IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 July 2023, accepted 3 August 2023, date of publication 14 August 2023, date of current version 30 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3304908

RESEARCH ARTICLE

Star Identification Algorithm Based on Multi-Dimensional Features and Multi-Layered Joint Screening for Star Sensors

HAODONG YAN[1,2,3], XURUI CHEN[1,2], GUOPENG DING[1,2], SHUAI ZHI[1,2], YONGHE ZHANG[1,2], AND ZHENCAI ZHU[1,2]

[1]Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai 201210, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China

Corresponding author: Zhencai Zhu (zczhu@hotmail.com)

This work was supported by the National Natural Science Foundation of China under Grant 42001408.

ABSTRACT The algorithm for star identification is a crucial technology for determining the orientation of spacecraft using star sensors. Traditional star identification algorithms achieve matching by seeking a unique or a few optimal solutions. However, in high-noise environments, some solutions may be lost, which could result in matching failure. A new lost-in-space architecture algorithm aimed at rapid identification under high star position noise conditions by directly using star positions for final matching is proposed in this paper. The main idea of this algorithm is to construct sufficiently redundant navigation triangles, fully utilizing the physical relationships of the features and forming a screening method from high to low dimensions and from loose to strict. During identification, a multi-layer joint screening matching method is adopted to screen triangles as a whole, narrowing the range of matches quickly while retaining error tolerance. In a series of simulation experiments, this algorithm achieved identification rates of 99.51%, 99.06%, and 98.42% for 2.0 pixel star position noise, 1.0 Mv star magnitude noise, and 5 false stars, respectively. In terms of practical application, all 1000 star images taken by the star sensor in orbit have been successfully identified, and it only takes 28ms to identify each image. In addition, star images taken by consumer-grade cameras from the ground also show that the algorithm has strong robustness to star position noise, magnitude error and false star interference in more severe environments. This method provides partial algorithmic reference for non-specialized design of star sensors for low-cost, large-scale satellites in the future.

INDEX TERMS Multi-dimensional features, multi-layered joint screening, star identification, star sensor, star tracker.

I. INTRODUCTION

As one of the most accurate sensors for determining satellite attitude [1], the star sensor can calculate the attitude without the need for prior information, which has been widely used in spacecraft. Star identification obtains the spacecraft spatial attitude information by matching stars within the current field of view(FOV) of the star sensor with reference stars in the navigation star catalog [2].

The operation of the star sensor can be divided into two modes [3]:lost-in-space (LIS) mode and tracking mode. When the star sensor is first activated or loses attitude data, it operates in LIS mode, which means the autonomous star identification across the full-sky must be performed without any prior information to establish the initial attitude. Once the initial attitude is obtained, the star sensor switches to tracking mode, which is less challenging to implement than the LIS mode. Therefore, researching a star identification algorithm with fast and highly robust performance in the LIS mode has always been an important research direction.

The associate editor coordinating the review of this manuscript and approving it for publication was Bin Xu.

Based on the methods of feature extraction, the star identification algorithms can be divided into three categories: pattern recognition algorithms, artificial intelligence algorithms, and subgraph isomorphism algorithms. The first category is pattern recognition algorithms, which typically construct a unique star pattern for each star based on the geometric distribution of stars around it. Matching is accomplished by searching the feature database for the navigation star that is most similar with the observed star pattern. They are mainly divided into grid algorithms [4], [5], radial and cyclic algorithms [6], Log-Polar transformation(LPT) algorithms [7] and so on. Pattern recognition algorithms have some advantages in false stars tolerance and storage capacity, but are generally more complex and depend on pre-processing, and sometimes the robustness and running time cannot be guaranteed.

The second category is the artificial intelligence method. Artificial intelligence algorithms mainly use neural networks [8], [9] to identify the imaging stars captured by star sensors. For example, methods based on convolutional neural networks [10], [11], [12], multi-layer SOM neural networks [13], and representative learning algorithms [14]have been designed to solve the star identification problem. These algorithms store features implicitly within the structure of the neural network, obviating the need for a feature database to be stored. However, their high complexity, extensive training time, and demand for large-scale parallel architectures limit their use.

The third category is subgraph isomorphism algorithm, which takes the angular distance between stars as edges and stars as vertices. This algorithm considers the observed star map as a subgraph of the entire sky map by constructing basic elements such as triangles and quadrilaterals. Once a unique matching area that satisfies the matching conditions is found in the full-sky star map, the identification is successful. For example, the polygonal angular distance identification algorithm [15]matches by comparing the angular distance between two observed stars with the angular distance in the feature database. This method is simple and easy to implement, but generally requires a prior estimate of the star sensor axis direction, and matching time will become too long and algorithm complexity will increase rapidly when there are many navigation stars. The triangle algorithm and a series of improved algorithms are currently the most used and mature star identification algorithms, which use the angular distances between three stars in a triangle as the matching feature. However, these methods rely on relatively accurate stellar magnitude information, and there are also many cases of incorrect matches. Zhang et al. [16]proposed an improved triangle identification algorithm that uses angle distance matching. They directly store the star pair's angle distance, and the identification speed was improved significantly. Mortari et al. [17]introduced the fourth star and used the k-vector method to create a pyramid algorithm, which greatly enhanced the robustness against false stars and other interferences. Zhao et al. [18]acquired better identification accuracy by using the Karhunen-Loève (K-L) transform to form star rows, which display the geometric distribution of each star in the neighborhood. Sun et al. [19]proposed a multi-viewpoint double triangle star identification algorithm based on the triangle algorithm. This algorithm constructs the double triangle feature of the star point using angle and distance information, exhibiting strong anti-interference ability against false stars and missing stars, but its robustness against noise in the star position is not well evaluated. Du et al. [20]proposed a star identification algorithm based on the radial triangle mapping matrix pattern to improve the robustness by adopting a geometric voting scheme and considered the radial triangle as the basic voting element. The subgraph isomorphism algorithm also includes matching group algorithms [21], which regards the navigation star with the most occurrences as the matching star through the selection of the main star and the companion star to form a star pair for matching. However, this method requires a large feature database, and its robustness performance is average.

Constellation satellites are becoming a trend in satellite systems due to their ability to provide global communication services in a short amount of time by enabling large-scale production and launch. Unlike traditional satellite systems, some constellation satellites typically use low-cost industrial cameras as their star sensors. Because these cameras lack the precise ground calibration of traditional star sensors, they require strong resistance to interference during operation, especially when it comes to robustness against noise errors in star positions. While subgraph isomorphism algorithms have strong robustness against positional and brightness noise, most algorithms currently focus on seeking unique or optimal solutions. However, this approach may lead to a loss of tolerance for more noise, resulting in matching failures.

To enhance robustness for future development, this article proposes two improvements. Firstly, the algorithm adopts the triangle as a matching unit, and generate sufficient redundant features during feature database generation, whereby any triangle matched by the algorithm results in a successful match. Secondly, a multi-dimensional matching framework is constructed, which use a multi-layered screening approach from high-dimensional features such as the shape factor, area, and polar moment of triangles to low-dimensional features such as the three side lengths. Matching and validation are performed on the pixel positions of the lowest dimensional features to avoid the destruction of possible solutions by high-dimensional features, and the robustness against noise in star point positions can be further enhanced. To shorten the matching time, the physical relationships of features are fully utilized, and a multi-feature joint matching method is adopted to perform selection of triangles as a whole unit. Simulation experiments and real star map matching results show that the proposed algorithm has significant advantages over mainstream subgraph isomorphism and pattern recognition

algorithms in terms of robustness against noise in star point positions, as well as the time required.

This article is organized as follows. Section II provides a detailed description of the algorithm, including the generation of triangle features, database construction, and identification algorithm. Section III presents the determination of algorithm parameters and experimental results, followed by discussion. Finally, Section IV presents conclusions.

II. METHODS

In this section, the implementation process of this algorithm will be detailed. First, the triangles generation and feature extraction methods are presented. Then, the generation of the feature database and its data storage structure are described. Finally, the details and flowchart of the algorithm are provided.

A. GENERATING MULTI-DIMENSIONAL TRIANGLE FEATURES

In the star identification of triangles, parameters with scale, translation, and rotation invariance are needed as features due to the involvement of multiple coordinate system conversions. Generally, features such as side length, diagonal, height, area, and perimeter are selected for identification, but a single feature cannot provide sufficient information in most cases, especially in the presence of interference, which will lead to erroneous matching. Therefore, it is necessary to introduce new features or filter the combination of existed features. This paper makes full use of the multiple dimensions of triangle features and constructs a filtering and matching framework from high to low dimensions.

1) SELECTION OF TRIANGLES

Assuming there are n navigation stars in the navigation star catalog, theoretically C_n^3triangles can be constructed, which will result in a huge storage and search burden. However, triangles constructed arbitrarily are meaningless due to the constraint of FOV. In addition, there are problems in selecting any three stars from all the stars in FOV to form a navigation triangle. On the one hand, too many triangles will reduce the differences within the features, which results in more mismatches and longer matching times. On the other hand, random star points will lose some important edge information. Therefore, the proposed algorithm takes each navigation star as the center of the FOV and all stars satisfying the FOV are sorted by their distance from the center of the FOV. The stars nearest to the central star are selected as the neighboring star to form the triangle. As shown in Fig.1, a star Sm is chosen as the central star and three surrounding stars S1, S2 and S3 that meet certain conditions forms three feature triangles with the central star.

