

A star identification algorithm based on simplest general subgraph

Hao Liu, Xinguo Wei^{*}, Jian Li, Gangyi Wang

School of Instrumentation and Optoelectronic Engineering, Beihang University, 100191, China

ARTICLE INFO

Keywords:

Star tracker
Star identification
Subgraph isomorphism
Analytical model

ABSTRACT

Subgraph isomorphism-based star identification algorithms require fewer stars than pattern-based algorithms and are suitable for practical application. Polygon algorithms and match group algorithms, as two typical subgraph isomorphism-based algorithms, both have disadvantages in efficiency and reliability. A novel star identification algorithm is presented in this study to solve this problem. We develop an analytical model to evaluate the validity of different subgraphs, which provides guidance to choose subgraphs. Based on the model, a series of effective and reliable subgraphs with different numbers of vertices, defined as the simplest general subgraphs, are chosen to achieve fast and direct star identification. The star matching strategy is divided into two basic steps. Based on the voting strategy, a star is initially identified by building match groups. It's further identified by building the simplest general subgraphs determined by the size of match groups. A verification approach of reprojection is adopted to improve the robustness of the algorithm. Compared with similar algorithms, the simulation test and night sky image test both show that the proposed algorithm is more robust to position noise, brightness noise, and false stars.

1. Introduction

As an important attitude measurement device, star trackers can generate three-axis attitudes by observing stars. As stars remain relatively still in the inertial frame, star trackers can generate attitude with high precision, the error is within a few arcseconds, better than other devices such as sun sensor and magnetometer. Meanwhile, unlike inertial measurement devices, star trackers have no cumulative error. These characters make star trackers widely used in spacecraft. Star identification is used to match given sensor stars detected in the field of view (FOV) to catalog stars of the database. As an important technique for star trackers, it has much research potential. With the development of the technique, star identification algorithms operate even in smartphones and commercial Micro-Drones [1].

Generally, the existing star identification algorithms are mainly divided into two categories: the algorithms based on subgraph isomorphism and the algorithms based on pattern recognition. The grid algorithms [2–5] and the radial and cyclic feature-based algorithms [6–10] are two typical algorithms of pattern recognition-based algorithms. Generally, the size of the database for pattern-based algorithms is smaller than that of subgraph isomorphism-based algorithms. Also, pattern-based algorithms have more potential for speed improvement. However, the pattern-based algorithms are more sensitive to the effect of

noise, especially for false stars. To ensure a good identification probability, the pattern-based algorithms usually require a large minimum number of stars. Unlike the pattern-based algorithms, the subgraph isomorphism-based algorithms, developed by introducing graph theory, require fewer stars and are simpler to implement. With stars as vertices and the angular distances of stars as edges, an all-sky star map is defined as a graph consisting of all the catalog stars. The subgraph is defined as a graph consisting of the observed stars in the FOV. Therefore, the principle of these algorithms is to search for the closest match of subgraph from the all-sky star map. Polygon algorithms and match group algorithms are two typical subgraph isomorphism-based algorithms. Triangle algorithm [11–14] and pyramid algorithm [15] are two classical algorithms of polygon algorithms. For triangle algorithms, the angular distances or interior angles of stars are used to generate a triangle feature. The algorithm usually just requires 3 true stars. However, it's restricted by large memory requirements for the feature database. Zhao et al. [13] and Wang et al. [14] reduced the dimension of the triangle feature by introducing the K-L transform and a hash map into the algorithm, respectively. As the triangle subgraph is not reliable enough, an extra star is added in the pyramid algorithm [15]. Although the pyramid algorithm is more robust to false stars, it becomes more time consuming as the number of false stars increases like the triangle algorithm. Arnas et al. [16] presented a fast and robust kernel generator, focus on

* Corresponding author. The authors are with the School of Instrument Science and Opto-Electronics Engineering, Beihang University, Beijing, 100191, China.
E-mail address: wxg@buaa.edu.cn (X. Wei).

generating sequences of triads (three stars) or quads (four stars) without repetition, providing an optimization method for triangle algorithm and pyramid algorithm to choose the stars for identification. For some other polygon algorithms, Kim et al. [17] introduced a labeling technique to improve algorithm speed and Toloei et al. [18] studied pentagon subgraph, quadrilateral pyramid subgraph, and triangular dipyradid subgraph. The polygon subgraph is reliable and effective for several stars, especially for 3–4 stars as the subgraph is simple to implement. However, it has a common problem as it may choose false stars for identification as the number of false stars increases, and the algorithm will be more time-consuming. Besides, the size of the database also limits the performance of the algorithms. Van et al. [19] first presented the match group algorithm. With one of the given stars as the kernel star and others as companion stars, match groups are generated by matching angular distances from companion stars to the kernel star. We define this kind of subgraphs as group subgraphs. A given star is identified by the size of the match groups with each candidate as the kernel star. The algorithm requires a large scratch space to form match groups. Several algorithms [20–22] were developed to solve that problem. Kolomenkin et al. [20] developed a geometric voting algorithm that first used the voting strategy to describe the size of match groups. The catalog star with the most votes is recognized as the match of the given star. However, this algorithm is sensitive to false stars. To solve this problem, Li et al. [21] presented an iteration algorithm that iteratively determined a catalog star until a unique match was found. The iteration algorithm is robust to position noise, brightness noise, and false stars. However, it has the risk of non-convergence when the number of true stars is not enough or applies in a large FOV. Compared with polygon algorithms, match group algorithms are more effective when the number of true stars is enough due to the voting strategy. Even when a few false stars are contained in the subgraph, the algorithms are still valid. The match group algorithms also have a low requirement for storage space as they use the database of angular distance. However, the group subgraphs are not reliable and effective enough with several stars. In addition to subgraph isomorphism-based algorithms and pattern-based algorithms, some other algorithms were presented in recent years, such as image processing-based algorithms [23,24], Hausdorff distance-based algorithm [25], and neural networks-based algorithm [26–28]. They are either inefficient or sensitive to the false stars.

Polygon algorithm and match group algorithm have disadvantages in efficiency and reliability with different numbers of stars. To solve this problem, we present a novel star identification algorithm based on the simplest general subgraph. An analytical model was developed to estimate the validity of different subgraphs, which provided guidance to choose subgraphs for subgraph isomorphism-based algorithms. A series of effective and reliable subgraphs including different numbers of vertices, defined as the simplest general subgraphs, were chosen to achieve fast and direct star identification. Such as the pyramid subgraph with 4 vertices and group subgraphs with more than 14 stars. Like other improved match group algorithms, the voting strategy is adopted in the process of building match groups to enhance efficiency. The remainder of this paper is organized as follows: In Section 2, the analytical model and the details of the proposed algorithm are described; in Section 3, the performance of the algorithm is demonstrated and compared with other similar algorithms; in Section 4, discussion and conclusion are drawn.

2. Algorithm description

In this section, an analytical model for the proposed algorithm is derived. After that, some implementation details of the algorithm are provided, such as database generation and isomorphic subgraph generation.

2.1. Analytical model

Some subgraph isomorphism-based algorithms [15,21,29]

developed analytical models to demonstrate the validity of the chosen subgraphs and guided algorithm design. Li [21] developed an analytical model to estimate the number of candidates for the kernel star of a group subgraph. For a group subgraph, vertices refer to the given stars, and one of the given stars is defined as a kernel star, and the rest are defined as companion stars. The edges of group subgraphs, named major edges, refer to the angular distances from the kernel star to each companion star. Besides, we define the angular distances of companion stars as minor edges. We extended Li's model and made it suitable for more general subgraphs with minor edges adding to group subgraphs. For a matched star pair, it must satisfy the following equation,

$$|\theta_c - \theta_s| \leq \sigma \quad (1)$$

where θ_c and θ_s refer to the angular distance of a catalog star pair and a sensor star pair, respectively, and σ denotes the permissible error. The analytical model also assumes that the catalog and observed stars are uniformly distributed across the celestial sphere.

With one catalog star as a pole, an area is defined as a spherical surface on the celestial sphere of angle that ranges from $\theta - \sigma$ to $\theta + \sigma$. This area is shown as the shadowed region in Fig. 1.

