



Article

A Robust High-Accuracy Star Map Matching Algorithm for Dense Star Scenes

Quan Sun , Zhaodong Niu *, Yabo Li and Zhuang Wang

National Key Laboratory of Science and Technology on ATR, College of Electronic Science and Technology, National University of Defense Technology, Changsha 410073, China; sun_quan@nudt.edu.cn (Q.S.)

* Correspondence: niuzd@nudt.edu.cn

Abstract: The algorithm proposed in this paper aims at solving the problem of star map matching in high-limiting-magnitude astronomical images, which is inspired by geometric voting star identification techniques. It is a two-step star map matching algorithm relying only on angular features, and adopts a reasonable matching strategy to overcome the problem of poor real-time performance of the geometric voting algorithm when the number of stars is large. The algorithm focuses on application scenarios where there are a large number of dense stars (limiting magnitude greater than 13, average number of stars per square degree greater than 185) in the image, which is different from the sparse star identification problem of the star tracker, which is more challenging for the robustness and real-time performance of the algorithm. The proposed algorithm can be adapted to application scenarios such as unreliable brightness information, centroid positioning error, visual axis pointing deviation, and a large number of false stars, with high accuracy, robustness, and good real-time performance.

Keywords: star map matching; dense star scenes; robustness; real time; high accuracy



Citation: Sun, Q.; Niu, Z.; Li, Y.; Wang, Z. A Robust High-Accuracy Star Map Matching Algorithm for Dense Star Scenes. *Remote Sens.* **2024**, *16*, 2035. <https://doi.org/10.3390/rs16112035>

Academic Editor: Dusan Gleich

Received: 13 May 2024

Revised: 1 June 2024

Accepted: 4 June 2024

Published: 6 June 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Space environment observation is the basis for ensuring the safety of space activities. As an important part of space environment observation, optical observation of space orbiting objects, improving the accuracy of orbital object celestial fixing, is one of the key technologies to ensure high-precision orbit determination and space environment forecasting. Star map matching is the core technology to achieve high-accuracy astronomical positioning, the key is to find suitable invariant features and use them to match stars in star images and catalogs [1]. Brightness, angle, and their derived features are the basic features of almost all star map matching algorithms [2]. The algorithm proposed in this paper only relies on inter-star angular features, and the test results on the synthetic and measured data show that it solves the star map matching problem well for dense star scenes, and can provide a better basis for the astronomical positioning of orbital objects in images.

There are three main types of algorithms for star map matching: subgraph isomorphism, pattern recognition, and neural network [3]. In the subgraph isomorphism type of algorithm, stars are considered as vertices in a graph, whose edges are generally the angles between neighboring stars, and the matching is considered successful when a subgraph in the star image matches with a subgraph in the database. The main algorithms include triangle algorithms [4–10], and polygon algorithms [11–15] derived from triangle algorithms. The number of candidate combinations for these algorithms is C_N^n , where N is the number of stars in the field of view and n is the number of subgraph vertices; the number of combinations in a dense stellar scene is massive, leading to a lack of real-time performance of the algorithms [16]. Initially most of the algorithms in this type were similar to Liebe's algorithm [17,18], relying only on angular information independent of stellar brightness information, but in order to speed up and reduce mis-matches later on many subgraph

isomorphism-type algorithms chose the combination of angular and magnitude features as the matching features, which resulted in algorithms that were sensitive to magnitude noise [19].

Pattern recognition-type algorithms generally need to select two stars to form the base axis according to the position and brightness, and correlate each star with the patterns or features determined by its neighboring stars. When the similarity between the pattern in the star image and the pattern in the database is the largest, the matching is considered successful, but the false matching rate is high due to the influence of false stars and brightness uncertainty, and the process of establishing the pattern and calculating the similarity is extremely time consuming in the case of a large number of dense stars. Grid algorithms are typical representatives of pattern recognition algorithms and are classified into grid algorithms based on rectangular coordinates [20–24] and grid algorithms based on polar coordinates [25–30]. Compared to subgraph isomorphism-type algorithms, grid algorithms are more suitable for star-rich scenes, but when the limiting magnitude is greater than magnitude 13 (more than 185 stars per square degree), this results in all grids being full of stars, making the feature vectors lack differentiation. The solutions that can be considered include (a) reducing the buffer radius, which leads to a sharp increase in the length of the feature vector for each star, making the memory required to store the catalog massive, and the excessively long vectors will inevitably slow down the matching speed; (b) reducing the pattern radius, which seems to be a reasonable solution, but a large number of faint stars have a very concentrated distribution of brightness, and their imaging quality is highly susceptible to background interference. Therefore, the probability that the closest neighboring star selected when building the grid using the catalogs is consistent with the closest neighboring star in the grid in the image is very low, and the inconsistency of the closest neighboring star will lead to the consequence of mis-matching or failure to match. In addition, rotation-invariant additive vector sequence (RIAV) [31] also belongs to subgraph isomorphism algorithms; it has a certain robustness to position deviation and false stars, but its performance will significantly decrease as the number of stars and magnitude of noise increase. In order to improve the shortcomings of pattern recognition algorithms, Refs. [32,33] used the geometric voting (GMV) strategy to effectively improve the robustness of the algorithms, but its real-time performance is poor when the number of stars is large.

