see from Figure 10 that, in most cases, there are more than five stars within the FOV, which means that our PTP algorithm can work well in most cases. Our PTP algorithm achieves a balance between memory storage and star identification time.

3.4. Robustness against Star Positional Noise. To simulate the star positional error, a Gauss noise with a zero mean and a standard deviation ranging from 0 to 2 pixels was used.

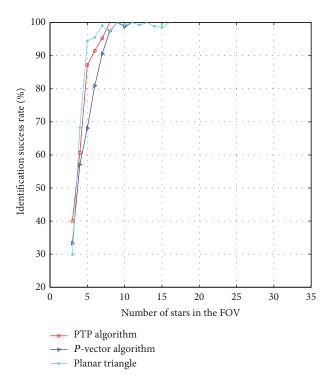


FIGURE 11: Identification rate versus number of stars in FOV.

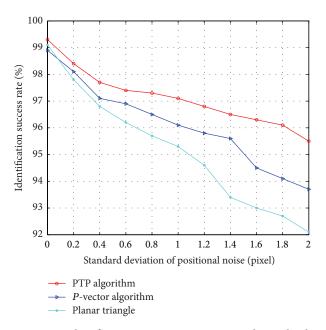


FIGURE 12: Identification rate versus star positional noise level.

Figure 12 shows the statistical results of the star identification process with more than five stars in the FOV for the three algorithms. All the algorithms were implemented under the same simulation conditions. It can be seen that when the standard deviation of the star positional error changed from 0 to 2 pixels, the identification rate of the planar triangle algorithm dropped from 99.1% to 92.1%, the identification rate of the *P*-vector algorithm dropped from 98.9% to 93.7%, and

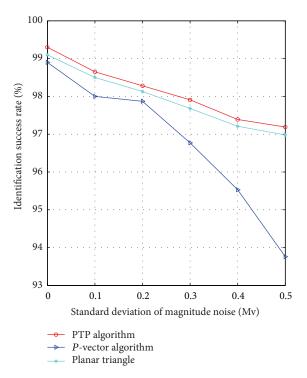


FIGURE 13: Identification rate versus star magnitude noise level.

the identification rate of our PTP algorithm dropped from 99.3% to 95.5%. Compared with the other two algorithms, our PTP algorithm achieved a higher identification rate at the noise level of 2 pixels. Because of the star position noise, the redundant match and mismatch of star triangles will occur during the star identification process, which could reduce the success rate of the algorithms. The redundant rate of match and mismatch of triangles increases as the star position noise grows. However, the sharing star validation method proposed here can reduce the probability of the redundant match and mismatch triangles. Therefore, our PTP algorithm could be more robust than the other two algorithms towards star position noise.

3.5. Robustness against Star Magnitude Noise. In addition, a magnitude deviation of 0.5 Mv was added to each star in order to simulate the influence of brightness inaccuracy. Simulated sky images with more than five stars were selected to test the identification rate for the three algorithms. The statistical results with magnitude noise are shown in Figure 13. As noise was added, the identification rate of the P-vector algorithm dropped from 98.9% to 93.7%. The magnitude noise impacted the P-vector algorithm's ability to generate the pattern triangles, which in turn reduced the identification rate. The other two algorithms do not use the magnitude information to generate the pattern triangles. Because of the magnitude noise, some stars are missing or added in the star image, which could reduce the other two algorithms' identification rate. It can be seen that the magnitude noise exerted less influence on the PTP and planar triangle algorithms than that on P-vector algorithm.

Table 6: Comparison of the three algorithms (positional noise std. dev = 1 pixel, and magnitude noise std. dev = 0.5 My).

Algorithm	Identification rate	Average identification time	Memory requirement (MB)	
PTP	96.79%	3.28 ms	0.784	
P-vector	94.91%	7.63 ms	0.537	
Planar triangle	93.25%	583.75 ms	178	

3.6. Comparison and Analysis. A comprehensive comparison of the three algorithms is presented in Table 6. It indicates that the proposed algorithm was more robust than the other two. The memory size for the guide database of our PTP algorithm was 0.784 MB. Under similar conditions, 0.537 MB was required by the P-vector algorithm and 178 MB was needed by the planar triangle algorithm. This was more than 200 times larger than the equivalent PTP database. Moreover, the average time needed for the computation of the proposed algorithm was less than that for the other two. This is because a k-vector search technique is used in the proposed algorithm. The most important result was that the identification rate of our PTP algorithm was higher than those of the other two algorithms.

4. Conclusion

An effective algorithm for star identification has been developed using the three sides of planar triangles that also incorporates principal component analysis (PCA) and a k-vector technique (PTP). The algorithm generates the star pattern in a projection that is computed using the three sides of the triangle projected on a projection line. The guide database is constructed similarly. For reliability, a projection is used for the initial match, and the adoption of the *k*-vector range search technique makes this algorithm very fast. Moreover, a sharing star validation method is adopted for the final match. The algorithm presented here can be used in the lost-in-space mode to determine the attitude for a star sensor that has no prior attitude information. Simulation results indicate that our PTP algorithm is more robust against noise than are the P-vector and planar triangle algorithms. In addition, it is seen from the simulation results that the identification rate of the proposed algorithm is comparatively high and the storage of the guide database and the identification speed are both acceptable. The algorithm should be transplanted to a star sensor immediately for supporting a wide variety of missions. However, the algorithm proposed here was not tested on real sky images. We will try to acquire the real images to test the robustness of the algorithm in the future. Another future direction is to track and locate the space object using star sensor technique.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The work was supported by the National Natural Science Foundation of China (no. 61372177).