The area of shadow region is expressed as,

$$S_1 = 2\pi \int_{\theta-\sigma}^{\theta+\sigma} \sin \theta d\theta = 4\pi \sin \theta \sin \sigma \quad (2)$$

If at least one catalog star falls in this area, the pole catalog star matches the given sensor star pair. As the distribution of each catalog star is assumed to be independent from the other stars, the probability of pole star matches the given star pair is given by [21].

$$P_1 = 1 - \left(1 - \frac{S_1}{4\pi}\right)^{N-1} \quad (3)$$

For N catalog stars, the expected number of spurious candidates for the kernel star is approximated as follows

$$E = N \cdot P_1 \quad (4)$$

For the case of matching a kernel star of a group subgraph with m major edges, the expected number of spurious matches for kernel star is

$$E = N \prod_{i=1}^m \left[1 - (1 - \sin \theta_i \sin \sigma)^{N-1}\right] = N \prod_{i=1}^m P_1(\theta_i) \quad (5)$$

An approximate expression of Eq. (5) is given by

$$E = N \cdot \bar{P}_1^m \quad (6)$$

where \bar{P}_1 is obtained by substituting the angular distances of 10^5 star pairs in Eq. (3) and computing the average value of P_1 . The star pairs are sampled from the simulated star images with random orientations. For a triangle subgraph with three stars numbered 1–3, we define star 1 as the kernel star and the rest as companion stars. To match the kernel star of a triangle subgraph, the match of 3 star pairs (short for SP) including two

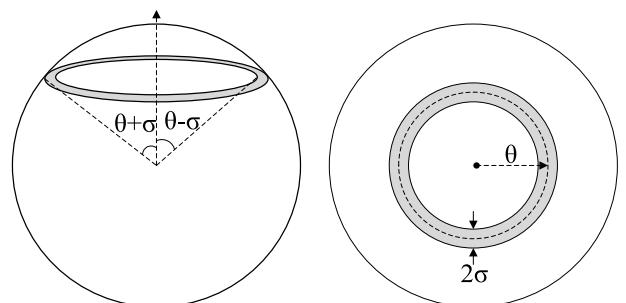


Fig. 1. Illustration of the spherical surface on the celestial sphere of angle that ranges from $\theta - \sigma$ to $\theta + \sigma$

major edges and a minor edge should be taken into consideration. With one catalog star as the pole, the probability of SP (1, 2) being matched can be obtained by Eq. (3). As for SP (1, 3) and SP (2, 3) with a common star 3, an area is defined as the overlapped region of two spherical surfaces with two corresponding catalog stars of star 1 and star 2 as poles like Fig. 1. The area is shown as two symmetrical shadowed regions in Fig. 2.

An approximate expression for the area of one of the shadowed regions is

$$S_2 \approx l_1 l_2 \sin(\theta_3) \quad (7)$$

where θ_3 is the interior angle of the triangle. As σ is usually much smaller than the angular distances, S_2 can be approximated by a plane parallelogram shown in Fig. 3.

The approximate expression for S_2 is given by

$$S_2 \approx l_1 \cdot 2\sigma = l_2 \cdot 2\sigma = \frac{4\sigma^2}{\sin(\theta_3)} = l_1 l_2 \sin(\theta_3) \quad (8)$$

If at least one catalog star besides two corresponding catalog stars of star 1 and star 2 falls in overlapped regions, this catalog star matches star 3. The probability of matching two star pairs with a common star is given by

$$P_2 = 1 - \left(1 - \frac{2S_2}{4\pi}\right)^{N-2} \quad (9)$$

Similarly, we can obtain \bar{P}_2 by substituting the angular distances of 10^5 triangles in Eq. (9) and computing the average value of P_2 . The triangles are sampled from the simulated star images with random orientations. Now, considering a subgraph as shown in Fig. 4.

It consists of $n+1$ stars ($n \geq 3$) numbered 1 to $n+1$. We define star 1 as the kernel star and others as companion stars. To match the kernel star of this subgraph, the match of $2n-1$ star pairs including n major edges and $n-1$ minor edges should be taken into consideration. Assuming the corresponding catalog stars of star 1 to n have been determined, the probability of matching n star pairs with the common star $n+1$ (SP(1, $n+1$), SP(2, $n+1$), ..., SP(n , $n+1$)) equals to the probability of at least a catalog star besides n determined catalog stars distributes in the overlapped region of n spherical surfaces with these determined catalog stars as poles like Fig. 1. This probability is defined as \bar{P}_n . With one catalog star as the match of star 1, if we match SP (1, 2), SP (1, 3), ..., SP (1, n) separately, then match the rest star pairs {SP (1, $n+1$), SP (2, $n+1$), ..., SP (n , $n+1$)}, the expected number of spurious candidates for the kernel star of the subgraph is given by

$$E = N \cdot \bar{P}_1^{-n-1} \cdot \bar{P}_n \quad (10)$$

Similarly, with one catalog star as the match of star 1, if we match SP (1, $n+1$) first, then match {SP (1, 2), SP (2, $n+1$)}, {SP (1, 3), SP (3, $n+1$)}, ..., {SP (1, n), SP (n , $n+1$)} separately, the expected number of

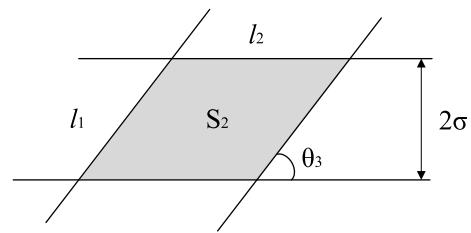


Fig. 3. Illustration of the plane parallelogram.

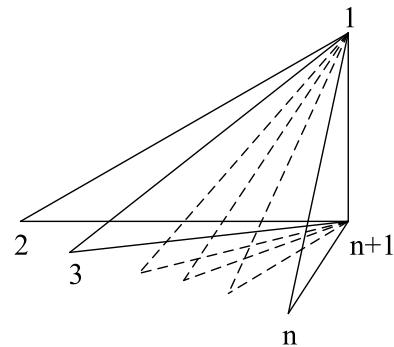


Fig. 4. Illustration of a subgraph with $n+1$ vertices, n major edges, and $n-1$ minor edges.

spurious candidates for the kernel star of the subgraph is given by

$$E = N \cdot \bar{P}_1 \cdot \bar{P}_2^{n-1} \quad (11)$$

Thus, \bar{P}_n , defined as the probability of matching n star pairs with a common star, is expressed as follow

$$\bar{P}_n = \frac{\bar{P}_2^{n-1}}{\bar{P}_1^{n-2}} \quad (12)$$

For any subgraph, we can estimate the number of candidates for the kernel star by dividing the subgraph into a combination of star pairs with different common stars. As shown in Fig. 5, a subgraph consists of 6 stars numbered 1 to 6, star 1 is defined as kernel star and the rest are defined as companion stars.

We divided the subgraph into the following combination of star pairs: SP (1, 2), {SP (1, 3), SP (2, 3)}, {SP (1, 4), SP (2, 4)}, {SP (1, 5), SP (2, 5), SP (3, 5)} and SP (1, 6). The expected number of spurious candidates for kernel star of the subgraph is given by

$$E = N \cdot \bar{P}_1 \cdot \bar{P}_2^{-2} \cdot \bar{P}_3 \cdot \bar{P}_4 = N \cdot \bar{P}_1 \cdot \bar{P}_2^4 \quad (13)$$

Using the model to analyze a star tracker, of which the key parameters are listed in Table 1.

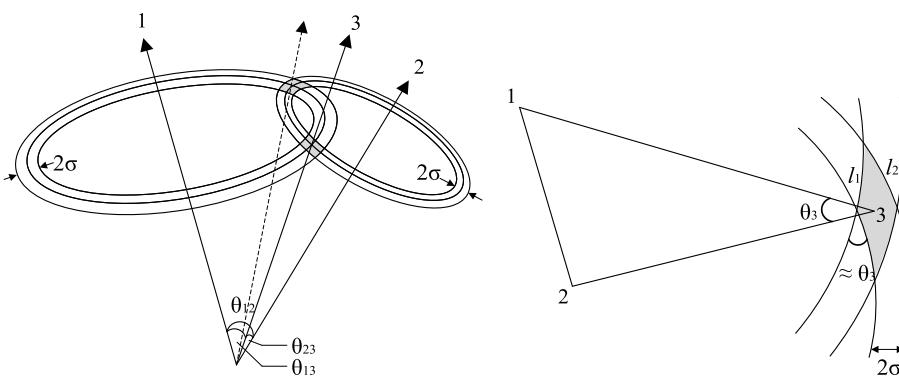


Fig. 2. Illustration of the overlapped region of two spherical surfaces.

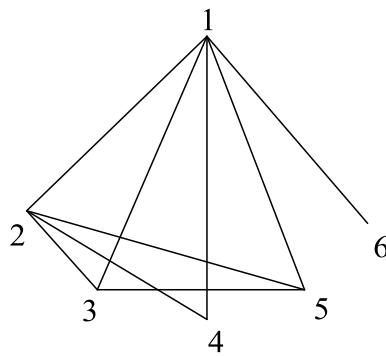


Fig. 5. Illustration of a general subgraph with 6 vertices, 5 major edges, and 4 minor edges.

Table 1
Parameters of the star tracker.