Neural network star map matching algorithms are an important direction in this field in recent years, this type of algorithm usually uses artificial intelligence networks to extract combinatorial features for star map matching; to obtain robust neural network algorithms, over-parameterization is necessary [34]. Therefore, if the neural network star identification algorithms need to have high robustness, the model needs to be large enough, and its requirement of computational resources is often higher than that of traditional algorithms. Ref. [35] utilizes neural networks and robust feature extraction methods for star map matching in the lost-in-space (LIS) state of the star tracker, a database lookup step is not required in the matching process as the neural network implicitly stores patterns associated with the guide stars. Ref. [36] is a star map matching algorithm based on Vgg16 convolutional neural networks. Ref. [37] proposed an end-to-end star recognition network for smeared star scenarios. Ref. [38] constructed a spider web image and proposed a hierarchical convolutional neural network model to recognize the spider web image. Refs. [39,40] are all one-dimensional convolutional neural network (1D CNN)-based star map matching algorithms. Neural networks have shown good potential in the field of star map matching, but at present almost all neural network star map matching algorithms are still only applicable to sparsely distributed star scenarios, in order to ensure robustness in the dense star scenarios they will be faced with the challenge of over-parametrization, resulting in too large a network model and the huge computational resources required.

From the above overview of star map matching algorithms, it can be seen that most of the current algorithms are only applicable to sparse star scenes, and there are many shortcomings in accuracy, robustness, and real-time performance for dense star scenes.

Nowadays, the limiting magnitude of scientific astronomical telescopes for optical observation of orbiting objects is usually better than 13 magnitudes, while most of the limiting magnitudes of star trackers are around 6 magnitudes. From Figure 1, it can be seen that the average number of stars per square degree when the limiting magnitude is equal to 13 is about 185, while the average number of stars per square degree when the limiting magnitude is equal to 6 is only 0.12, i.e., the former is about 1500 times as many as the latter, and the number of stars in a single image of a dense star scenario is often larger than that in a whole catalog of sparse stars. Moreover, in order to observe faint orbital objects, exposure time is usually longer and the stars in the image are striped, resulting in larger positioning errors compared to point-like images. Therefore, it is particularly important to develop star map matching algorithms that are suitable for massive and dense star scenes.

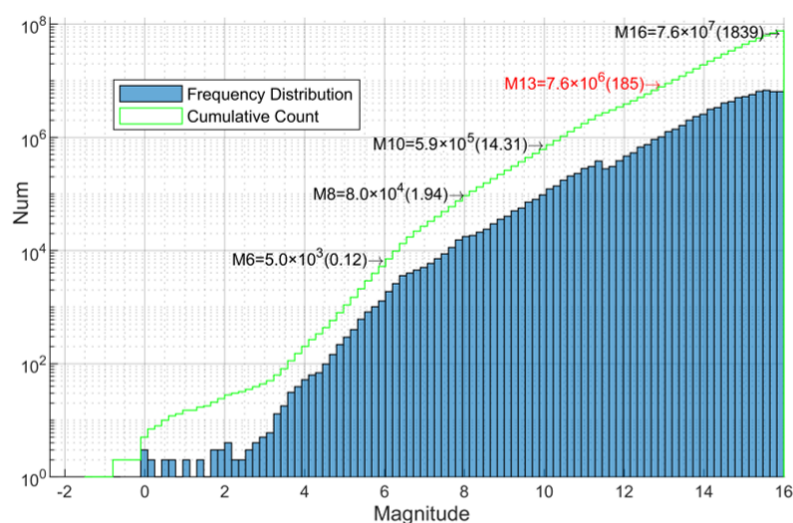


Figure 1. Frequency distribution and cumulative counts of stars of different magnitudes. (The numbers in parentheses are the average number of stars per square degree.)

In general, the difficulties faced in star map matching of orbiting objects' optical observation are mainly in the following three aspects: (1) These factors make it difficult to use brightness as a reliable feature for star map matching because of the unreliability of brightness information due to background interference such as clouds and earth-atmosphere radiation, and the concentration of brightness when the magnitude is large due to the inclusion of a large number of stars; (2) due to CCD stitching, background interference, and not all stars being included in the catalogs, there are a large number of unmatched stars in both the stellar images and the catalogs during the star map matching process (here, unmatched stars in the image include false stars caused by noise and stars not included in the catalog, and unmatched stars in the catalog include stars that are not imaged on the star image although they are within the limiting magnitude due to CCD stitching or other reasons); and (3) long exposures cause motion blur, resulting in large centroid positioning errors. In addition, the center of the field of view is often an orbiting object when the sensor is operating in target tracking mode, which causes significant problems for matching algorithms that select the central bright star as the reference star. Unlike the lost-in-space state, this scenario is usually known to have coarse pointing information, and the inaccuracies in pointing come from three main sources: (1) errors in the reference datum for pointing; (2) errors in the sensor's main optical axis due to manufacturing and assembly; and (3) errors due to the telescope's pointing mechanism (pitch and azimuth rotational axis), and in the case of space-based telescopes, also from errors due to the satellite's attitude adjustments [41].