Once the position of a star among the three points for forming a triangle is established, all edges do not need to be matched anymore and the workload can be reduced to one-third of the original in the subsequent ''edge-edge-edge'' joint screening, which simplifies the matching process and

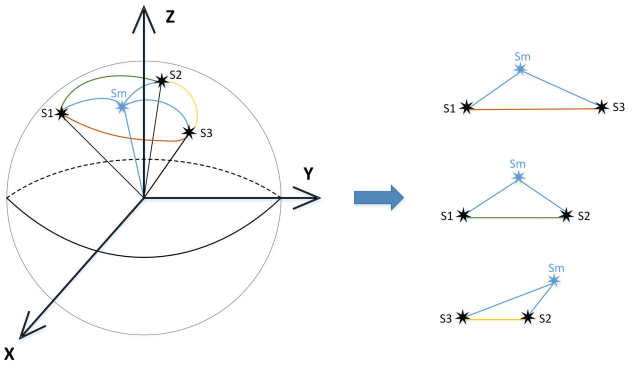

FIGURE 1. Construction of navigation triangles. Sm is the central star and S1, S2, S3 are neighboring stars.

reduces the matching range greatly. Also, the position of the star point will be more accurate and the identification accuracy will be higher in the actual scenario when the distortion of the star sensors' lens is minimal at the center of FOV. What's more, the robustness against star magnitude noise can get greatly improved when distance is preferred because the star magnitude is relatively inaccurate compared to the star position during imaging.

Considering that stars that are too close to the central star may form binary stars and extreme features. A small circle with a radius of R_min and a large circle with a radius of R_max are constructed for each central star as a restriction, and star within the two circles are selected as candidate neighboring stars. As shown in Fig.2, compared with the stars S2 and S6, the star Sm is closest to the image center O. When Sm is selected as the central star, ten triangles generated by combining the five stars S1, S2, S3, S4, and S5 around the central star. Any one of these triangles can independently solve the spatial attitude of the star sensor at this time, which accelerates the identification and improves the robustness greatly.

Additionally, information on star magnitude should be utilized as an important supplement, as brighter stars are generally considered more reliable. The steps for selecting neighboring stars are as follows: if the number of stars in FOV is greater than twice N_center, then the N_range closest stars to the central star are selected as potential neighboring stars, then the brightest N_center stars are selected as the central star's neighbors and sorted by brightness. If the number of stars in FOV is less than twice N_center, then the brightest N_center stars are directly selected as the central star's neighbors and sorted by brightness. If the number of stars in FOV is less than N_center, then all stars are considered as the central star's neighbors. All neighboring stars are paired with the central star to form triangles, which not only satisfy the requirement for high robustness but also narrow down the matching range.

2) FEATURE EXTRACTION

The new triangular algorithm is based on the extraction of various features, ranging from high-dimensional to

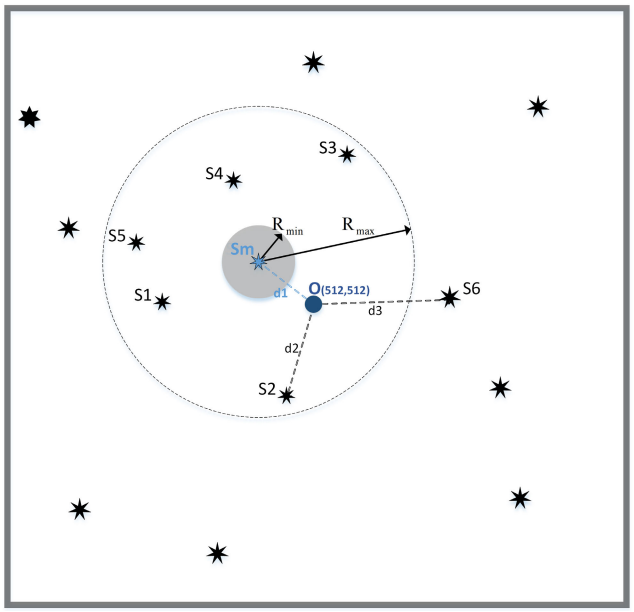

FIGURE 2. Selection of Central and Neighboring Stars. The star Sm, which is closest to the image center O, is selected as the central star. With the central star as the center of the circle, the limited FOV is constructed and neighboring stars are selected.

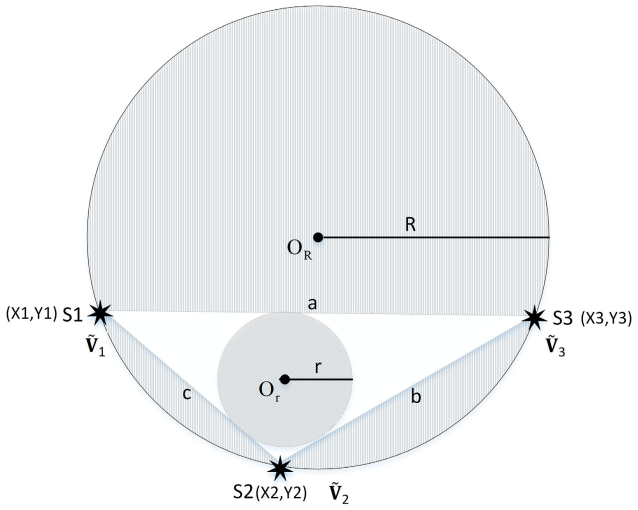

FIGURE 3. Inscribed and circumscribed circles of the triangle. The three stars S1, S2, and S3 form a navigation triangle. The coordinates of the three stars in the image coordinate system are (X1,Y1), (X2,Y2), (X3,Y3), and the star vectors in the celestial coordinate system are $\tilde{V}_1$, $\tilde{V}_2$, and $\tilde{V}_3$.

low-dimensional values which include the shape factor, area, polar moment, edge length, and pixel information of the star position. Then the multi-level screening is constructed through a selective process using these features from lenient to strict. The shape factor 'h' of a triangle is defined as the ratio of the inradius to the circumradius of the triangle and only dependent on the edge lengths of the triangle, which has the lower computational complexity and excellent discrimination ability. Therefore, it is utilized as the first level of hierarchical screening. As shown in Fig.3, assuming the three edges of the triangle are 'a', 'b' and 'c', with a perimeter of '2p' and an area of 'S', where 'a' is the longest edge, then

The inradius of a triangle is

$$r = \frac{S}{P} \tag{1}$$

The circumradius of a triangle is

$$r = \frac{abc}{4S} \tag{2}$$

The shape factor 'h' of a triangle is

$$h = \frac{r}{R} = \frac{4S^2}{pabc} \tag{3}$$

According to Cole et al. [22], the area and polar moment of a triangle are complementary high-dimensional feature information because two planar triangles with the same area may have significantly different polar moments. The proposed algorithm fully exploits the physical significance of the joint constraint relationship between the the area and polar moment. In the second screening level, joint matching is employed, where only triangles that satisfy both area and polar moment constraints can enter the third screening level. When matching planar triangles seen in FOV of a star tracker with those in the database, these two methods will quickly reduce the number of possible triangles to be matched.

According to the Heron's formula, the area(S) of a triangle is given by

$$S = \sqrt{p(p-a)(p-b)(p-c)} \tag{4}$$

The polar moment(M) of feature triangle is given by

$$M = S\left(a^2 + b^2 + c^2\right)/36 \tag{5}$$

where the semiperimeter(p) of the triangle is given by half the sum of the triangle's three sides a, b and c,

$$\begin{aligned} p &= \frac{1}{2}(a+b+c) \\ a &= \left\|\tilde{\mathbf{V}}_2 - \tilde{\mathbf{V}}_3\right\| \\ b &= \left\|\tilde{\mathbf{V}}_1 - \tilde{\mathbf{V}}_3\right\| \\ c &= \left\|\tilde{\mathbf{V}}_2 - \tilde{\mathbf{V}}_1\right\| \end{aligned} \tag{6}$$

To calculate the lengths of the sides of the triangle, which are based on the star vectors of two stars, the vectors $\tilde{V}_1$, $\tilde{V}_2$ and $\tilde{V}_3$ corresponding to the stars S1, S2 and S3 in the celestial coordinate system are used. Assuming that the coordinates of the star S1 in the image coordinate system are (X1, Y1), and the focal length is f, then the side lengths of the triangle can be calculated as

$$\tilde{\mathbf{V}}_1 = \frac{1}{\sqrt{X_1^2 + Y_1^2 + f^2}} \begin{pmatrix} X_1 \\ Y_1 \\ -f \end{pmatrix} \tag{7}$$

By substituting the area, the inradius r and circumradius R can be obtained as

$$R = \frac{abc}{4\sqrt{p(p-a)(p-b)(p-c)}}$$
$$r = \sqrt{\frac{(p-a)(p-b)(p-c)}{p}} \tag{8}$$

Substituting the area into the formula for the shape factor h,

$$h = \frac{4(p-a)(p-b)(p-c)}{abc} \tag{9}$$

For the sake of computational convenience, the shape factor can be simplified as

$$h = \frac{(p-a)(p-b)(p-c)}{abc} \tag{10}$$

Apart from the pixel positions of star points, the lowest dimension feature of a triangle is its three sides, which, also use a joint matching approach at this screening level like area and polar moment. Only triangles that satisfy the joint constraints of the three sides can enter the final matching stage.

B. ONBOARD DATABASE GENERATION

The onboard database of this algorithm consists of a Reference catalog C and a pattern database containing triangle features (shape factor, area, polar moment).