Parameter	Value
FOV	$\Phi 17^\circ$
Focal length	37.6846 mm
Resolution	2048×2048 pixels
Principal point	(1024, 1024)
Magnitude Limit	6.0Mv

When N is 3769 and σ is 0.04° , the estimated results of different subgraphs made up of group subgraph and some minor edges are listed in [Table 2](#).

It's a remarkable fact that the reliability of subgraphs improves dramatically after adding several minor edges. For algorithms based on subgraph isomorphism, the chosen subgraph must be effective enough to achieve fast and direct star identification. We define pyramid subgraphs as the simplest general subgraph with 4 stars. For a safe margin, we take 10^{-4} as the standard to choose the subgraph with more than 4 stars. We define the simplest general subgraph as the smallest subgraph for a given number of observed stars that satisfies the requirement of the expected number of spurious matches being less than 10^{-4} . It's made up

of a group subgraph with the same number of vertices and a smallest number of minor edges. As we can see, the number of minor edges decreases as the size of the group subgraph increases, such as 4 minor edges for a group subgraph with 5 stars and 2 minor edges for a group subgraph with 10 stars. Besides, the simplest general subgraphs with the same number of stars may have different structures, which demonstrates the general character of the chosen subgraphs. A series of simplest general subgraphs are listed in [Table 3](#). In fact, stars do not follow a uniform probability distribution in the celestial sphere, they are more concentrated around the galactic plane. Usually, some methods are adopted to select guide stars from the all-sky star catalog, which will make the distribution of guide stars more uniform. We selected 3769 guide stars from 5103 catalog stars of the SAO catalog. However, guide stars still do not follow a uniform probability distribution strictly. To verify the validity of the analytical model, we first randomly generated 3769 simulated stars with an approximate uniform distribution in the celestial sphere. Based on simulated stars and guide star catalog, we simulated 10^6 star images, respectively. For each simulated star image, we randomly choose a certain number of observed stars to form a specific simplest general subgraph with one of them as the kernel star, and we recorded the number of spurious matches for the kernel star in the database. The simulation1 column refers to the results of simulated stars, the simulation2 column refers to the results of catalog stars, and the model column refers to the results derived by the analytical model. As we can see, the results of simulation1 are consistent with the results of the analytical model, which proves the validity of the analytical model. Certainly, as guide stars do not follow a uniform probability distribution strictly, there is a deviation between the results of simulation 2 and the results of the model. Considering this fact, a larger safety margin could be considered depending on the actual situation. Considering we choose 10 observed stars for star identification, the number of spurious matches for kernel stars of these chosen subgraphs are about 10^{-3} to 10^{-4} according to the values provided in column simulation2, except for pyramid subgraphs. The reliabilities of the chosen subgraphs are acceptable for star identification.

2.2. Database generation

The database is composed of two parts, a star catalog and a database

Table 2
Number of spurious candidates of kernel star for different subgraphs.

Number of vertices	Number of minor edges					
	0	1	2	3	4	5
2	1074.2					
3	306.1	7.3				
4	87.2	2.1	5.0×10^{-2}	1.2×10^{-3}		
5	24.9	5.9×10^{-1}	1.4×10^{-2}	3.4×10^{-4}	8.1×10^{-6}	1.9×10^{-7}
6	7.1	1.7×10^{-1}	4.0×10^{-3}	9.6×10^{-5}	2.3×10^{-6}	5.5×10^{-8}
7	2.0	4.8×10^{-2}	1.1×10^{-3}	2.7×10^{-5}	6.5×10^{-7}	1.6×10^{-8}
8	5.8×10^{-1}	1.4×10^{-2}	3.3×10^{-4}	7.8×10^{-6}	1.9×10^{-7}	4.5×10^{-9}
9	1.6×10^{-1}	3.9×10^{-3}	9.3×10^{-5}	2.2×10^{-6}	5.3×10^{-8}	1.3×10^{-9}
10	4.7×10^{-2}	1.1×10^{-3}	2.7×10^{-5}	6.4×10^{-7}	1.5×10^{-8}	3.6×10^{-10}
11	1.3×10^{-2}	3.2×10^{-4}	7.6×10^{-6}	1.8×10^{-7}	4.3×10^{-9}	1.0×10^{-10}
12	3.8×10^{-3}	9.1×10^{-5}	2.2×10^{-6}	5.2×10^{-8}	1.2×10^{-9}	2.9×10^{-11}
13	1.1×10^{-3}	2.6×10^{-5}	6.2×10^{-7}	1.5×10^{-8}	3.5×10^{-10}	8.4×10^{-12}
14	3.1×10^{-4}	7.4×10^{-6}	1.8×10^{-7}	4.2×10^{-9}	1.0×10^{-10}	2.4×10^{-12}
15	8.8×10^{-5}	2.1×10^{-6}	5.0×10^{-8}	1.2×10^{-9}	2.8×10^{-11}	6.8×10^{-13}

Table 3
The structures and reliabilities of different simplest general subgraphs.

Number of vertices	Number of minor edges	Number of candidates		
		Model	Simulation 1	Simulation 2
4	3	1.2×10^{-3}	2.6×10^{-3}	1.1×10^{-2}
5	4	8.1×10^{-6}	2.5×10^{-5}	3.0×10^{-4}
6	3	9.6×10^{-5}	1.0×10^{-4}	2.2×10^{-3}
7	3	2.7×10^{-5}	1.9×10^{-5}	7.5×10^{-4}
8	3	7.8×10^{-6}	1.3×10^{-5}	4.0×10^{-4}
9	2	9.3×10^{-5}	1.1×10^{-4}	4.6×10^{-3}
10	2	2.7×10^{-5}	1.7×10^{-5}	2.3×10^{-3}
11	2	7.6×10^{-6}	7.0×10^{-6}	1.2×10^{-3}
12	1	9.1×10^{-5}	1.8×10^{-4}	1.1×10^{-2}
13	1	2.6×10^{-5}	5.7×10^{-5}	5.2×10^{-3}
14	1	7.4×10^{-6}	2.4×10^{-5}	2.5×10^{-3}
15	0	8.8×10^{-5}	5.6×10^{-4}	3.7×10^{-2}

of angular distances. The star catalog records the information of each catalog star, including orientation vector, brightness information, and ID number. With the limiting magnitude of star tracker setting to 6Mv, a $17 \times 17^\circ$ circular FOV is adopted to ensure a sufficient number of stars. After deleting double stars, we generate 10^5 random sensor orientations, and the brightest 10 stars of each star image were marked. Finally, 3769 marked stars from the SAO star catalog are defined as catalog stars. The distribution of catalog stars is shown in Fig. 6.

At least 5 catalog stars are imaged in FOV, and the frequency of fewer than 10 stars is less than 1%. All the neighboring star pairs within the size of FOV were saved in the database in ascending order. We generate a catalog index with an interval of 0.04° which coincides with the permissible error. The storage space of the whole database is less than 2 megabytes.

2.3. Simplest general subgraph algorithm (SGS)

The principle of the algorithm is to identify a given star by building isomorphic subgraphs with the database. A two-step strategy is adopted in the algorithm. After two rounds of voting, the candidates of the given star with sufficient votes will remain, and for some of the candidates with the most votes, the match groups and the simplest general subgraphs with them as the kernel star will be constructed subsequently. Finally, the catalog stars of the isomorphic subgraph will be verified in the verification process. The pseudo-code for the algorithm is shown in Fig. 7 and a list of each function call used in the algorithm is provided in Table 4.

2.3.1. Voting

The voting strategy is used to record the number of matched star pairs from kernel star to each companion star. We select α brightest observed stars for identification. As too many stars will lead to a large amount of calculation, α is limited to 10. Each given star is defined as the kernel star in turn. For one of the given star s_i , defined as the kernel star, all of the $n-1$ angular distances from it to each companion star will be matched successively with the permissible error σ . We assign N counters for s_i , and each counter corresponds to a catalog star. We also assign an N long Boolean array that corresponds to N catalog stars for each given star pair, and each element is set when the corresponding catalog star is a candidate for the given star pair. In the first round of voting, for each catalog star pair (c_i , c_j) that matches the given star pair (s_i , s_j), the Boolean elements of c_i and c_j are set to 1 as they both could be a candidate of s_i . After scanning each of $n-1$ given star pairs, we will add the Boolean array to N counters of s_i and then clear the Boolean array. Finally, the catalog stars with no fewer than T votes are initially considered as the candidates of s_i . As a threshold, T directly affects the number of candidates. Based on the analytical model, we set T to 4. In the second round of voting, an extra limited condition is supplemented. Each star pair (c_i , c_j) matches a given star pair (s_i , s_j), only if c_i is a

candidate of s_i and c_j is a candidate of s_j . For each matching star pair, the counter of c_i is increased by one. In the pseudo-code presented in Fig. 7, l_{ij} is the matched segment of θ_{ij} in distance database, $cat_star_present$ is an N long Boolean array, and $cat_star_present [l_{ij} [k, 0:1]] \leftarrow 1$ means that all elements of array $cat_star_present$ at positions given in the subarray l_{ij} [row k , columns 0 to 1] are set to 1. E.g., if $k = 7$ and $l_{ij} [7, 0:1] = [253, 714]$, positions 253 and 714 in $cat_star_present$ are set to 1, and $l_{ij} [k, 0:1]$ is a shorthand notation for l_{ij} [row k , columns 0 to 1].