The star map matching algorithm based on the GMV [32] is a good solution to the above problem, which relies almost exclusively on star angular information and is robust to unreliable brightness information, centroid positioning errors, and false stars. However,

it is designed for star trackers with a sparse star distribution, and faces the following shortcomings when solving the problem of high-density star map matching: (1) GMV votes for all the stars in the star image, which is not a problem when there are very few stars, but when a large number of stars exists, the real-time performance will be seriously affected; and (2) although GMV is highly robust to unmatched stars, when there are massive amounts of unmatched stars, its robustness will still be affected, resulting in a lower matching rate. Aiming at the above deficiencies, this paper proposes an improved algorithm that optimizes the voting process, which not only improves the real-time performance, but also enhances the robustness to a large number of unmatched stars. And an implementation suitable for matrix parallel computing is designed to avoid a large number of loops and judgment instructions in the voting process and further improve the computational speed.

The rest of this paper is structured as follows. In Section 2, we describe in detail the principle and implementation steps of the proposed algorithm. Section 3 deals with the verification of the algorithm performance using simulated and measured data, as well as a comparison with good star map matching algorithms. Section 4 discusses the comprehensive performance of star map matching algorithms. Finally, Section 5 concludes the paper.

2. Star Map Matching

This part is the core of this paper, which treats the star map matching process as two sets of three-dimensional vector transformation processes. The celestial sphere coordinates of any star C_i in the catalog are (α, δ) , then its unit right angle coordinates are $\mathbf{v}_{C_i} = (\cos \alpha \cos \delta, \sin \alpha \cos \delta, \sin \delta)^T$. The image coordinates of any star S_i in the catalog are (x, y) , then its image space coordinates are $\mathbf{v} = (x - x_0, y - y_0, -f_0)^T$, where (x_0, y_0) are the rank coordinates of the center point of the catalog, f_0 is the focal length, and the corresponding unit vector is $\mathbf{v}_{S_i} = \mathbf{v} / \|\mathbf{v}\|$. Let there be N pairs of matched stars in the catalog and star image, forming the two unit vector groups shown in Equation (1):

$$\mathbf{V}_C = \begin{bmatrix} \mathbf{v}_{C_1} \\ \dots \\ \mathbf{v}_{C_N} \end{bmatrix}; \mathbf{V}_S = \begin{bmatrix} \mathbf{v}_{S_1} \\ \dots \\ \mathbf{v}_{S_N} \end{bmatrix}. \quad (1)$$

According to the small-aperture imaging principle, the angles of the star vectors in the catalog and star image are equal, so the transformation relation between matched star pairs can be represented by the transformation matrix \mathbf{T} , shown in Equation (2):

$$\mathbf{V}_C = \mathbf{V}_S \mathbf{T}. \quad (2)$$

From the above analysis, it is clear that the key to matching is to find at least three matching pairs of stars, and then, use the least squares method, as shown in Equation (3), to calculate the transformation matrix \mathbf{T} :

$$\mathbf{T} = (\mathbf{V}_S^T \mathbf{V}_S)^{-1} (\mathbf{V}_S^T \mathbf{V}_C). \quad (3)$$

In the process of calculating \mathbf{T} , the measured data face the following problems: (1) the stars recorded in the catalog are not all extracted in the image due to CCD stitching and background occlusion; (2) the points extracted in the image are not all stars due to the interference of radiation noise and orbital objects; and (3) the stellar angles in the image are not exactly equal to those of the corresponding stars in the catalog due to unavoidable positioning errors. Figure 2 shows the correspondence between catalog and star image stars.

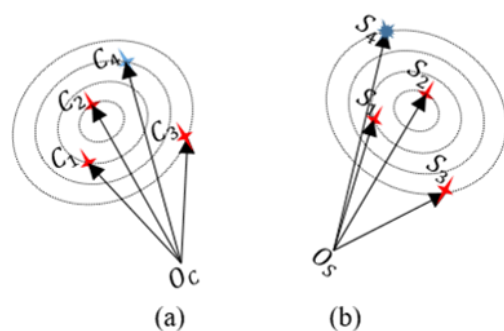


Figure 2. Correspondence between catalog and star image stars. The stars in (a) are located in the catalog, the star points in (b) are extracted from the star image; C_1 , C_2 , C_3 match with S_1 , S_2 , S_3 , and no star points corresponding to C_4 have been extracted from the star image; S_4 is a false star due to noise.