1) NAVIGATION STAR CATALOG

The star catalog is an indispensable part of star identification based on star sensors, containing tens of thousands of stars, each with parameters such as position, brightness, and spectrum, which can be used to establish a star triangle feature database. For celestial navigation, we mainly focus on the coordinates and magnitudes of stars. Considering the attitude and optical parameters of the star sensor, star images can be simulated easily using the star catalog. The detectable magnitude of the star sensor in this paper is about 6.0 Mv, the FOV of the optical system is 15° × 15°, and the image plane is 1024 pixels × 1024 pixels, with each pixel occupying 15/1024°. If the distance between two stars on the CCD plane is less than 10 pixels, or less than 0.15°, they are defined as double stars.

Catalog C is obtained by screening the SKY2000 catalog for stars with a brightness higher than the limiting magnitude (6Mv) and removing interference from double stars, including the star number, right ascension, declination, and apparent magnitude information of 4915 stars. The catalog, which consists of star number (16 bits), right ascension (32 bits), and declination (32 bits), occupies a total of 48.0KB (4915*(16+32+32) bits) of memory.

2) NAVIGATION TRIANGLE FEATURE DATABASE

The calculation of the lengths of the sides and the transformation matrix used in this paper requires vectors in the celestial coordinate system that make up the three stars forming the navigation triangle. Vectors in the celestial coordinate system can be directly obtained from the right ascension and declination of the stars in the navigation star catalog. Considering the relatively low computational and the burden direct storage of star vectors, the triangular feature database foregoes the storage of each star's vector. Therefore, the navigation triangle feature database only stores the three high-dimensional features of the navigation triangles.

The preliminary construction of the navigation feature database can be completed using the method in section before, and the navigation triangles generated by the same central star are also stored in order of decreasing brightness. However, the number of triangles in the navigation feature database is usually large, and traditional retrieval methods cannot quickly retrieve. To improve the retrieval speed of the navigation feature database and avoid searching the entire navigation triangles, a hash function was used to devide the shape factors. The navigation feature database was divided into NR blocks and the Index_h was used as the block coefficient. 'h' represents the hash function value of this navigation triangle, which indicates the block number of the feature triangle in the navigation feature database, ranging from 1 to NR. That is, a mapping function is established to map the shape factor h of each triangle in feature database block N(h) in a certain way, and the mapping can be obtained as

$$N(h) = \text{fix}\left(\frac{h + 0.1250}{Index\_h} + 1\right) \tag{11}$$

The feature database of navigation triangles occupies a total of 866.25KB (49280*(163+323) bits) of memory. It should be noted that in the past, the size of on-board memory was very limited, so it was important to pursue a lower database in algorithm design to reduce the required memory cost. However, embedded systems can easily store 5 MB of data, the memory size gradually becomes a less important part of the algorithm compared with the identification rate and running time in star identification technology.

Therefore, the onboard feature database occupies approximately 914.25KB (866.25KB+48KB) of memory, and the storage structure is shown in Fig.4.

C. STAR IDENTIFICATION ALGORITHM

Most star identification algorithms currently select the most similar navigation star after matching it with observed stars, and consider it a successful match. If there are multiple solutions, follow-up verification is needed, or select the closest observed stars for verification. One unique aspect of the proposed algorithm is that its goal is not to search for a unique solution, but rather to construct a sufficient number of possible solutions while generating a feature database and processing observed star picture. Any one of these possible solutions that passes through the algorithm is considered a successful match which means that even in the presence of significant noise, there is always a possibility of finding a matching solution. Correspondingly, theoretically, each star

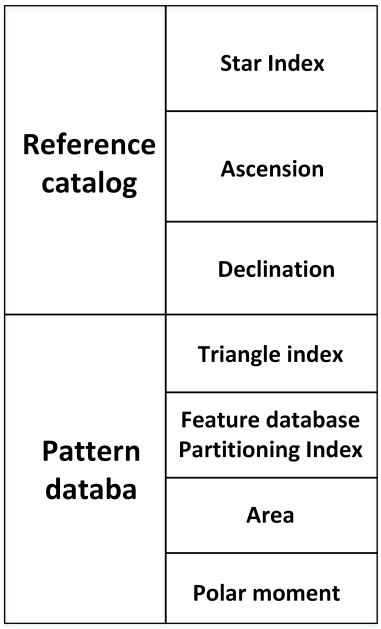

FIGURE 4. Structure of the onboard database.

point in FOV will select five neighboring stars in a certain way to form ten feature triangles with the central star, and each triangle can represent the position of three star points. Each star in FOV can be a central star, and every triangle generated by each central point can be a possible successful match. After maximizing the robustness of the algorithm, the algorithm's computation time is an another important factor to consider.

To enhance the identified speed, this algorithm differs from most other algorithms in that it matches the entire feature triangle. A multi-layer joint matching method based on multi-dimensional feature is employed to screen all triangles' features in the feature database. A multi-dimensional identification architecture is constructed through a multi-layer screening approach that ranges from high-dimensional features such as shape factors, area, and polar moment to low-dimensional features such as triangle sides and star pixel positions. The purpose of each layer of matching is to narrow down the range of triangles for the next layer of matching, rather than to find the final solution. The algorithm uses a joint matching approach for triangle area and polar moment, as well as triangle side length, to reduce the potential solutions while maintaining tolerance for noise. In the final layer of the algorithm, any potential solutions are matched in sequence using the smallest dimension of pixel differences among star points. Successful matching is achieved as long as the criteria are met, without consideration for optimal solutions.

1) MULTI-LAYER JOINT SCREENING AND VALIDATION OF MULTI-DIMENSIONAL FEATURES

This algorithm makes full use of various features of the triangles from high dimensionality to low dimensionality to perform screening. High-dimensional features have greater generality, and larger thresholds are used at high-dimensional features to exclude options that are significantly different from the navigation triangles, while retaining sufficient redundancy for noise. The joint feature screening makes full use of the physical complementary properties of different features and can quickly improve the screening speed. Low-dimensional features are the source of high-dimensional features and have higher accuracy. The main purpose of each layer of matching is to screen rather than determine a unique solution. Finally, based on the calculated attitude, the star catalog within FOV is projected onto the detector plane and compared with the observed image. Matching is considered successful if it satisfies the requirements of the star position features.

The first feature of the matching process is the shape factor of the triangle. The corresponding feature database block number can be obtained by inputting the factor of the observed triangle into a hash function. All matching triangles that meet the criteria are then input into the second layer of matching. To improve robustness against noise, the blocks of the two adjacent feature database are selected as the first set of matched triangles.

The second matching process involves the joint screening of triangle area and polar moment. The area describes the size of the observed triangle, and the polar moment serves as a complementary descriptor of its shape. The joint screening makes use of the complementary nature of the features, where only those triangles that meet the threshold values of T_A and T_M for the difference in area and polar moment with the observed triangle can be retained for the next step.

The third matching step utilizes the triangle's three sides as low-dimensional features. Similarly, only candidate triangles whose differences in side lengths with the navigation triangle and feature triangle are less than the threshold T_dij can be considered. It should be noted that the triangles are formed using the central star, and only two matching steps are needed instead of six, as the diagonal edges of the central star are fixed. During matching, only the positions of the two adjacent edges of the central star need to be considered in case of noise-induced swapping.

In the final matching step, the lowest-dimensional feature, the pixel position of the star on the star map, is used. The rotation matrix quickly calculated by the dual-vector positioning method. Then the star catalog within the FOV are projected onto the detector plane based on the calculated attitude. The star points in the FOV constrained by the central star of this triangle in the projected star map are matched with the corresponding star points in the observed star map. If the distance between the matched points is less than the N_pixel threshold, these points are considered matched. If the number of matched star points exceeds the specified threshold N_point, the entire star map is considered to be successfully matched with high probability.

Finally, all star positions within the entire 15°x15° FOV are verified. The simulated star picture of the observed star

TABLE 1. Parameters in the proposed algorithm and brief descriptions.

Parameters	Description
R_max & R_min	The range of neighboring stars to select when constructing navigation triangles
N_center & N_rangey	Selecting the number and order of neighboring stars when constructing navigation triangles
R_max_map & R_min_map	Maximum and minimum angular distance between the central star and its neighbors when matching
N_center_map & N_range_map	Selecting the number and order of neighboring stars when matching
Index_h	The hashing function's block coefficient
T_A & T_M	The threshold for area and polar moment values
T_dij	The threshold for triangle edge lengths
N_pixel	The threshold for star point positions

map is obtained through the rotation matrix and the stored star position information in the star catalog and all stars are needed to be matched. The match is considered successful if the distance between the matched stars is less than the defined N_pixel value. If the number of matched stars exceeds a certain ratio(N_percentage) of all stars, the entire star map is considered successful.

2) ALGORITHM PARAMETERS

The required parameters for this algorithm are shown in Table 1. These parameters can be roughly divided into two categories: (1) those primarily responsible for algorithm speed, such as T_A, T_M, T_dij, N_center_map and N_range_map, (2) those primarily related to accuracy, such as R_max, R_min, N_pixel, N_point, N_percentage and (3) Index_h, N_center and N_range are particularly relevant to the algorithm's accuracy and speed, especially storage space. The specific selection of algorithm parameters will be detailed later.

3) ALGORITHM FLOW

The input of this algorithm consists of the coordinates and magnitudes of the stars in the image coordinate system. The first step is to select the most likely stars to construct the observation triangle. Take the process of constructing navigation triangles as reference, the first step is to calculate the distances of all stars from the image center point, and sort them in order of increasing distance.