2.3.2. Building isomorphic subgraph

After two rounds of voting, for each given star s_i , we choose 5 candidates with the most votes, and each given star will be further identified by building an isomorphic subgraph with each candidate as the kernel star. For each candidate c_i , the votes refer to the size of the match group with it as the kernel star. To determine the companion stars, we scan all of the matched segments of the distance database that correspond to each major edge. As mentioned before, s_i is defined as the kernel of the subgraph and s_j is a companion star of the subgraph. Each star pair (c_i , c_j) matches a given star pair (s_i , s_j), only if c_i is a candidate of s_i and c_j is a candidate of s_j , then c_j is regarded as a companion star of the match group with c_i as the kernel star. If the size of the match group, defined as the number of catalog stars corresponding to different given observed stars, is no less than 4, we will continue building an isomorphic subgraph by adding the matched minor edges to it, until the isomorphic subgraph turns to the simplest general subgraph chosen by the size of the match group, such as 2 minor edges for a match group with 10 stars. Assigning a counter to record the number of matched minor edges as s_num in Fig. 6. To improve the efficiency of this process, we will give priority to test the star pair composed of companion stars with more votes. In the pseudo-code presented in Fig. 6, for a group member of the match group at position k , $match_group [k, 0]$ and $match_group [k, 1]$ refers to its ID number and corresponding observed star, respectively. And the group member at position 0 is the kernel star, other group members starting from position 1 are companion stars. Besides, $edge_present$ is a 45 long Boolean array, each element corresponding to one of 45 minor edges formed by two of 9 companion stars. When a minor edge is matched, the corresponding element of $edge_present$ is set to 1.

2.3.3. Verification

A reprojection approach is adopted to verify the catalog stars of the isomorphic subgraph. In the last process, a certain number of matched minor edges made up of companion stars were added to the isomorphic subgraph. Among these companion stars, the one with the most votes and the candidate of the kernel star c_i are used to compute the attitude of the star tracker. Based on the TRIAD algorithm, we can obtain the attitude matrix by two vectors [30]. Thus, we can infer the catalog stars of star image and locate their positions, projecting inferred catalog stars to star image. For each observed star, if only one catalog star distributes in the region within a certain distance to it, the catalog star will be recognized as the match of the observed star. And the distance is set to 3 pixels according to the specification of the star tracker in this algorithm. If the number of matches is no less than 4, the algorithm will return the identification results of the given stars. Otherwise, we will clear the votes of candidate c_i . If all of the candidates with the most votes have been checked and none of them pass the validation, the whole process will repeat num_iter times (as the high-level noise is added to the star images in the simulation experiments, we set num_iter to 2). The reprojection approach is an effective way to verify false matches, which is proven by the experimental results, as the algorithm has a low mis-identification probability. Also, reprojection could provide the extra matches of other observed stars beside the chosen stars in the entire FOV, which is beneficial for attitude determination.

3. Experimental results

A set of simulated star images and a set of night sky real images were

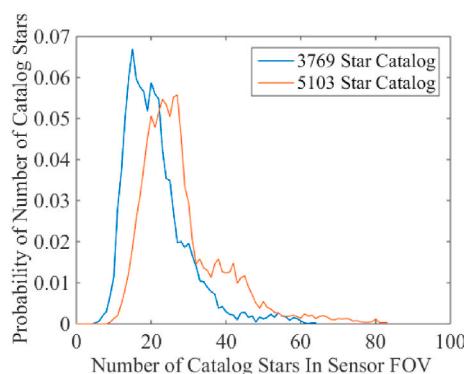


Fig. 6. Catalog star distribution over the celestial sphere.

```

procedure SGS( $S, D$ )
begin
     $flag \leftarrow 0$ 
    for  $i \leftarrow 0$  to  $\alpha-2$  do // first round of voting
        for  $j \leftarrow i+1$  to  $\alpha-1$  do
             $\theta_{ij} \leftarrow distance(s_i, s_j)$ 
             $[l_{ij}, len\_l_{ij}] \leftarrow find\_star\_pair(\theta_{ij})$ 
             $cat\_star\_present[0:N-1] \leftarrow 0$ 
            for  $k \leftarrow 0$  to  $len\_l_{ij}-1$  do
                 $cat\_star\_present[l_{ij}[k, 0:1]] \leftarrow 1$ 
             $counter[i, 0:N-1] \leftarrow counter[i, 0:N-1]+cat\_star\_present[0:N-1]$ 
             $counter[j, 0:N-1] \leftarrow counter[j, 0:N-1]+cat\_star\_present[0:N-1]$ 
     $n \leftarrow 0$ 
    while  $n \leq num\_iter$  do
        for  $i \leftarrow \alpha-1$  to  $0$  do // second round of voting
             $counter2[0:N-1] \leftarrow 0$ 
            for  $j \leftarrow 0$  to  $\alpha-1$  do
                 $cat\_star\_present[0:N-1] \leftarrow 0$ 
                for  $k \leftarrow 0$  to  $len\_l_{ij}-1$  do
                    if  $counter[i, l_{ij}[k, 0]] \geq T$  and  $counter[j, l_{ij}[k, 1]] \geq 4$  then
                         $cat\_star\_present[l_{ij}[k, 0]] \leftarrow 1$ 
                    elseif  $counter[i, l_{ij}[k, 1]] \geq T$  and  $counter[j, l_{ij}[k, 0]] \geq 4$  then
                         $cat\_star\_present[l_{ij}[k, 1]] \leftarrow 1$ 
                     $counter2[0:N-1] \leftarrow counter2[0:N-1]+cat\_star\_present[0:N-1]$ 
                     $counter[i, 0:N-1] \leftarrow counter2[0:N-1]$ 
            for  $i \leftarrow 0$  to  $\alpha-1$  do // build simplest general subgraph
                for  $j \leftarrow 0$  to  $4$  do
                     $Can_j \leftarrow select\_a\_candidate(counter)$ 
                     $[match\_gr, size\_gr, len\_gr] \leftarrow build\_match\_group(l, Can_j, counter)$ 
                    if  $size\_gr \geq 4$  then
                         $edge\_present[0:44] \leftarrow 0$ 
                         $T\_edge \leftarrow determine\_threshold(size\_gr)$ 
                        for  $k \leftarrow 1$  to  $len\_group-2$  do
                            for  $m \leftarrow k+1$  to  $len\_group-1$  do
                                 $SP\_dist \leftarrow distance(match\_gr[k, 0], match\_gr[m, 0])$ 
                                if  $|SP\_dist - \theta_{match\_gr[k, 1]match\_gr[m, 1]}| \leq \sigma$  then
                                     $s\_num \leftarrow seq\_num(match\_gr[k, 1], match\_gr[m, 1])$ 
                                     $edge\_present[s\_num] \leftarrow 1$ 
                                     $num\_minor\_edge \leftarrow sum(edge\_present[0:44])$ 
                                    if  $num\_minor\_edge = T\_edge$  then
                                         $[match\_star, len\_match\_star] \leftarrow verify(match\_gr, S, D)$ 
                                        if  $len\_match\_star \geq 4$  then
                                             $flag \leftarrow 1$ 
                                            return  $match\_star$ 
                                            goto end_procedure
                                     $counter[i, Can_j] \leftarrow 0$ 
                             $n \leftarrow n+1$ 
                        if  $flag=0$  then
                            return NULL
            end_procedure:
    end

```

Fig. 7. The pseudo-code of the simplest general subgraph (SGS) algorithm.

used to evaluate the performance of several similar algorithms, including the proposed algorithm, match group algorithm [19], geometric voting algorithm [20], and iteration algorithm [21]. We test the algorithms with the same database of angular distance, the same star images, and the same platform. These algorithms are implemented in MATLAB R2014b in the Microsoft Windows 10 environment on a PC

with a 2.3 GHz dual-core CPU.

3.1. Simulation

Using stars of SAO catalog with magnitudes range down to 9.0Mv, a simulation platform was developed to simulate star images. Three kinds

Table 4
The functions of the SGS algorithm.