We follow the two-step matching strategy of rough matching first and fine matching later, refer to the more robust geometric voting star recognition technique, and overcome its disadvantage of poor real-time performance when the number of stars is large, and finally, obtain a high-accuracy transformation matrix T , which effectively solves the problem of star map matching in dense star scenarios.

2.1. Rough Matching

The purpose of rough matching is to find three matching pairs of stars, and then, roughly calculate the transformation matrix T . When there are massive and dense stars, the process of determining matching stars is computationally expensive. Therefore, in order to improve real-time performance, only three pairs of matching stars need to be found here—the minimum requirement for calculating the transformation matrix T . This strategy is a significant improvement to the GMV algorithm. GMV tries to find all matching star pairs at once, which requires voting on all stars, and the computational effort is huge. In addition, the one-time voting strategy of GMV may lead to matching failure when there are many unmatched stars, while the strategy in this paper is a feedback iterative process that can avoid this problem as much as possible. The rough matching process in this paper is similar to the RANSAC strategy, with the main difference that the rough matching process in this paper selects the stars in order, whereas RANSAC achieves its goal by repeatedly selecting a random subset of the data. In dense star scenes, the RANSAC strategy faces the disadvantages of difficult subset selection, no upper limit on the number of iterations, and no assurance of obtaining the correct transformation matrix. Compared to RANSAC, the rough matching process in this paper avoids the randomness of the algorithm, there exists an upper limit on the number of iterations, and it can ensure obtaining the correct transformation matrix (if the correct transformation matrix exists). In addition, since the rough matching process in this paper is similar to Liebe's method [17,18], in that the three matching star pairs are selected based on angles, it is necessary to emphasize the main differences between the two in order to avoid confusion. Firstly, the angular features used for identification are different, the angular features chosen in this paper are only the angle distances between stars, while the angular features of Liebe's method include two angle distances and a spherical triangle inner angle. Secondly, the search methods used to determine the matching star pairs are different. Simply because the angular features selected in this paper are only the angle distances between stars, this paper can use all the star angle information in the field of view to determine the matching stars using the voting method, which is more robust. Liebe's method, on the other hand, is similar to the typical triangular star identification method, in which the main star and the corresponding first neighboring star and second neighboring star need to be identified first, and then, the candidate matches are found by comparing the features of all the combinations, and when the number of stars and the degree of denseness are very

large, neither real-time performance nor robustness of this strategy can be guaranteed. The principle and implementation process of rough matching are described in detail below.

For any star point S_i extracted from the star image, the problem of finding its matching star C_j in the catalog can be described as the process of taking the maximum value of the matching metric function shown in Equation (4):

$$U_{ij} = \max(D(S_i, C_j), j = 1, \dots, N_C), \tag{4}$$

where U_{ij} denotes the degree of match when star S_i in the image is matched with star C_j in the catalog (the maximum value of the degree of match between S_i and all the stars in the catalog). The matching metric function $D(S_i, C_j)$ is defined as follows. Let the angles between all stars extracted from the star image and S_i be $\mathbf{A}_{S_i} = [A_{S_i, S_1}, A_{S_i, S_2}, \dots, A_{S_i, S_{N_S}}]$, and the angles between all stars in the catalog of the sky region of interest and C_j be $\mathbf{A}_{C_j} = [A_{C_j, C_1}, A_{C_j, C_2}, \dots, A_{C_j, C_{N_C}}]$, then $D(S_i, C_j) = |\mathbf{A}_{S_i} \cap \mathbf{A}_{C_j}|$, where $|\cdot|$ denotes the potential of the set, i.e., the number of elements.

Since there are positioning errors of stars in the star image, it should be noted that the above intersection operation does not require the angles to be exactly equal; as long as the difference between the two angles is within the set threshold, they are considered equal.

In view of the large computational effort required to calculate the above matching metric function when the number of stars is large, the joint sorting algorithm is used here to reduce the computational effort and to further improve the real-time performance by avoiding frequent loops and judgment instructions as much as possible in the specific implementation of the algorithm. Firstly, the angle vectors of the star image and the catalog are joined, and then, the joint angle vectors are sorted, and the sorting result needs to return the number of the original position at the same time, as shown in Figure 3.