On the imaging plane with size of 1024pixel × 1024pixel, locate the star closest to the center of the image as the central star Sm, and select stars within the range of radius greater than R_min and less than R_max as candidate neighboring stars. If the number of stars in FOV filtered is greater than twice N_center_map, the N_range_map stars closest to the central star are selected, and then select the brightest N_center_map stars among them as neighboring stars of central star, sorted by brightness. If the number of stars in FOV is less than twice N_center_map, directly select the brightest N_center_map stars as neighboring stars of central star, sorted by brightness. If the number of stars in the field of view is less than N_center_map, all stars are selected as neighboring stars of central star.

Combine the central star with the two brightest neighboring stars to form an observation triangle. Extract the shape factors, area, polar moment, side length, and star position of the observation triangle, and perform joint screening according to the description in section II-C1. If all screening or verification fails, combine the central star with other bright neighboring stars in pairs to form other observation triangles, and repeat the matching step. If all generated triangles do not meet the criteria, change the central star. Select other stars closest to the image center as the central star, repeat the above steps until verification succeeds or matching fails. The specific algorithm flowchart is shown in Fig.5.

III. RESULTS

The performance of the algorithm is evaluated by simulation test, star identification from satellite in orbit and star identification test from consumer-grade camera. It was compared with the improved triangle algorithm, the grid algorithm, and the radial and cyclic algorithm. These algorithms were tested using MATLAB on a PC with an Intel(R) Core(TM) i9-9900K CPU @ 3.60GHz, running on a Microsoft Windows environment.

A. SELECTION OF PARAMETERS

The images used to test each parameter in Sections III-A1 through 3.1.9 were generated from 5,000 random directions created by the Monte Carlo method [23]. Different star position noises were used as the testing variables. The images used to test the robustness of the entire algorithm in Sections III-B1 through 3.2.3 were generated from 10,000 random directions created by the Monte Carlo method. The simulation platform configuration in this section is shown in Table 2.

1) THE SELECTION OF R_max & R_min

The values of R_max and R_min determine how many neighboring stars within FOV are selected to form each navigation triangle for each central star, which are two important parameters when generating the feature database. To avoid the influence of binary stars and extreme triangles, R_min is

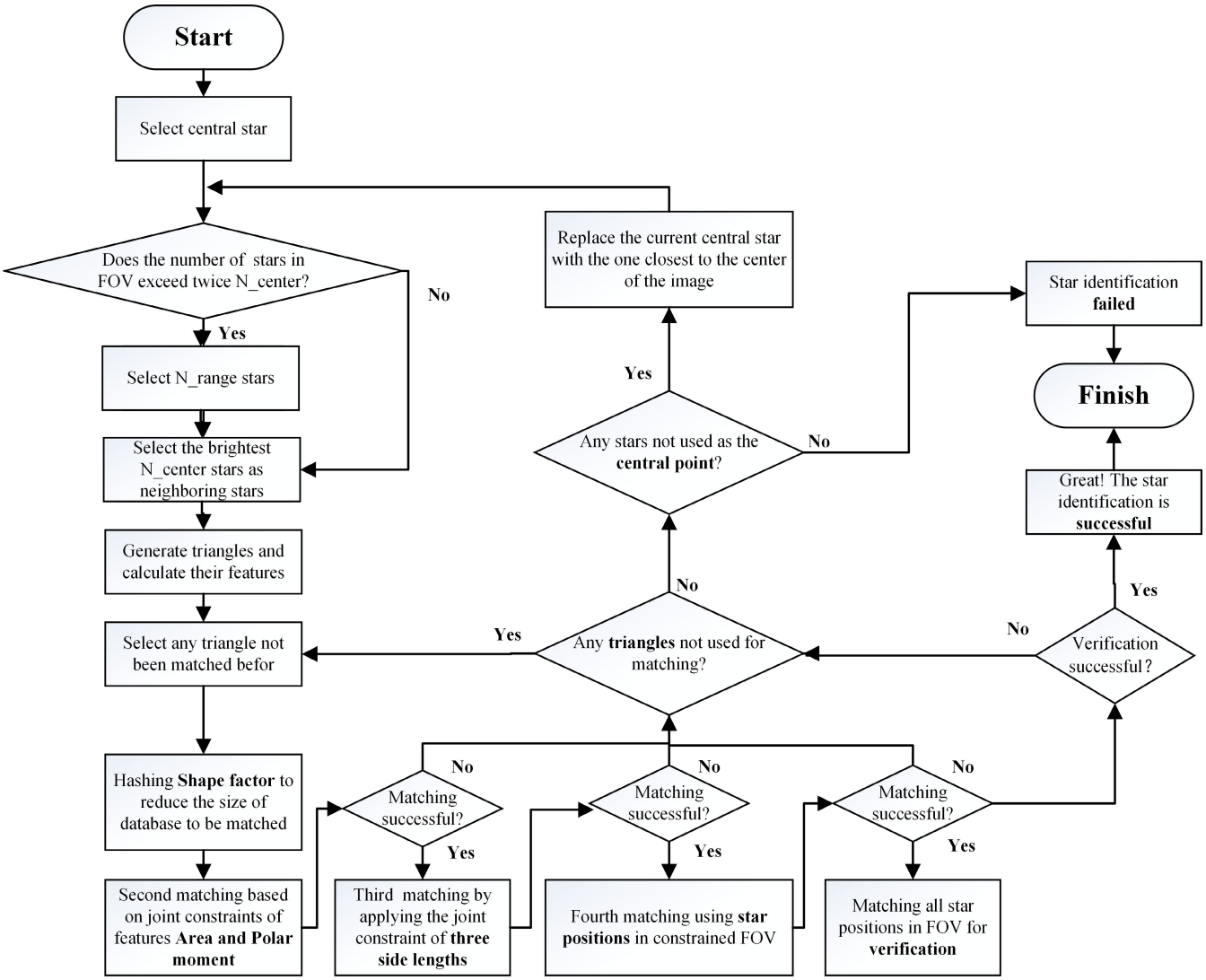

FIGURE 5. Flowchart of the star identification algorithm. Every star closest to the center of the image is selected as the central star to pick candidate neighboring stars, and the feature triangle that is most likely to be recognized is constructed. Then, all possible solutions are reversely projected to the image plane after multi-dimensional and multi-layered joint screening of triangular shape factor, area, polar moment, three-side length, etc. As long as the star position meets the requirements, the identification is successful.

TABLE 2. Parameters for simulations.

Parameters	Value
Focal length	49.37mm
Field of view(FOV)	15°×15°
Resolution	1024pixel × 1024pixel
Pixel size	0.013mm
Max. magnitude	6Mv

chosen to be as large as possible, in our case 1°. A too small R_max is meaningless, as at least three stars are needed to form a navigation triangle, so we need to ensure that there are at least two or more neighboring stars around each central star. Although a larger R_max can ensure that there are more neighboring stars available for selection around a central star, too much overlap of neighboring stars between different central stars may cause interference. This simulation compared R_max values of 4, 5, 6, 7 and 8, and Fig.6 indicates that when N_max is 1 or 2, a large number of central stars cannot form navigation triangles. Taking all factors into consideration, this algorithm adopts an R_max of 7.

2) THE SELECTION OF N_center AND N_range

The parameters N_center and N_range are crucial in determining the number of neighboring stars used to construct a navigation triangle for each central star. They play a vital role in the selection of subsequent parameters and determine the upper limits of both the accuracy and speed of the identification algorithm. A larger N_center produces a vast feature database, and provides sufficient redundancy during matching, but poses significant challenges for storage as well,

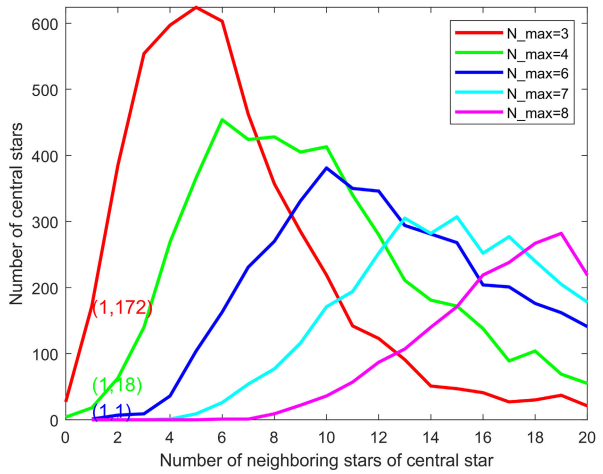

FIGURE 6. The relationship between number of central stars and number of neighboring stars under different R_max.

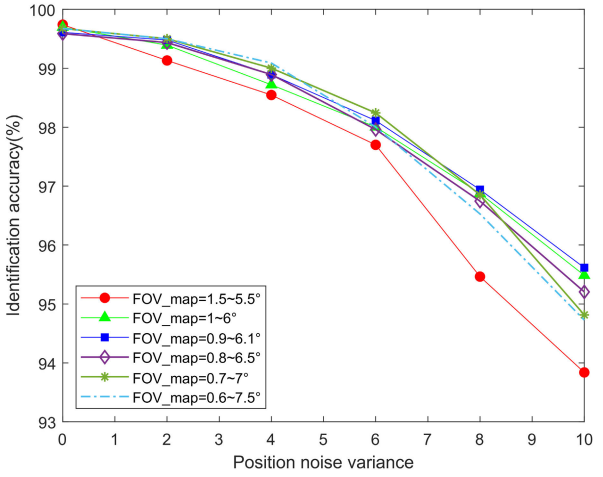

FIGURE 8. The relationship between R_max & R_min and identification rate under different position noise.