Function	Description
<i>Distance</i>	Computes the angular distance of the star pair and returns θ
<i>find_star_pair</i>	Extracts the matched segment in distance database, returns l and the length len_l
<i>select_a_candidate</i>	Selects a candidate with the most votes and returns <i>Can</i>
<i>build_match_group</i>	Determines the match group with <i>Can</i> as the kernel star, returns <i>match_group</i> and the size <i>size_group</i>
<i>determine_threshold</i>	Determines the number of matched minor edges according to the size of the match group and returns <i>T_edge</i>
<i>seq_num</i>	Computes the sequence number of the matched minor edges and return <i>s_num</i>
<i>Sum</i>	Sums the array of <i>edge_present</i> and returns <i>num_minor_edge</i>
<i>Verify</i>	Computes the attitude of the star tracker by the TRIAD algorithm and derives the catalog stars of the FOV, then projects them to the star image. Finds the overlapping couples of observed star and catalog star, returns <i>match_star</i> and length <i>len_match_star</i>

of noise are introduced, including brightness noise, position noise, and false stars. Among them, brightness noise could cause some catalog stars to disappear and some faint stars to appear in the FOV. The position noise could cause a deviation of centroid localization of the observed stars. In this platform, both brightness noise and position noise obey Gaussian distribution with a mean value of 0. To simulate non-stars such as particle beams, false stars are introduced with random brightness magnitudes ranging down to 6.0Mv and random positions. The parameters of the platform are consistent with the parameters of the analytical model, which is listed in Table 1, the permissible tolerance σ is set to 0.04° . For each noise at a certain level, the simulated star images of 10^4

random sensor orientations were performed. A successful identification needs to meet all of the following criteria:

- (1) No less than four true stars are identified correctly.
- (2) No false stars are identified.
- (3) No true stars are identified incorrectly.

3.1.1. Performance towards position noise

With the standard deviation increased from 0 to 3 pixels ($0\text{--}0.025^\circ$, 1 pixel for 0.0083°), we add position noise to both the vertical and horizontal centroid coordinates of each star. Meanwhile, the standard deviation of brightness noise is maintained at 0.4Mv. In such a condition, about 4.6 stars with a magnitude range down to 6.0Mv disappear and 5.7 faint stars appear in the FOV. Fig. 8 Shows the results of four algorithms.

As shown in Fig. 8(a), the identification probability of our algorithm drops from 99.86% to 94.12% when the standard deviation increases from 0 to 3 pixels. The performance of our algorithm for identification probability is slightly better than the iteration algorithm and significantly better than the other two algorithms, especially for the position noise at higher levels. At a 3-pixel deviation, the average error of angular distance for each star pair is about 0.03° , the identification probability of our algorithm is 1.4% higher than the iteration algorithm, 9.3% higher than the match group algorithm, and 34.4% higher than the geometric voting algorithm. As shown in Fig. 8(b), the mis-identification probabilities of our algorithm and geometric voting algorithm are generally lower than the other two algorithms. As shown in Fig. 8(c), compared with other algorithms, our algorithm has a superior

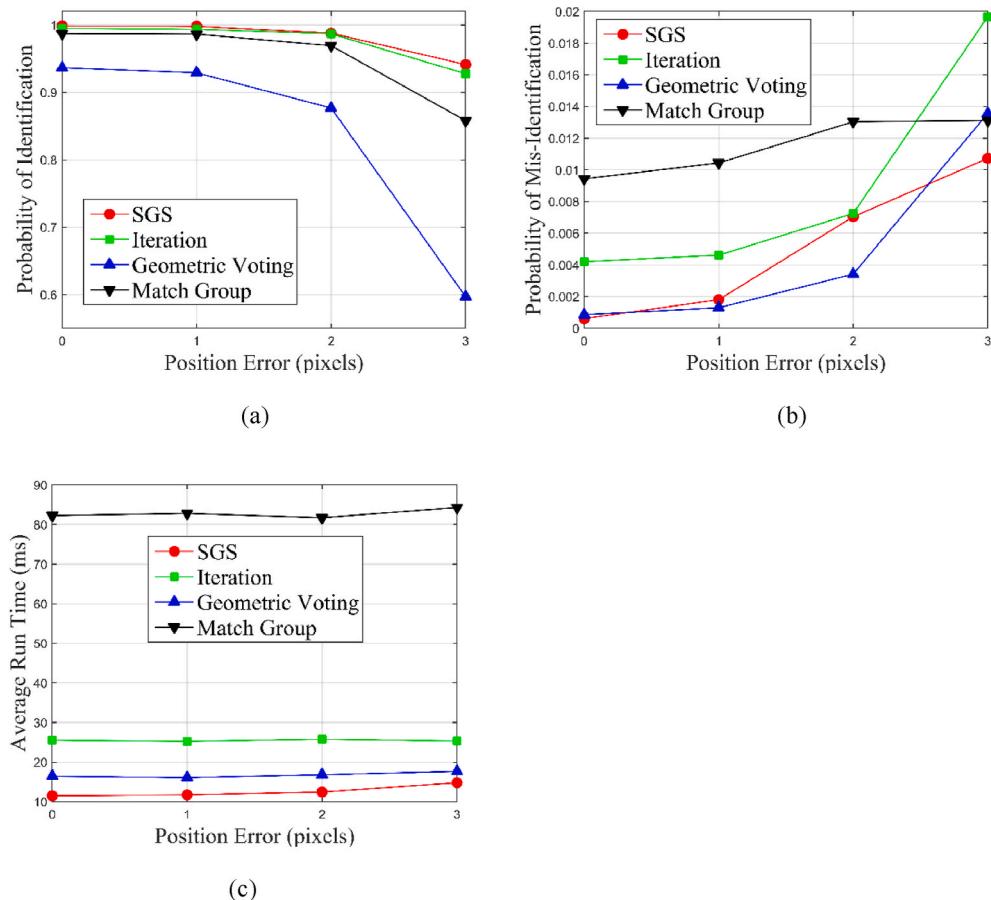


Fig. 8. Results versus position accuracy. (a) Identification probability versus position accuracy. (b) Mis-identification probability versus position accuracy. (c) Run time for star identification versus position accuracy.

performance. At a 3-pixel deviation, the average run time for the iteration algorithm, geometric voting algorithm, and match group algorithm is 25.3 ms, 17.7 ms, and 84.3 ms, respectively. Our algorithm only costs 14.8 ms. Compared with the performance in recognition rate and speed of the other three algorithms, our algorithm is more robust for position noise.

3.1.2. Performance towards brightness noise

We add brightness noise to each star with standard deviations increasing from 0 to 1Mv. Meanwhile, the standard deviation of position noise is maintained at 1 pixel (0.0083°), the average error of angular distance for each star pair is about 0.01° . With the deviation increasing to 1Mv, about 8.0 stars with the magnitude range down to 6.0Mv disappear and 29.3 faint stars appear in the FOV. Fig. 9 Shows the results of four algorithms, respectively.

As shown in Fig. 9(a), the identification probability of the presented algorithm drops from 99.76% to 92.20% when the standard deviation increases from 0 to 1Mv. The performance of our algorithm for identification probability is slightly better than the iteration algorithm and significantly better than the other two algorithms, especially for the brightness noise at higher levels. For the case of 1Mv, the identification probability of our algorithm is 1.4% higher than the iteration algorithm, 10.6% higher than the match group algorithm, and 46.4% higher than the geometric voting algorithm. As shown in Fig. 9(b), the mis-identification probabilities of our algorithm and geometric voting algorithm are generally lower than the other two algorithms. At a 1Mv deviation, the mis-identification probability of the iteration algorithm is a lot higher than other algorithms. As the deviation increases in Fig. 9(c), the real-time performance of our algorithm is consistently better than other algorithms. For the case of 1Mv, the average run time for the

iteration algorithm, geometric algorithm, and match group algorithm is 24.6 ms, 18.0 ms, and 74.4 ms, respectively. Our algorithm only cost 16.5 ms. Compared with the performance in recognition rate and speed of the other three algorithms, our algorithm is more robust for brightness noise.

3.1.3. Performance towards false stars

We replace a certain number of the observed stars to be identified with the same number of false stars. Meanwhile, the standard deviation of position noise is maintained at 1 pixel (0.0083°) and the standard deviation of brightness noise is maintained at 0.4Mv. In such a condition, about 4.6 stars with the magnitude range down to 6.0Mv disappear and 5.7 faint stars appear in the FOV and the average error of angular distance for each star pair is about 0.01° . Fig. 10 Shows the results of four algorithms.