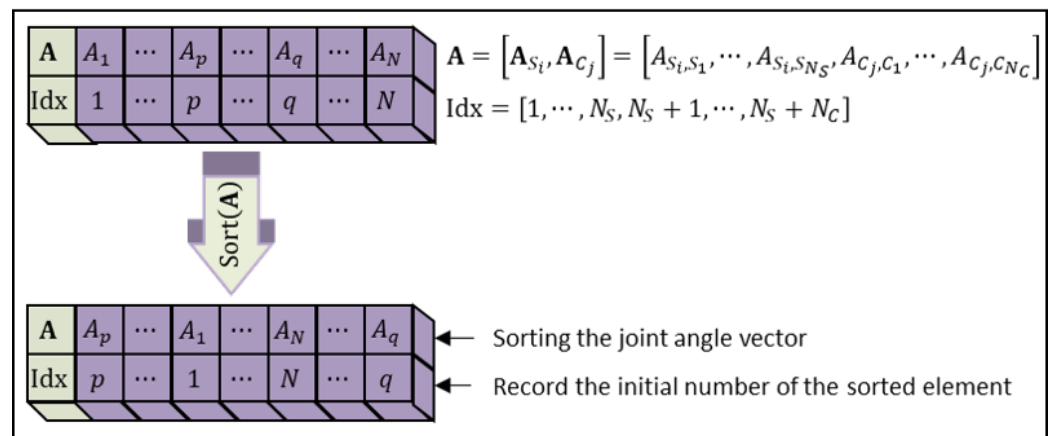


Figure 3. Schematic diagram of joint sorting operation. The star image angle vector is in front and the catalog angle vector is behind to form the joint angle vector. N_S and N_C are the number of elements of the star image angle vector and the catalog angle vector, respectively.

After sorting, it is only necessary to check whether the angular difference between adjacent combinations whose original position number is greater than N_S and the other less than or equal to N_S is less than the set threshold.

The absolute value of the difference between adjacent elements of the angle vector after sorting and the sum of adjacent elements of the original position number vector is calculated, as shown in Equations (5) and (6):

$$\overline{\overline{\nabla}}\mathbf{A} = [|A_k - A_{k-1}|]_{k=2}^{N_S+N_C}, \tag{5}$$

$$\overline{\overline{\text{S}}}\text{Idx} = [\text{Idx}_k + \text{Idx}_{k-1}]_{k=2}^{N_S+N_C}, \tag{6}$$

where A_k and Idx_k denote the value and ordinal number of the k th element, respectively, $\overline{\overline{\mathbf{V}}\mathbf{A}}$ denotes the set consisting of the absolute values of the differences of all neighboring elements, and $\overline{\overline{\text{Idx}}}$ denotes the set consisting of the sum of the ordinal numbers of all neighboring elements. As shown in Equation (7), candidate combinations are selected based on the result $\overline{\overline{\text{Idx}}}$ of Equation (6):

$$\mathbf{F}_M = \text{Bool}(\overline{\overline{\text{Idx}}} > N_S + 1) \& \text{Bool}(\overline{\overline{\text{Idx}}} < 2N_S), \tag{7}$$

here “&” denotes the elementwise AND operation, \mathbf{F}_M is the logical matrix of candidate combinations, and the element that is *true* represents the combination associated with it as a candidate combination. To calculate the matching metric function it is also necessary to eliminate angular differences greater than the threshold and to ensure that there are no one-to-many combinations. Combinations with angular differences greater than the threshold are eliminated using Equation (8):

$$\dot{\mathbf{F}}_M = \mathbf{F}_M \& \text{Bool}(\overline{\overline{\mathbf{V}}\mathbf{A}} < T_A), \tag{8}$$

where $\overline{\overline{\mathbf{V}}\mathbf{A}}$ is the result of Equation (5) and T_A is the set threshold value. The threshold T_A needs to be set according to the pixel scale and the stellar centroid positioning accuracy, and it is recommended to set it to half of the pixel scale when the centroid positioning accuracy is unknown.

Finally, the non-one-to-one combinations are processed and the matching metric function is calculated. To simplify the calculation, this paper adopts the matching degree maximum criterion, which can avoid loops and judgment instructions as much as possible. In fact, the number of matching pairs (matching degree) that can be obtained according to Figure 4 is not unique, and here, the maximum matching degree is the maximum number of matching pairs that can be obtained.

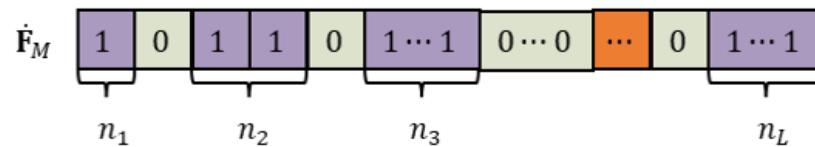


Figure 4. Schematic diagram of the maximum matching degree calculated from $\dot{\mathbf{F}}_M$. L denotes the number of subsegments in $\dot{\mathbf{F}}_M$ whose elements are all *true*, and n_k represents the number of *true* elements in the k th subsegment.