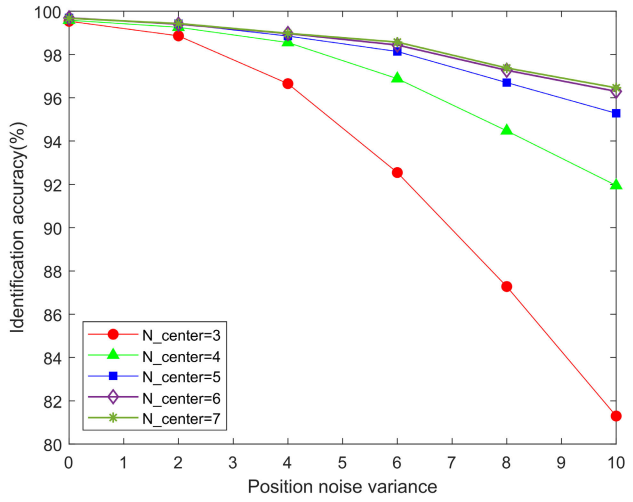

FIGURE 7. The relationship between N_center and identification rate under different position noise.

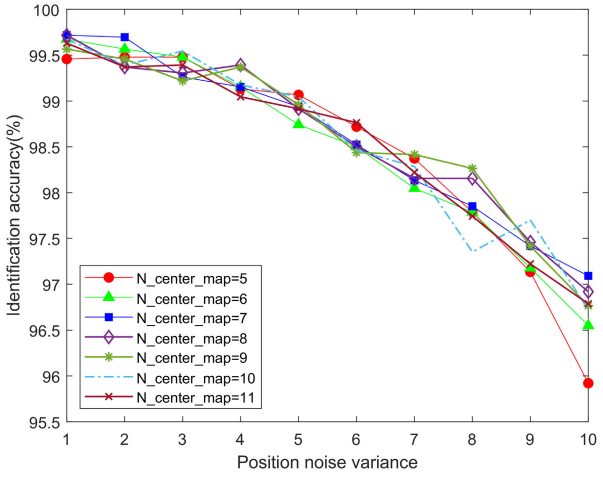

FIGURE 9. The relationship between N_center_map & N_range_map and identification rate under different position noise.

particularly matching speed. Fig.7 compares N_center values of 3, 4, 5, 6 and 7, revealing that identification rates only slightly improve with more than six stars, but require more matching time, while identification rates drop significantly below five stars. Therefore, this algorithm employs N_center of 5 and N_range of 7, based on the comprehensive consideration of these factors.

3) THE SELECTION OF R_max & R_min

The usable FOV of each observation star is determined by R_max and R_min. A large FOV will significantly increase computation time, while a small FOV will cause interference between binary stars. In theory, considering the star noises and to include all possible matched observation stars, the FOV should be slightly larger than that used to generate the feature database. However, it can be seen from the simulation results in Fig.8 that that a FOV range of 0.9 to 6.1 degrees provides slightly better accuracy and runtime performance in the presence of significant star point location noise.

4) THE SELECTION OF N_center_map & N_range_map

N_center_map and N_range_map determine the range and order of the selected observation stars during matching, and they need to be determined based on the corresponding N_center and N_range parameters used when generating the feature database. Through simulation, it has been found in Fig.9 that when N_center_map is set to 7, it performs better in both the normal working range where the positional noise of the stars is less than 2 and in extreme errors. Taking all factors into account, N_center_map=7 is selected.

5) THE SELECTION OF Index_h

The shape factor of the feature triangle is the first criterion in the matching process, which effectively screening out most unsuitable triangles. The value of Index_h determines the

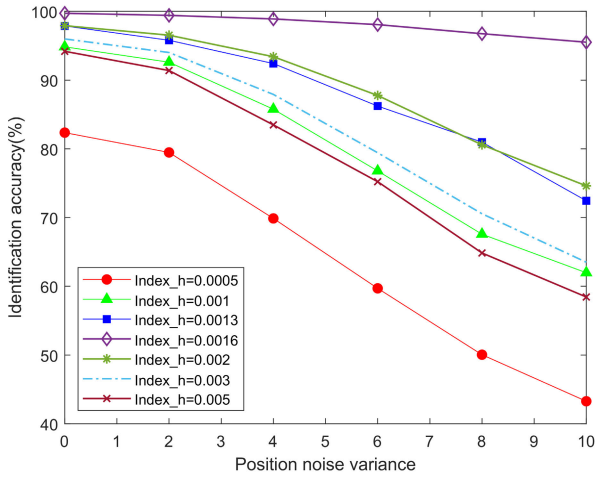

FIGURE 10. The relationship between Index_h and identification rate under different position noise.

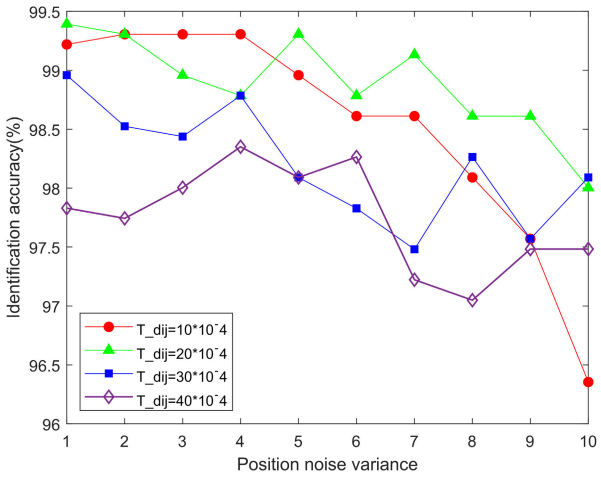

FIGURE 11. The relationship between T_dij and identification rate under different position noise.

number of partitions constructed, and reflects the robustness against noise. Even slight changes to this value can have a significant impact on the matching accuracy and time which can be seen from Fig.10. Through the simulation, both the accuracy and the running time reach the optimum when Index_h coefficient is 0.0016.

6) THE SELECTION OF T_dij

The joint matching of the three sides of the triangle is the third step of algorithm matching and the first matching of low-dimensional features. Considering the robustness requirement of the algorithm, it is better to choose a slightly larger T_dij. If T_dij is too small, it will greatly reduce the accuracy of the matching. However, too large T_dij cannot play its necessary role, consuming a lot of time and leading to a lower matching rate. Simulation results show that when the T_dij is between 10*10ˆ-4 and 20*10ˆ-4, the accuracy is not significantly different with a small star magnitude noise. When the noise is high, 20*10ˆ-4 has a significant advantage. Therefore, this algorithm selects T_dij as 20*10ˆ-4, as shown in Fig.11.

7) THE SELECTION OF N_pixel

Due to the imprecise nature of dual-vector attitude determination and the need for noise tolerance, a larger N_pixel threshold is necessary when projecting onto the image plane for the lowest feature dimension matching. Fig.12 indicates that identification accuracy will be significantly affected if the pixel threshold is lower than 7 pixels and larger pixel thresholds do not significantly increase matching time. Therefore, the pixel threshold of 13*10ˆ-4.

8) THE SELECTION OF N_point & N_percentage

The final step of matching and the validation process both involve directly comparing the positions of stars on a plane, with the only difference is the size of the range.

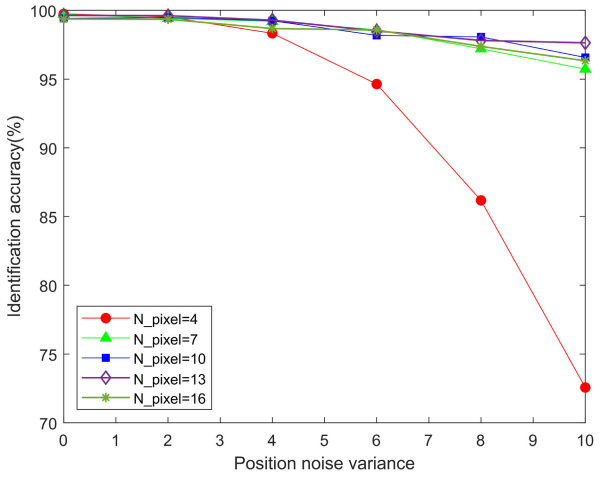

FIGURE 12. The relationship between N_pixel and identification rate under different position noise.

Clear simulation results in Fig.13 show that to increase the robustness of the entire algorithm, successful matching only requires two or more stars within the restricted FOV. This approach can significantly enhance the accuracy of identification in high-noise environments while reducing matching time. The simulation also demonstrates that increasing the number of star matches in the validation process can improve noise tolerance without significantly increasing matching time. Therefore, the N_percentage of 50% is selected.

B. ALGORITHM PERFORMANCE ANALYSIS

Interference noise refers to the position noise, the magnitude noise of stars and false stars. The position deviation of stars mainly comes from calibration errors of star sensors, such as errors in focal length measurement, lens distortion, and errors in optical axis deviation. The magnitude noise reflects the sensitivity of star sensors to the brightness of stars. In the subgraph isomorphism algorithm category, we choose the mature

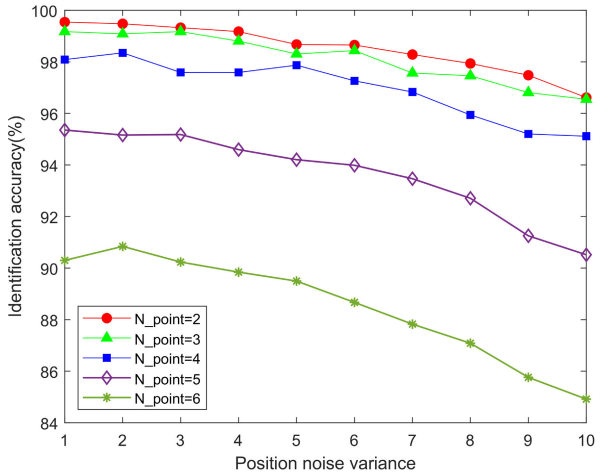

FIGURE 13. The relationship between N_point & N_percentage and identification rate under different position noise.