As shown in Fig. 10 (a), the identification probability of our algorithm drops from 99.74% to 93.68% when the number of false stars increases from 0 to 4. The performance of our algorithm for identification probability is slightly better than the iteration algorithm and significantly better than the other two algorithms, especially for the case of 4 false stars. In that case, the identification probability of our algorithm is 1.3% higher than the iteration algorithm. And the identification probabilities of the other two algorithms are lower than 50%. As shown in Fig. 10 (b), the mis-identification probability of our algorithm is generally lower than other algorithms. For the case of 4 false stars, the mis-identification probabilities of the iteration algorithm and geometric voting algorithm are a lot higher than the other two algorithms. As shown in Fig. 10 (c), our algorithm is faster than other algorithms when the number of false stars is fewer than 4. Although the geometric algorithm is faster than our algorithm for the case of 4 false stars, it's

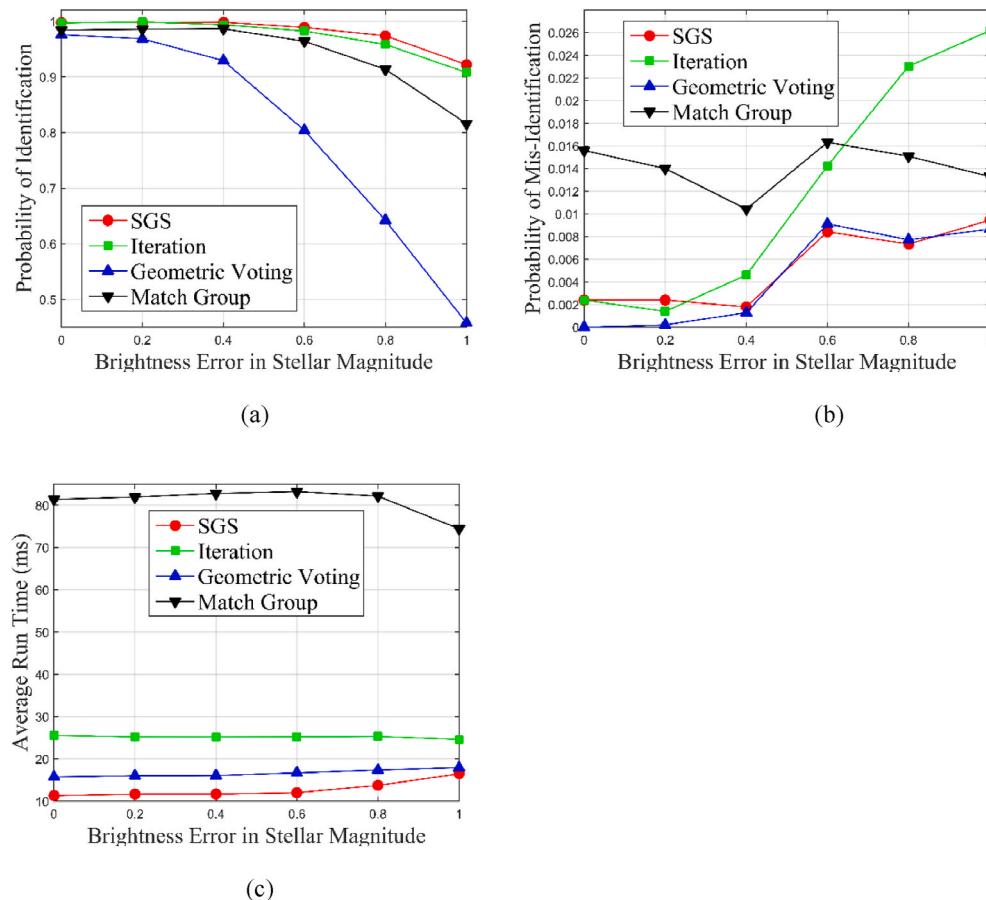


Fig. 9. Results versus brightness accuracy. (a) Identification probability versus brightness accuracy. (b) Mis-identification probability versus brightness accuracy. (c) Run time for star identification versus brightness accuracy.

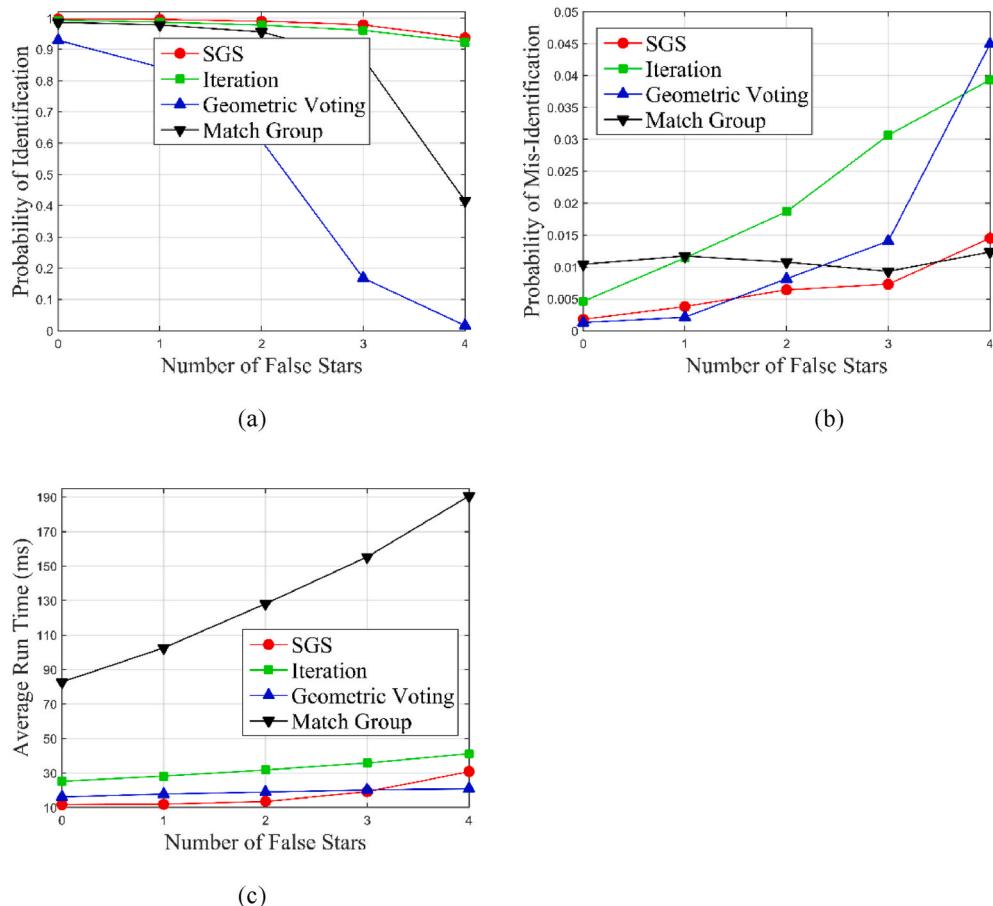


Fig. 10. Results versus the number of false stars. (a) Identification probability versus the number of false stars. (b) Mis-identification probability versus the number of false stars. (c) Run time for star identification versus the number of false stars.

ineffective for star identification as the identification probability is lower than 2%. For the case of 4 false stars, the average run time for our algorithm, iterative algorithm, and match group algorithm is 30.8 ms, 41.2 ms, and 190.1 ms, respectively. Compared with the performance in recognition rate and speed of other algorithms, our algorithm is more robust for false stars.

3.2. Experiments with real star tracker images

In addition to the simulated images, we also tested our algorithms on a set of night sky real images and on-orbit images captured by two CMOS APS star trackers, the parameters of the star trackers are listed in Table 5.

3.2.1. On the ground

The ground test was performed at Xinglong Astronomical Observatory of the Chinese Academy of Sciences as shown in Fig. 11 (e). The star tracker was used to capture star images only. We tested 2279 real star images of random sensor orientations at different times. One original star image and its identification results are shown in Fig. 11 (a)-(d).

Table 5
Parameters of the star trackers.

Parameter	On-ground	On-orbit
FOV	$20^\circ \times 15^\circ$	$20^\circ \times 20^\circ$
Focal length	13.5263 mm	43.2580 mm
Resolution	1280×960 pixels	1024×1024 pixels
Principal point	(600.2, 457.1)	(515.25, 513.32)
Magnitude limit	6.0Mv	6.0Mv

Fig. 11 (c) presents the star centroid results indicated by circles and the simplest general subgraph constructed with 10 stars. Catalog star 53,204 is a candidate of observed star 1 with the most votes, the match group with it as the kernel star is listed in Table 6.

The category column shows the sequent number of observed stars that each catalog star belongs to. The size of the match group is 10, as it has 9 companion stars that belong to 9 different given stars. Based on Table 3, it requires 2 minor edges for the group subgraph (match group) turning to the simplest general subgraph with 10 vertices. As shown in Fig. 11 (c), two minor edges SP (35,938, 52,871) and SP (35,938, 52,516) labeled with dotted lines are added to the group subgraph. And the attitude of the star tracker is derived by the TRIAD algorithm with two vectors of catalog star 53,204 and catalog star 35,938. After reprojection, all of the 10 brightest sensor stars are identified successfully, and the identification results labeled by catalog star ID numbers are shown in Fig. 11 (d). The detailed identification results are listed in Table 7.