As shown in Figure 4, the number of elements of the subsegments in $\dot{\mathbf{F}}_M$ whose elements are all *true* is counted, and then, the maximum matching degree is calculated as shown in Equation (9):

$$D(S_i, C_j) = \sum_{k=1}^L \left\lfloor \frac{n_k + 1}{2} \right\rfloor, \tag{9}$$

where “ $\lfloor \cdot \rfloor$ ” denotes rounding down. To avoid loops and judgment instructions as much as possible, the following reference algorithm is given in this paper for the process of calculating the maximum matching degree from $\dot{\mathbf{F}}_M$ (Algorithm 1).

Algorithm 1 Calculate Matching Degree**Input:** \mathbf{F}_M // Logical Matrix of Candidate Combinations**Output:** D // Matching Degree

```

1:  $\nabla \mathbf{F}_M \leftarrow \mathbf{F}_M[0 : end - 1] - \mathbf{F}_M[1 : end]$ 
2:  $Idx \leftarrow 1 : \nabla \mathbf{F}_M.length()$ 
3:  $\nabla \mathbf{F}_M \leftarrow \nabla \mathbf{F}_M * Idx$ 
4:  $remove(\nabla \mathbf{F}_M.begin(), \nabla \mathbf{F}_M.end(), 0)$ 
5:  $A \nabla \mathbf{F}_M = abs(\nabla \mathbf{F}_M)$ 
6: if  $\nabla \mathbf{F}_M[0] > 0$  then
7:    $insert(A \nabla \mathbf{F}_M.begin(), 0)$ 
8: end if
9: if  $\nabla \mathbf{F}_M[end] < 0$  then
10:   $tv \leftarrow N_S + N_C - 1$ 
11:   $insert(A \nabla \mathbf{F}_M.end(), tv)$ 
12: end if
13:  $tu \leftarrow A \nabla \mathbf{F}_M[1 : 2 : end] - A \nabla \mathbf{F}_M[0 : 2 : end - 1]$ 
14:  $tu \leftarrow tu + 1$  // Add 1 to all elements
15:  $tu \leftarrow floor(tu/2)$ 
16:  $D \leftarrow sum(tu)$ 

```

Based on the above definition of the matching metric function and its calculation method, the actual operation process of rough matching is described below.

A Calculate the Angle Matrix

The angles of stars extracted from the star image and the angles of stars in the sky region of interest in the catalog are calculated separately to obtain the angle matrices \mathbf{A}_S and \mathbf{A}_C , shown in Equations (10) and (11):

$$\mathbf{A}_S = \begin{bmatrix} A_{S_1, S_1} & \cdots & A_{S_1, S_{N_S}} \\ \vdots & \ddots & \vdots \\ A_{S_{N_S}, S_1} & \cdots & A_{S_{N_S}, S_{N_S}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{S_1} \\ \cdots \\ \mathbf{A}_{S_{N_S}} \end{bmatrix}, \quad (10)$$

$$\mathbf{A}_C = \begin{bmatrix} A_{C_1, C_1} & \cdots & A_{C_1, C_{N_C}} \\ \vdots & \ddots & \vdots \\ A_{C_{N_C}, C_1} & \cdots & A_{C_{N_C}, C_{N_C}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{C_1} \\ \cdots \\ \mathbf{A}_{C_{N_C}} \end{bmatrix}, \quad (11)$$

where N_S and N_C are the number of candidate stars in the image and catalog, respectively. The selection rule for the stars in the catalog in Equation (11) is N_C bright stars are selected in the sky region of interest in the catalog, $N_C = \min(150, 1.5 \times N_S)$, and the sky region of interest is determined by the a priori rough visual axis pointing and the field-of-view angle. When calculating the angle, in order to reduce the calculation, since the square of the difference vector mode is proportional to the angle, it is recommended to use the square of the difference vector mode instead of the angle, as shown in Equation (12):

$$l^2 = \|\mathbf{v}_1 - \mathbf{v}_2\|^2 \propto \langle \mathbf{v}_1, \mathbf{v}_2 \rangle. \quad (12)$$

B Finding Matching Stars

The angular vectors of the three stars extracted from the star image are selected, Equation (4) is used to find the matching stars in the catalog, and the information is recorded as shown in Table 1, where "Num_SM" denotes the star number in the star image,

“Num_SC” denotes the star number in the catalog, and “Deg_Match” denotes the matching degree of the candidate matching stars.

Table 1. Information on suspected matching stars recorded during rough matching.

Num_SM	Num_SC	Deg_Match
S_{i1}	C_{j1}	U_{i1j1}
S_{i2}	C_{j2}	U_{i2j2}
S_{i3}	C_{j3}	U_{i3j3}

A preliminary test of the three candidate pairs of matched stars is performed, using whether the mutual angular differences are all less than a threshold, as shown in Equation (13):

$$\begin{cases} |A_{S_{i1},S_{i2}} - A_{C_{j1},C_{j2}}| < T_A \\ |A_{S_{i1},S_{i3}} - A_{C_{j1},C_{j3}}| < T_A, \\ |A_{S_{i2},S_{i3}} - A_{C_{j2},C_{j3}}| < T_A \end{cases} \quad (13)$$

where the threshold T_A is the same as in Equation (8).