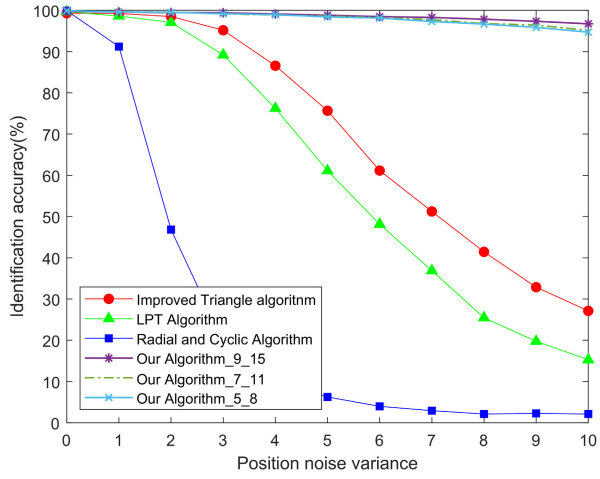

FIGURE 14. Identification rate for different levels of position noise.

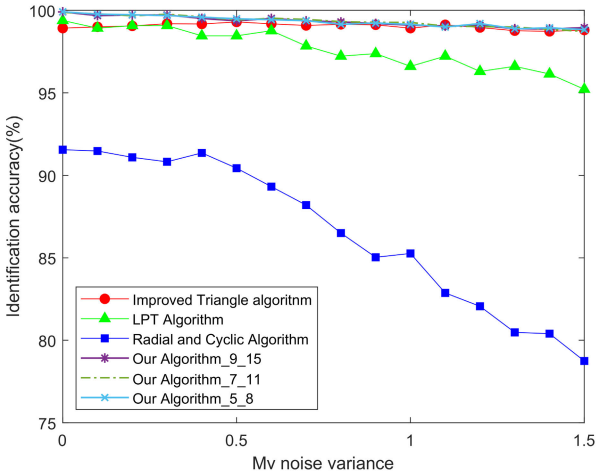

FIGURE 15. Identification rate for different levels of magnitude noise.

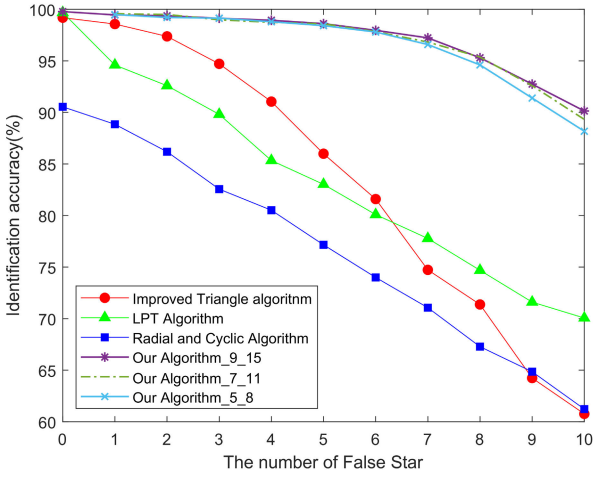

FIGURE 16. Identification rate for different number of false stars.

improved triangle algorithm for comparison. In the pattern recognition algorithm category, we choose the representative LPT algorithm and the radial and cyclic algorithm for comparison. In order to better show the upper limit of recognition accuracy of this algorithm, we adopted N_center_map and N_range_map as three parameters (9,15), (7,11) and (5,8) respectively.

1) ROBUSTNESS TO THE NOISE OF THE STAR POSITION

The optical distortion of sensor lens, quantization error and sub-pixel centroid algorithm will produce certain star position noise when extracting star position. To simulate this environment, a Gaussian noise standard deviation ranging from 0 10 pixels (0 0.146°, with 1 pixel being 0.0146°) is added to the vertical and horizontal centroid coordinates of each star. Meanwhile, the standard deviation of brightness noise remained at 0.322Mv. Fig. 14. displays the results for five different algorithms.

As shown in the Fig.14, compared to the other four methods, our algorithm exhibits extremely high robustness against star position noise. Even with star position noise reaching 10 pixels, this algorithm can still achieve identification success rates of 94.7% and even 96.7% if more time given, which is significantly better than other algorithms. When the star position noise is within 5 pixels, the improved triangle algorithm can achieve identification rates of over 90%, with a significantly slower decline rate than the LPT algorithm. Also, the radial and cyclic algorithm is more sensitive to star position noise.

2) ROBUSTNESS TO THE NOISE OF MAGNITUDE

Due to interference such as light or jitter, the magnitude of the stars will generate noise when sensor imaging, in this case, some darker star points may not be observed and interference to algorithms that rely heavily on magnitude will be greater. As it is impossible to precisely measure the brightness of stars, random Gaussian noise is added to the star's magnitude. Fig.15 shows the comparison of the accuracy of the four

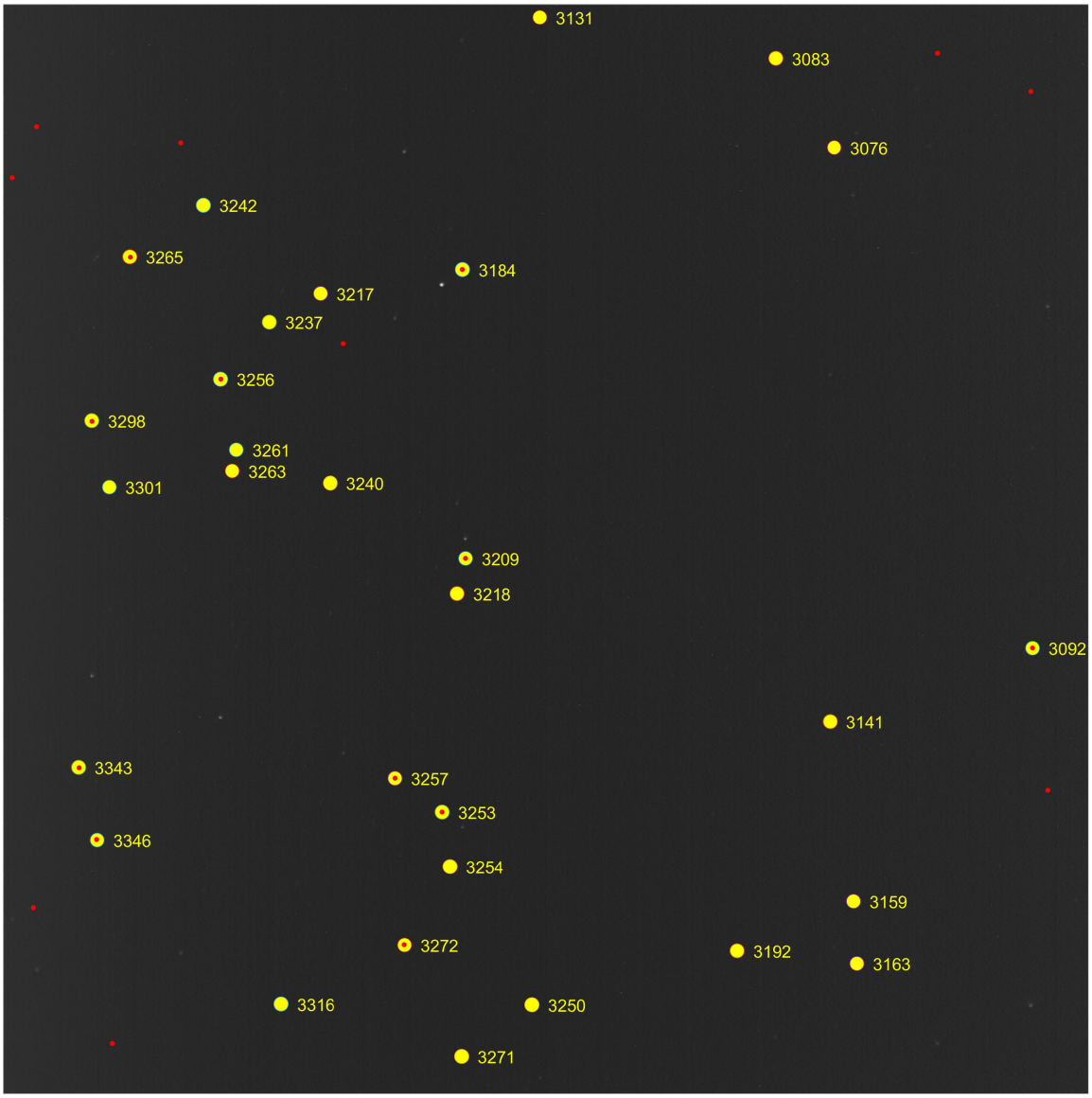

FIGURE 17. A star image downloaded from an orbiting satellite. The small red dot is the star point extracted after image preprocessing. The large yellow circles and the yellow numbers are the star points and asterisk projected by the database after the algorithm is successfully identified.

algorithms when the star point position error is kept at 1pixel, the magnitude noise variance is from 0 to 1.5Mv, and the interval is 0.1Mv.

Since our algorithm mainly uses the position information of stars, the magnitude information is only used to assist in extracting the matching stars. The identification rate of the proposed algorithm can achieve over 99% even under 1.5 Mv noise interference for its high robustness to star magnitude noise. The LPT algorithm experiences a rapid decline in identification rate when the star magnitude noise exceeds 0.7 Mv, while the improved triangular algorithm is less sensitive to magnitude and is less affected by such noise.