3.2.2. Using on-orbit images

A total of 1864 on-orbit star images were obtained from the ZY-3 satellite, including two sets of consecutive frames of two periods. Besides, each image includes a sufficient number of observed stars with more than 10 true stars and few false stars. An original star image and its identification results are shown in Fig. 12 (a)-(d). Fig. 12 (c) presents the star centroid results indicated by circles and the simplest general subgraph constructed with ten stars. The kernel star of the isomorphic subgraph is 120,238, as the size of the match group is 10, 2 minor edges SP (120,601, 120,339) and SP (120,601, 120,434) are added to the subgraph. All of the 10 brightest sensor stars are identified successfully,

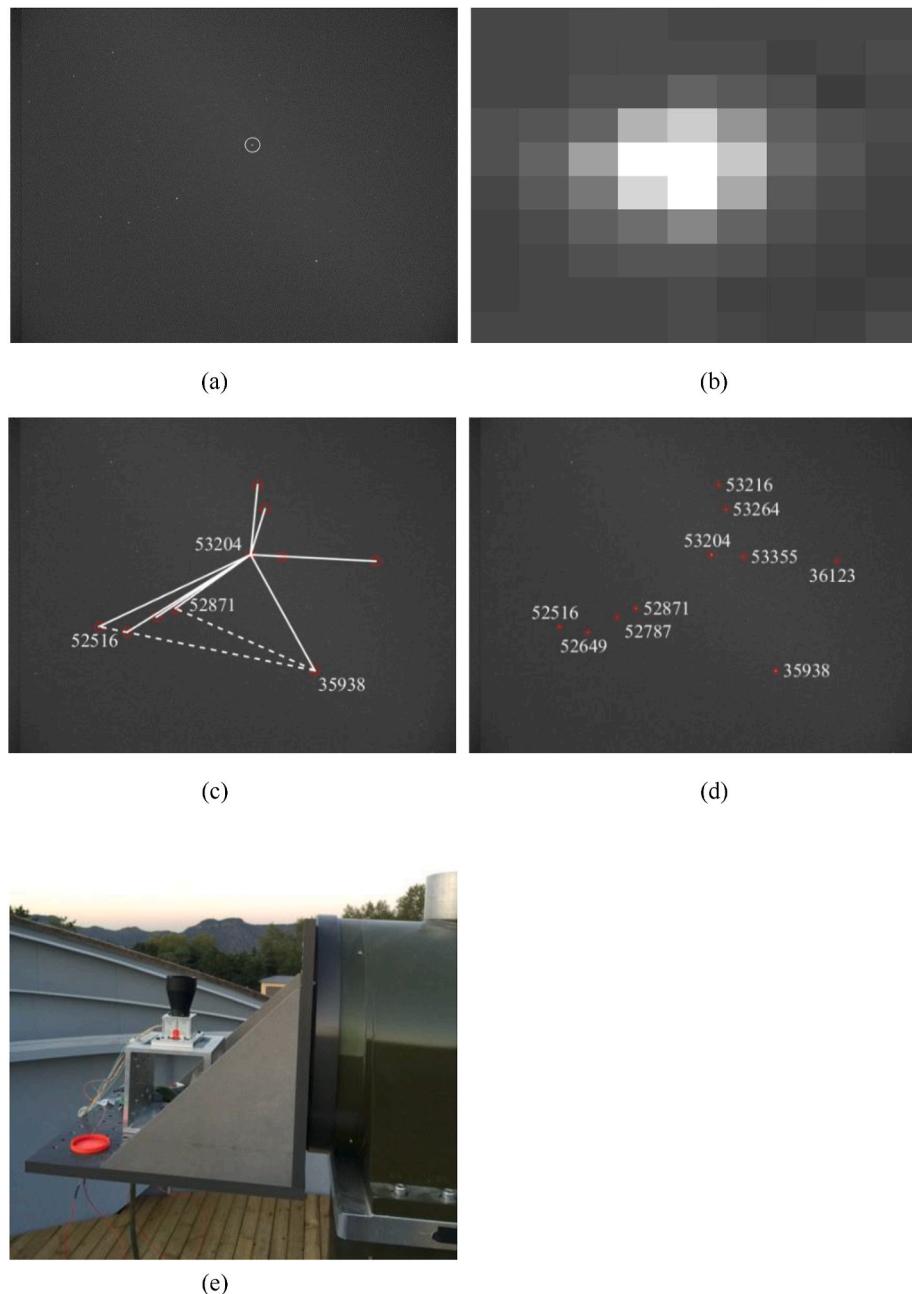


Fig. 11. Star identification on a real star image captured on the ground. (a) Original star image. (b) An enlarged bright star is circled in (a). (c) An isomorphic simplest general subgraph of (a). (d) Identification results with the proposed algorithm. (e) Star tracker used for capturing star images.

Table 6
The match group with star 53,204 as the kernel star.

ID number	Category	votes
53,204	1	9
35,938	2	9
52,871	3	9
52,516	4	9
52,649	5	9
53,264	6	9
53,216	7	9
53,355	8	9
52,787	9	9
36,123	10	9

and the identification results labeled by catalog star ID numbers are shown in Fig. 12 (d). The detailed identification results are listed in Table 8.

3.2.3. Results

The results of on-ground images and on-orbit images for four algorithms are listed in Table 9. The proposed algorithm and the iteration algorithm show better performance in identification, and the geometric voting algorithm shows better performance in speed, followed by the proposed algorithm.

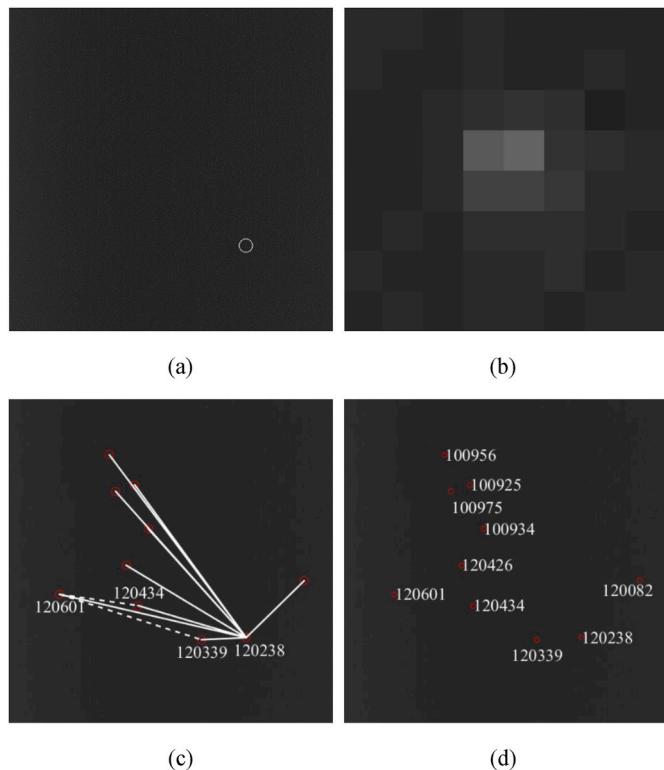
4. Conclusion

A novel star identification algorithm based on the simplest subgraph isomorphism is presented in this study. The major innovation is the

Table 7

Identification results of the on-ground star image.

Sensor star	Centroids		Total grayscale	Catalog star			
	x	y		ID	Right ascension (rad)	Declination (rad)	Apparent magnitude
1	692.7	393.2	1645	53,204	6.1853	0.8180	4.0
2	876.2	725.2	1420	35,938	6.2763	0.8969	4.8
3	477	546.9	1026	52,871	6.0988	0.8555	5.0
4	258.8	599.1	750	52,516	6.0058	0.8680	5.1
5	339.2	615.1	716	52,649	6.0396	0.8736	4.9
6	733.6	262.3	571	53,264	6.1977	0.7738	4.3
7	712.8	193.0	520	53,216	6.1878	0.7552	4.3
8	784.4	398.6	377	53,355	6.2222	0.8102	5.1
9	423.4	571.7	372	52,787	6.0761	0.8623	4.6
10	1051	411.9	272	36,123	0.0450	0.8041	5.1

**Fig. 12.** Star identification on a real on-orbit star image (a) Original star image. (b) An enlarged bright star is circled in (a). (c) An isomorphic simplest general subgraph of (a). (d) Identification results with the proposed algorithm.

development of an analytical model which is used to estimate the number of spurious matches for kernel star of different subgraphs. The analytical model provides guidance to choose effective subgraphs for

subgraph isomorphism-based star identification algorithms. Unlike match group algorithm, geometric voting algorithm, and iteration algorithm using group subgraph (match group), a series of chosen subgraphs named simplest general subgraph with several minor edges adding to the group subgraphs are chosen to achieve fast and direct star identification. For example, the classical pyramid subgraph is one of the simplest general subgraphs. As it can choose different effective subgraphs during the process of identification, the algorithm is effective and reliable for different numbers of true stars, like several stars for the polygon algorithms and a sufficient number of stars for the match group algorithm. Based on the analytical model, the number of spurious candidates for the kernel star of the simplest general subgraph is much fewer than the group subgraph with the same number of vertices, which is the key to the high efficiency of our algorithm. Compared with the iteration algorithm, the proposed algorithm avoids a large number of iterations as it chooses a more effective subgraph, the experimental results indicate that our algorithm is quite faster than the iteration algorithm with similar performances of identification probability. Similarly, compared with the geometric voting algorithm, the algorithm avoids a large number of spurious matches as it chooses a more effective subgraph, the experimental results indicate that the identification probability of our algorithm is quite higher than the geometric voting algorithm with similar real-time performances.