References

- [1] H. Dang, A. Maloof, and M. Romalis, "Ultrahigh sensitivity magnetic field and magnetization measurements with an atomic magnetometer," *Applied Physics Letters*, vol. 97, no. 15, Article ID 151110, 2010.
- [2] A. Lambert, P. Furgale, T. D. Barfoot, and J. Enright, "Field testing of visual odometry aided by a sun sensor and inclinometer," *Journal of Field Robotics*, vol. 29, no. 3, pp. 426–444, 2012.
- [3] M. Wei, F. Xing, and Z. You, "An implementation method based on ERS imaging mode for sun sensor with 1 kHz update rate and 1" precision level," *Optics Express*, vol. 21, no. 26, pp. 32524–32533, 2013.
- [4] Y. Wang, E. Chang-Siu, M. Brown, M. Tomizuka, M. I. Almajed, and B. N. Alsuwaidan, "Three dimensional attitude estimation via the triad algorithm and a time-varying complementary filter," in *Proceedings of the 5th ASME Annual Dynamic Systems and Control Conference joint with the 11th JSME Motion and Vibration Conference*, pp. 157–165, American Society of Mechanical Engineers, 2012.
- [5] M. V. Arbabmir, S. M. Mohammadi, S. Salahshour, and F. Somayehee, "Improving night sky star image processing algorithm for star sensors," *Journal of the Optical Society of America A: Optics and Image Science, and Vision*, vol. 31, no. 4, pp. 794–801, 2014.
- [6] M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum, "Geometric voting algorithm for star trackers," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 441–456, 2008.
- [7] C. R. McBryde and E. G. Lightsey, "A star tracker design for CubeSats," in *Proceedings of the IEEE Aerospace Conference*, pp. 1–14, IEEE, Big Sky, Mont, USA, March 2012.
- [8] P. Zhang, Q. Zhao, J. Liu, and N. Liu, "A brightness-referenced star identification algorithm for APS star trackers," *Sensors*, vol. 14, no. 10, pp. 18498–18514, 2014.
- [9] B. B. Spratling and D. Mortari, "A survey on star identification algorithms," *Algorithms*, vol. 2, no. 1, pp. 93–107, 2009.
- [10] C. C. Liebe, "Star trackers for attitude determination," *IEEE Aerospace and Electronic Systems Magazine*, vol. 10, no. 6, pp. 10–16, 1995.
- [11] N. G. Roman and W. H. Warren Jr., Smithsonian Astrophysical Observatory (SAO) star catalog (Sao staff 1966, edition ADC 1989): Documentation for the machine-readable version, 1989.
- [12] D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junkins, "The pyramid star identification technique," *Navigation*, vol. 51, no. 3, pp. 171–184, 2004.
- [13] C. L. Cole and J. L. Crassidis, "Fast star-pattern recognition using planar triangles," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 1, pp. 64–71, 2006.
- [14] J. Yang, G.-J. Zhang, and J. Jiang, "Fast star identification algorithm using P vector," *Acta Aeronautica et Astronautica Sinica*, vol. 28, no. 4, pp. 897–900, 2007.
- [15] B. M. E. Quine, Autonomous star identification, Google Patents, 1999.
- [16] H. Abdi and L. J. Williams, "Principal component analysis," Wiley Interdisciplinary Reviews: Computational Statistics, vol. 2, no. 4, pp. 433–459, 2010.

- [17] J. L. Crassidis, F. Landis Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *Journal of Guidance*, *Control, and Dynamics*, vol. 30, no. 1, pp. 12–28, 2007.
- [18] A. Carmi and Y. Oshman, "Adaptive particle filtering for spacecraft attitude estimation from vector observations," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 232–241, 2009.
- [19] M. D. Shuster, "Filter quest or request," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 2, pp. 643–645, 2009.
- [20] Y.-K. Chang, B.-H. Lee, and S.-J. Kang, "High-accuracy image centroiding algorithm for CMOS-based digital sun sensors," in Proceedings of the 6th IEEE Conference on SENSORS (SENSORS '07), pp. 329–336, October 2007.
- [21] W. Xu, Q. Li, H.-J. Feng, Z.-H. Xu, and Y.-T. Chen, "A novel star image thresholding method for effective segmentation and centroid statistics," *Optik*, vol. 124, no. 20, pp. 4673–4677, 2013.
- [22] J. Gong, L. Wu, J. Gong, J. Ma, and J. Tian, "Flower algorithm for star pattern recognition in space surveillance with star trackers," *Optical Engineering*, vol. 48, no. 12, Article ID 124401, 2009.
- [23] J. Jiang, G. Zhang, X. Wei, and X. Li, "Rapid star tracking algorithm for star sensor," *IEEE Aerospace and Electronic Systems Magazine*, vol. 24, no. 9, pp. 23–33, 2009.
- [24] C. Padgett and K. Kreutz-Delgado, "A grid algorithm for autonomous star identification," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 202–213, 1997.
- [25] Q. Wei, X. Liang, and F. Jiancheng, "A new star identification algorithm based on improved hausdorff distance for star sensors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 49, no. 3, pp. 2101–2109, 2013.
- [26] G. Zhang, X. Wei, and J. 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.

















Submit your manuscripts at http://www.hindawi.com