3) ROBUSTNESS TO FALSE STARS

Due to factors such as single event upset, sensor aging, and thermal drift, false stars can be produced during imaging. Fig.16 shows the identification accuracy of four algorithms when 0 to 10 false stars with magnitudes ranging from 3Mv to 6Mv are randomly added to simulate the star map, where the positional noise variance of the stars is kept at 1 pixel. The proposed algorithm demonstrates its robustness against false stars which can be verified from the identification rate of over 96% when the number of false stars in FOV is less than 7. In contrast, the compared algorithms are more sensitive to false stars and exhibit a faster decline in identification accuracy.

TABLE 3. Parameters in the proposed algorithm and brief descriptions.

Algorithm	Identification rate	Average time(ms)	Memory cost(KB)
The proposed algorithm	99.51%	7.6	914.25
Improved triangle algorithm	98.50%	12.4	746.67
LPT algorithm	97.07%	15.71	137.08
Radial and cyclic algorithm	46.84%	8.43	543

(a) (b)

FIGURE 18. Star sensor and environment of the night sky experiments. The camera is mounted on a tripod with the lens facing up. In order to verify the robustness of the proposed algorithm, the environment with heavy light pollution and more noise is selected.

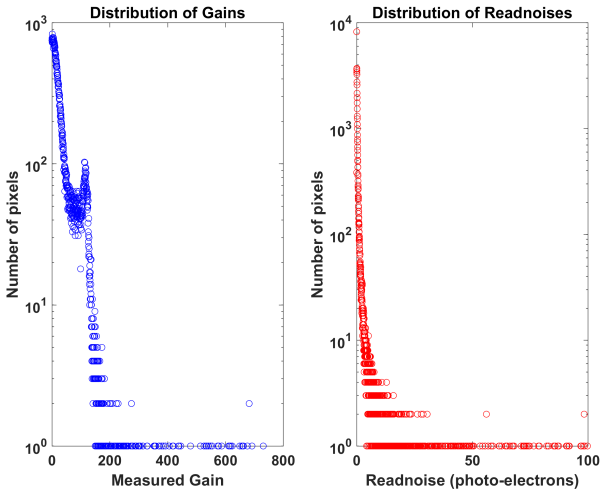

FIGURE 19. The gain and readout noise of each pixel of the device under ISO 12600.

4) STORAGE AND RUN-TIME PERFORMANCE

In practical applications of star identification algorithms, storage and identification time are important performance parameters. The identification rates, average time, and memory requirements of four algorithms under star position noise and star magnitude noise of 2 pixels and 0.322 Mv, are shown in Table 3. The proposed algorithm achieved the optimal recognition rate with the shortest time, which demonstrates the significant advantages. However, a total memory size of 914.25 KB is required, which is slightly larger than the other three algorithms due to more redundant triangles establishment. Nevertheless, it is still acceptable for practical use.

TABLE 4. Parameters for orbiting star sensor.

Parameters	Value
Focal length	43.3mm
Field of view(FOV)	20.0°×20.0°
Resolution	1024pixel × 1024pixel
Pixel size	0.015mm
Exposure time	0.25s

C. STAR IDENTIFICATION OF SATELLITE IN ORBIT

In order to verify the availability of the algorithm in orbit, 1000 images of a remote sensing satellite in the Sun synchronous orbit launched last year were selected for identification. The results show that all 1000 star images are successfully identified. And the average processing time of each image is only 28ms, which meets the requirement of on-orbit star identification. The lens parameters of the orbiting satellite sensor are shown in Table 4.

The identification results of a randomly selected in-orbit star image is shown in Fig.17. The small red dots are 20 star points extracted by the algorithm in the image. The 30 large yellow circles are all theoretical star positions on this image after successful matching. There are 9 false stars in this map, and only 11 pairs of extracted stars coincide with projected stars. The number of false stars accounts for 81.82% of all matched stars. At the same time, 19 star points that should be identified in theory were not extracted, and the missing star accounted for 172.73% of all matched star points. The orbiting star sensor testing shows that the algorithm is robust to both false stars and vanishing stars.

D. STAR IDENTIFICATION TEST CAPTURED ON THE GROUND

To verify the robustness of our algorithm and assess its performance under the influence of increased star position noise and environmental noise, highly light-polluted ground-based tests

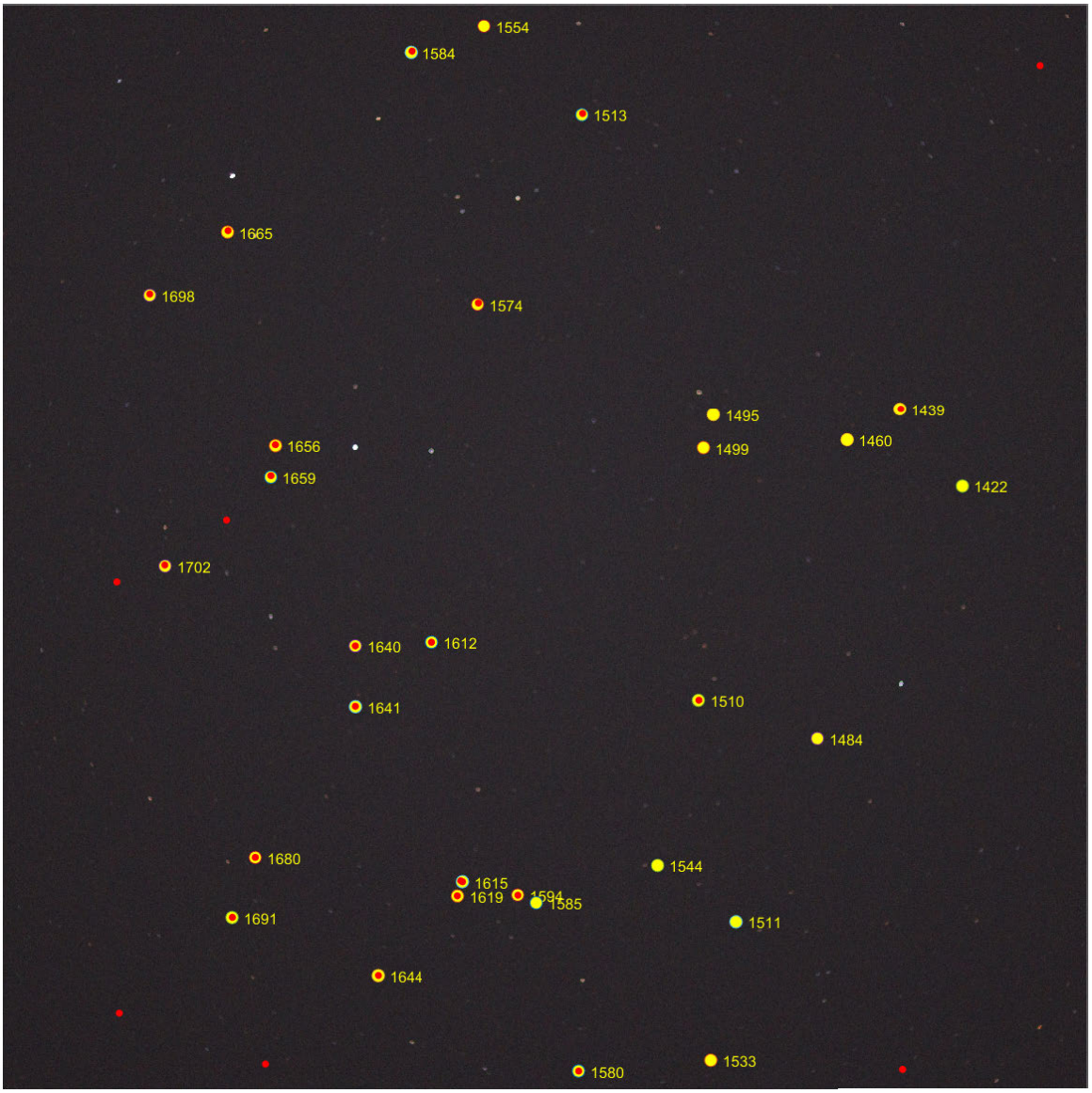

FIGURE 20. An example of star image by consumer-grade camera. The background is the sky image directly output by Canon camera. The small red dot is the star point extracted after image preprocessing. The large yellow circles and the yellow numbers are the star points and asterisk projected by the database after the algorithm is successfully identified.

are conducted using a consumer-grade camera. The camera is mounted on a tripod with the lens facing up. The real star images used in the tests were all captured with a Canon EOS 90D camera every 30 seconds, and all of them were successfully recognized through the proposed algorithm.In this shooting, the lens distortion is not corrected, and there is a certain focal length error and a large principal point offset. The star sensor used in the real sky image test is shown in Fig.18, and its configuration is presented in Table 5.

Because the device needs to shoot less bright starlight under light pollution at night, the camera's sensitivity ISO is set to 12800, and the exposure time is set to 0.5s. Background noise has a great impact on imaging results. Through our calculation, the mean value of background noise is 54.14, and the variance is 104.06. In order to better describe the noise mode of the shooting device, we calculated the gain and readout noise of each pixel of the device under this ISO, as shown in Fig.19.

The identification result of a randomly selected star image by consumer-grade camera is shown in Fig.20. The 27 red dots are the stars extracted by the algorithm from the image. The 30 yellow circles represent the theoretical stars from the navigation star catalog projected onto the real star image through a rotation matrix after a successful match. There are 21 pairs of overlapping real and projected stars in this image with a significant amount of noise. Despite adjusting the star extraction algorithm specifically, there are still 6 false stars, accounting for 28.57% of all correct stars, and 9 theoretically

TABLE 5. Parameters for consumer-grade camera.