If the preliminary test is not passed, the pair with the lowest matching degree among the three matching pairs is deleted, and then, an unmatched star is selected in the star image and a match is found for that star in the catalog using Equation (4). The appeal step is repeated until three matching pairs are found that pass the preliminary test.

C Determine the Rough Transformation Matrix

First, the transformation matrix for the rough matching process is calculated using the three pairs of matched stars found, as shown in Equations (14) and (15):

$$\mathbf{T} = \left(\mathbf{V}_S^3\right)^{-1} \mathbf{V}_C^3, \quad (14)$$

$$\mathbf{V}_S^3 = \begin{bmatrix} \mathbf{v}_{S_{i1}} \\ \mathbf{v}_{S_{i2}} \\ \mathbf{v}_{S_{i3}} \end{bmatrix}, \mathbf{V}_C^3 = \begin{bmatrix} \mathbf{v}_{C_{j1}} \\ \mathbf{v}_{C_{j2}} \\ \mathbf{v}_{C_{j3}} \end{bmatrix}. \quad (15)$$

Then, the number of all matched stars is counted and a quadratic test is performed on the rough transformation matrix. To count the number of all matching stars, all stars in the chart need to be transformed using the rough transformation matrix \mathbf{T} , as shown in Equation (16). After that, the number of matching stars is counted by calculating whether the angle between the transformed vector in the star image and the vector in the catalog is less than a threshold value.

$$\mathbf{V}_S^T = \mathbf{V}_S \mathbf{T}, \quad (16)$$

where \mathbf{V}_S and \mathbf{T} are shown in Equations (1) and (14), respectively.

The matching star pairs need to satisfy the constraint that any star in the star image has at most one star matching it in the catalog, and any star in the catalog has at most one star matching it in the star image. To satisfy this constraint, the process of counting the number of matching star pairs is as follows.

Calculate the angle matrix between the transformed star vector in the star image and the star vector in the catalog, as shown in Equation (17):

$$\mathbf{A}_{SC} = \begin{bmatrix} A_{S_1,C_1} & \cdots & A_{S_1,C_{N_C}} \\ \vdots & \ddots & \vdots \\ A_{S_{N_S},C_1} & \cdots & A_{S_{N_S},C_{N_C}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{S_1,C} \\ \cdots \\ \mathbf{A}_{S_{N_S},C} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{S,C_1} \\ \cdots \\ \mathbf{A}_{S,C_{N_C}} \end{bmatrix}^T, \quad (17)$$

where A_{S_i,C_j} denotes the angle between the i th transformed star vector in the image (see Equation (16)), and the j th star vector in the catalogs. $\mathbf{A}_{S_i,C}$ and \mathbf{A}_{S,C_j} denote the i th row and the j th column of \mathbf{A}_{SC} , respectively.

Use Equations (18) and (19) to obtain the logical matrix indicating the location of the row and column minima, respectively.

$$\mathbf{F}_C = \text{Bool} \left(\begin{bmatrix} \mathbf{A}_{S_1,C} \\ \dots \\ \mathbf{A}_{S_{N_S},C} \end{bmatrix} == \begin{bmatrix} \min(\mathbf{A}_{S_1,C}) \\ \dots \\ \min(\mathbf{A}_{S_{N_S},C}) \end{bmatrix} \right), \quad (18)$$

$$\mathbf{F}_S = \text{Bool} \left(\begin{bmatrix} \mathbf{A}_{S,C_1} \\ \dots \\ \mathbf{A}_{S,C_{N_C}} \end{bmatrix}^T == \begin{bmatrix} \min(\mathbf{A}_{S,C_1}) \\ \dots \\ \min(\mathbf{A}_{S,C_{N_C}}) \end{bmatrix}^T \right). \quad (19)$$

The logical matrices indicating the location of the row and column minima are elementwise AND to obtain the logical matrix indicating the candidate matches, as shown in Equation (20):

$$\mathbf{F} = \mathbf{F}_C \& \mathbf{F}_S. \quad (20)$$

Use Equation (21) to check whether the angle value of all candidate matches in \mathbf{A}_{SC} indicated in \mathbf{F} is smaller than the matching threshold T_A . If it is smaller, it is judged as a match, otherwise it is judged as no match, and the corresponding element in \mathbf{F} is set to *false*.

$$\dot{\mathbf{F}} = \mathbf{F} \& (\mathbf{A}_{SC} < T_A). \quad (21)$$

If $\dot{\mathbf{F}}_{ij} = \text{true}$, this means that the star in the catalog that has the smallest angle to the star S_i in the star image is C_j , and similarly the star in the star image that has the smallest angle to the star C_j in the catalog is S_i , and the angle between the two is small enough (less than the threshold) so that the two can be considered to be a match. Therefore, the number of *true* elements in $\dot{\mathbf{F}}$ is the number of matching star pairs that need to be counted. The rough transformation matrix is tested again using the counted number of matched star pairs. If the number of matched star pairs is greater than the matching threshold T_N , the rough match is judged to be successful, otherwise new matched star pairs need to be found until it passes the test. In this paper, the setting is $T_N = \max(100, \min(N_S, N_C)/2)$, in which N_S and N_C are the number of candidate stars in the image and catalog, respectively. In summary, Figure 5 gives the detailed flow of the rough matching process.