Table 9

Results of experiments using real sky images.

Algorithm	Identification rate (%)		Mis-identification rate (%)		Run time (ms)	
	On-ground	On-orbit	On-ground/On-orbit	On-ground	On-orbit	
SGS	100	100	0	8.6	17.8	
Iteration	100	100	0	18.2	24.7	
Geometric voting	100	91.4	0	7.9	19.8	
Match group	100	99.9	0	41.6	83.2	

Table 8

Identification results of the on-orbit star image.

Sensor star	Centroids		Total grayscale	Catalog star			
	x	y		ID	Right ascension (rad)	Declination (rad)	Apparent magnitude
1	750.4	754.2	441	120,238	3.6724	0.0270	4.3
2	160.2	617.4	351	120,601	3.8469	0.1424	5.0
3	609.2	761.1	321	120,339	3.7187	0.0421	4.9
4	408.2	652.8	311	120,434	3.7707	0.1016	5.1
5	441.0	409.9	294	100,934	3.7300	0.1763	5.4
6	370.9	525.3	248	120,426	3.7672	0.1474	5.1
7	318.1	176.1	239	100,956	3.7414	0.2664	6.0
8	338.0	291.7	183	100,975	3.7493	0.2270	5.3
9	398.3	271.0	178	100,925	3.7267	0.2262	5.5
10	933.8	574.4	157	120,082	3.5913	0.0617	5.6

The experimental results indicate that the algorithm is quite robust to position noise, brightness noise, and false stars. Besides, like other match group algorithms, the algorithm adopts the database of angular distance which requires a relatively small memory space. That means the algorithm has the potential for embedded systems. At the moment, the algorithm was only implemented on a computer. The future work will focus on incorporating the algorithm into a real star tracker for validation on orbit.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is funded by the National Key Research and Development Program of China (Grant No. 2019YFA0706002).

References

- [1] R. Marbel, B. Ben-Moshe, R. Yozevitch, Star-tracker algorithm for smartphones and commercial micro-drones, *Sensors* 20 (4) (2020).
- [2] C. Padgett, K. Kreutz-Delgado, A grid algorithm for autonomous star identification, *IEEE Trans. Aero. Electron. Syst.* 33 (1) (1997) 202–213.
- [3] H. Lee, H. Bang, Star pattern identification technique by modified grid algorithm, *IEEE Trans. Aero. Electron. Syst.* 43 (3) (2007) 1112–1116.
- [4] M. Na, D. Zheng, P. Jia, Modified grid algorithm for noisy all sky autonomous star identification, *IEEE Trans. Aero. Electron. Syst.* 45 (2) (2009) 516–522.
- [5] H. Zhu, B. Liang, T. Zhang, A robust and fast star identification algorithm based on an ordered set of points pattern, *Acta Astronaut.* 148 (2018) 327–336.
- [6] G. Zhang, X. Wei, J. Jiang, Full-sky autonomous star identification based on radial and cyclic features of star pattern, *Image Vis. Comput.* 26 (7) (2008) 891–897.
- [7] J. Jiang, F. Ji, J. Yan, L. Sun, X. Wei, Redundant-coded radial and neighbor star pattern identification algorithm, *IEEE Trans. Aero. Electron. Syst.* 51 (4) (2015) 2811–2822.
- [8] Y. Zhao, X. Wei, G. Wang, J. Li, Real-time star identification using synthetic radial pattern, *Acta Astronaut.* 131 (2017) 1–9.
- [9] M.D. Samirhai, S. Chen, K.S. Low, A hamming distance and spearman correlation based star identification algorithm, *IEEE Trans. Aero. Electron. Syst.* 55 (1) (2019) 17–30.
- [10] X. Wei, D. Wen, Z. Song, J. Xi, W. Zhang, G. Liu, Z. Li, A star identification algorithm based on radial and dynamic cyclic features of star pattern, *Adv. Space Res.* 63 (7) (2019) 2245–2259.
- [11] C.C. Liebe, Pattern recognition of star constellations for spacecraft applications, *IEEE Aero. Electron. Syst. Mag.* 8 (1) (1993) 31–39.
- [12] G. Zhang, *Star Identification: Methods, Techniques and Algorithms*, Springer, 2019, pp. 73–105.
- [13] Y. Zhao, X. Wei, J. Li, G. Wang, Star identification algorithm based on K-L transformation and star walk formation, *IEEE Sensor. J.* 16 (13) (2016) 5202–5210.
- [14] G. Wang, J. Li, X. Wei, Star identification based on hash map, *IEEE Sensor. J.* 18 (4) (2018) 1591–1599.
- [15] D. Mortari, M.A. Samaan, C. Bruccoleri, J.L. Junkins, The pyramid star identification technique, *Navigation* 51 (3) (2004) 171–183.
- [16] D. Arnas, M.A.A. Fialho, D. Mortari, Fast and robust kernel generators for star trackers, *Acta Astronaut.* 148 (2018) 327–336.
- [17] S. Kim, M. Cho, New star identification algorithm using labelling technique, *Acta Astronaut.* 134 (2017) 291–302.
- [18] A. Toloei, M. Zahednamazi, R. Ghasemi, F. Mohammadi, A comparative analysis of star identification algorithms, *Astrophys. Space Sci.* 365 (4) (2020).
- [19] R.W. Bezoijen Van, Techniques for optimizing an autonomous str tracker, U.S. Patent 5 (1997) 745–869.
- [20] M. Kolomenkin, S. Pollak, I. Shimshoni, M. Lindenbaum, Geometric voting algorithm for star trackers, *IEEE Trans. Aero. Electron. Syst.* 44 (2) (2008) 441–456.
- [21] J. Li, X. Wei, G. Zhang, Iterative algorithm for autonomous star identification, *IEEE Trans. Aero. Electron. Syst.* 51 (1) (2015) 536–547.
- [22] Q. Fan, X. Zhong, J. Sun, A voting-based star identification algorithm utilizing local and global distribution, *Acta Astronaut.* 144 (2018) 126–135.
- [23] T. Delabie, T. Durt, J. Vandersteen, Highly robust lost-in-space algorithm based on the shortest distance transform, *J. Guid. Contr. Dynam.* 36 (2) (2013) 476–484.
- [24] J. Roshanian, S. Yazdani, M. Ebrahimi, Star identification based on Euclidean distance transform voronoi tessellation and k-nearest neighbor classification, *IEEE Trans. Aero. Electron. Syst.* 52 (6) (2016) 2940–2949.
- [25] W. Quan, L. Xu, J. Fang, A new star identification algorithm based on improved Hausdorff distance for star sensors, *IEEE Trans. Aero. Electron. Syst.* 49 (3) (2013) 2101–2109.
- [26] L. Xu, J. Jiang, L. Liu, RPNet: a representation learning-based star identification algorithm, *IEEE Access* 7 (2019) 92193–92202.
- [27] D. Rijlaarsdam, H. Yous, J. Byrne, D. Oddenino, G. Furano, D. Moloney, Efficient star identification using a neural network, *Sensors* 20 (13) (2020).
- [28] J. Jiang, L. Liu, G. Zhang, Star identification based on spider-web image and hierarchical CNN, *IEEE Trans. Aero. Electron. Syst.* 56 (4) (2020) 3055–3062.
- [29] M. Kumar, D. Mortari, J.L. Junkins, An analytical approach to star identification reliability, *Acta Astronaut.* 66 (2010) 508–515.
- [30] M.D. Shuster, S.D. Oh, Three-axis attitude determination from vector observations, *J. Guid. Contr. Dynam.* 4 (1) (1981) 70–77.