Parameters	Value
Focal length	50mm
Real focal length	49.74mm
Principal point offset(x,y)	(78pixel , 8pixel)
Field of view(FOV)	16.4°×16.4°
Resolution	4640pixel × 4640pixel
Pixel size	0.0031mm
ISO	12800
Exposure time	0.5s
Mean of background noise	54.14
Variance of background noise	104.06

identifiable stars that were not extracted, accounting for 42.86% of all correct stars. This star identification algorithm took a total of 20.16ms, demonstrating the algorithm's strong robustness against noise in star positions, star magnitudes, false stars, and missing stars.

IV. CONCLUSION

A novel algorithm for star identification based on multi-dimensional features and multi-layer joint screening is proposed in this paper, which is designed for star maps with large position noise in LIS mode. The algorithm creatively uses star positions as the last matching conditions and verification conditions, by constructing a multi-layer screening method from high-dimensional features such as the shape factor, area, polar moment of triangles to low-dimensional features such as the length of triangle sides and star point positions. The use of the multi-dimensional matching architecture not only ensures efficient identification time but also significantly enhances the robustness of star identification. Besides, we perform a layered screening of potential matches. First, we quickly get the navigation feature database sub-block using hash function through the observed triangle's shape factor. Then, we use two high-dimensional features (triangle area and polar moment) in combination to screen potential matches. Rotation matrix will be calculated after three triangle sides joint screening are used. After that, the last screening for triangles will be used by comparing the star position between projected stars with stars in image. Any triangle needs to be verified after it has passed all the matching. If it passes the verification, the matching is successful.The match is successful once the conditions of any navigation triangle are met. Although this algorithm has significantly better identification accuracy than several other comparative algorithms, it still suffers from certain problems, such as algorithm complexity and excessive parameters. In the future, we can consider adding more features, such as centroid inertia ratio, Angle, etc., to build a richer multi-dimensional screening network to further improve the robustness and accuracy of the algorithm. In addition to the optimization of the star identification algorithm, the hardware noise of the sensor itself, such as lens distortion, is also a problem to be solved to improve the accuracy.

The star identification in orbit proves that the algorithm has strong robustness to disappearing stars and false stars. Simulation experiments and ground-based star identification tests also show that the algorithm can be successfully worked under harsh conditions such as high star position noise and magnitude noise. It can be considered to transplant the program after the algorithm is optimized, so as to realize the lightweight and miniaturization of the system, which can be used in space-borne or missile-borne and other fields. Facing the production of mass low-cost satellites and the large-scale launch of constellation satellites in the future, this paper provides the algorithm basis for cheap and possibly poor image quality star sensors.

REFERENCES

[1] S. Wang, "A survey of several common navigation methods for spacecrafts," *Acad. J. Sci. Technol.*, vol. 4, no. 1, pp. 67–72, Dec. 2022.

[2] L. He, W. Ma, P. Guo, and T. Sheng, "Developments of attitude determination and control system of microsats: A survey," *Proc. Inst. Mech. Eng., I, J. Syst. Control Eng.*, vol. 235, no. 10, pp. 1733–1750, Nov. 2021.

[3] D. Rijlaarsdam, H. Yous, J. Byrne, D. Oddenino, G. Furano, and D. Moloney, "A survey of lost-in-space star identification algorithms since 2009," *Sensors*, vol. 20, no. 9, p. 2579, May 2020.

[4] C. Padgett and K. Kreutz-Delgado, "A grid algorithm for autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 1, pp. 202–213, Jan. 1997.

[5] J. Li, X. Wei, G. Wang, and S. Zhou, "Improved grid algorithm based on star pair pattern and two-dimensional angular distances for full-sky star identification," *IEEE Access*, vol. 8, pp. 1010–1020, 2020.

[6] G. Zhang, X. Wei, and J. Jiang, "Full-sky autonomous star identification based on radial and cyclic features of star pattern," *Image Vis. Comput.*, vol. 26, no. 7, pp. 891–897, Jul. 2008.

[7] X. Wei, G. Zhang, and J. Jiang, "Star identification algorithm based on log-polar transform," *J. Aerosp. Comput., Inf., Commun.*, vol. 6, no. 8, pp. 483–490, Aug. 2009.

[8] S. Yang, L. Liu, J. Zhou, Y. Zhao, G. Hua, H. Sun, and N. Zheng, "Robust and efficient star identification algorithm based on 1-D convolutional neural network," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4156–4167, Oct. 2022.

[9] D. Rijlaarsdam, H. Yous, J. Byrne, D. Oddenino, G. Furano, and D. Moloney, "Efficient star identification using a neural network," *Sensors*, vol. 20, no. 13, p. 3684, Jun. 2020.

[10] B. Wang, H. Wang, and Z. Jin, "An efficient and robust star identification algorithm based on neural networks," *Sensors*, vol. 21, no. 22, p. 7686, Nov. 2021.

[11] H. Wang, Z.-Y. Wang, B.-D. Wang, Z.-Q. Yu, Z.-H. Jin, and J. L. Crassidis, "An artificial intelligence enhanced star identification algorithm," *Frontiers Inf. Technol. Electron. Eng.*, vol. 21, no. 11, pp. 1661–1670, Nov. 2020.

[12] J. Jiang, L. Liu, and G. Zhang, "Star identification based on spider-web image and hierarchical CNN," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 4, pp. 3055–3062, Aug. 2020.

[13] Y. Wang and H. Zhang, "Star recognition based on mixed star pattern and multilayer SOM neural network," in *Proc. IEEE Aerosp. Conf.*, Mar. 2017, pp. 1–6.

[14] L. Xu, J. Jiang, and L. Liu, "RPNet: A representation learning-based star identification algorithm," *IEEE Access*, vol. 7, pp. 92193–92202, 2019.

[15] E. A. Hernández, M. A. Alonso, E. Chávez, D. H. Covarrubias, and R. Conte, "Robust polygon recognition method with similarity invariants applied to star identification," *Adv. Space Res.*, vol. 59, no. 4, pp. 1095–1111, Feb. 2017.

[16] G. Zhang, X. Wei, and J. Jiang, "Star map identification based on a modified triangle algorithm," *Acta Aeronautica Et Astronautica Sinica-Ser. A B-*, vol. 27, p. 1150, 2006.

[17] D. Mortari, J. L. Junkins, and M. Samaan, "Lost-in-space pyramid algorithm for robust star pattern recognition," *Guid. Control*, vol. 2001, pp. 49–68, 2001.
[18] Y. Zhao, X. Wei, J. Li, and G. Wang, "Star identification algorithm based on K–L transformation and star walk formation," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5202–5210, Jul. 2016.
[19] L. Sun and Y. Zhou, "MVDT-Si: A multi-view double-triangle algorithm for star identification," *Sensors*, vol. 20, no. 11, p. 3027, May 2020.
[20] J. Du, X. Wei, J. Li, G. Wang, and C. Zang, "Star identification based on radial triangle mapping Matrix," *IEEE Sensors J.*, vol. 22, no. 9, pp. 8795–8807, Mar. 2022.
[21] M. S. Scholl, "Star field identification algorithm: Performance verification using simulated star fields," *Proc. SPIE*, vol. 2019, pp. 275–290, Oct. 1993.
[22] C. L. Cole and J. L. Crassidis, "Fast star-pattern recognition using planar triangles," *J. Guid., Control, Dyn.*, vol. 29, no. 1, pp. 64–71, Jan. 2006.
[23] J. Paul Roth, "Diagnosis of automata failures: A calculus and a method," *IBM J. Res. Develop.*, vol. 10, no. 4, pp. 278–291, Jul. 1966.

HAODONG YAN received the B.S. degree in electronic and information engineering from ShanghaiTech University, Shanghai, China, in 2018. He is currently pursuing the Ph.D. degree in computer science with the Innovation Academy for Microsatellites, CAS, Shanghai, and ShanghaiTech University. His main research interests include star sensor and image processing.

XURUI CHEN received the B.S. degree in space information and technology from Xidian University (XDU), Xi'an, China, in 2019. He is currently pursuing the master's degree in information and communication engineering with the Innovation Academy for Microsatellites, CAS, Shanghai, China. His main research focuses on star sensors and image processing.

GUOPENG DING received the Ph.D. degree from the University of Chinese Academy of Sciences (UCAS), Beijing, China. He is currently an Associate Researcher with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai, China, and UCAS. His current research interests include image intelligent processing and visual measurement.

SHUAI ZHI received the B.S. degree from the School of Physics and Electronic, Hunan University (HNU), China, in 2012, and the M.S. degree from the Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences (CAS). Currently, she is with the Innovation Academy for Microsatellites, CAS. Her research interests include intelligent vision measurement and multi-sources date fusion.

YONGHE ZHANG received the Ph.D. degree from the University of Chinese Academy of Sciences (UCAS), Beijing, China. He is currently a Researcher with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai, China, and UCAS. His research interests include spacecraft dynamics and intelligent control, formation flight and control, intelligent space robotics, and teleoperation.

ZHENCAI ZHU received the Ph.D. degree from Zhejiang University (ZJU). He is currently a Researcher with the Innovation Academy for Microsatellites, Chinese Academy of Sciences, Shanghai, China, and the University of Chinese Academy of Sciences, Beijing, China. His research interests include satellite design and simulation, micro satellite and attitude control, and sensor technology research.