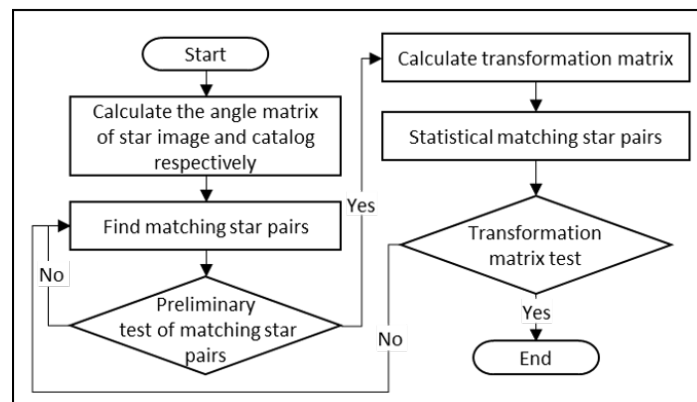


Figure 5. Rough matching process flow chart.

2.2. Fine Matching

The accuracy of the transformation matrix obtained by the rough matching process is limited due to the stellar centroid localization error. In this paper, the least squares

iterative algorithm is used to further improve the accuracy of the transformation matrix. The process uses adaptive thresholding to delete matched star pairs with large errors and uses least squares iterative techniques to continuously improve the matching accuracy of the transformation matrix. The vectors of matched star pairs obtained from the rough matching process, as shown in Equation (22), are used as the iterative initial values for fine matching.

$$\mathbf{V}_C^0 = \begin{bmatrix} \mathbf{v}_{C_1} \\ \dots \\ \mathbf{v}_{C_{N_0}} \end{bmatrix}; \mathbf{V}_S^0 = \begin{bmatrix} \mathbf{v}_{S_1} \\ \dots \\ \mathbf{v}_{S_{N_0}} \end{bmatrix}. \quad (22)$$

The iterative initial transformation matrix for the fine matching process is calculated using the least squares method, as shown in Equation (23):

$$\mathbf{T}^0 = \left(\mathbf{V}_S^{0T} \mathbf{V}_S^0 \right)^{-1} \left(\mathbf{V}_S^{0T} \mathbf{V}_C^0 \right). \quad (23)$$

The matching error is calculated using Equation (24):

$$\mathbf{E}^0 = \left\| \mathbf{V}_C^0 - \mathbf{V}_S^0 \mathbf{T}^0 \right\|_2^r = \begin{bmatrix} \left\| \mathbf{v}_{C_1} - \mathbf{v}_{S_1} \mathbf{T}^0 \right\|_2 \\ \dots \\ \left\| \mathbf{v}_{C_{N_0}} - \mathbf{v}_{S_{N_0}} \mathbf{T}^0 \right\|_2 \end{bmatrix}, \quad (24)$$

where $\|\cdot\|_2^r$ denotes a row-by-row computation of 2-paradigm numbers. Adaptive thresholds are calculated using Equation (25):

$$t^0 = \overline{\mathbf{E}^0} + k_t \sigma_{\mathbf{E}^0}, \quad (25)$$

where $\overline{\mathbf{E}^0}$ and $\sigma_{\mathbf{E}^0}$ represent the expectation and standard deviation of the error \mathbf{E}^0 , respectively, and k_t is the adjustment coefficient (it was set to 0.5 for the tests in this paper).

Matching star pairs with errors greater than the adaptive threshold are deleted, and the transformation matrix is recalculated until the adaptive threshold is less than the set matching accuracy threshold T_M .

3. Synthesis and Measured Data Testing

The specifications of the testing platform are as follows: CPU, Montage Jintide(R) C6248R×4; RAM, 128 GB. This platform is produced by super cloud (Beijing, China) Technology Co., Ltd.

3.1. Introduction to Comparison Algorithms

Currently, there are few star map matching algorithms applicable to dense stellar scenes in the accessible literature, and after sufficient analysis, RIAV [34] and GMV [37] are selected as comparison algorithms in this paper. RIAV can be regarded as an improved grid algorithm, which overcomes the problem of redundant or non-unique features of the grid method to a certain extent; the GMV algorithm is a pattern recognition type of algorithm aiming to improve robustness, and the algorithm proposed in this paper draws on its ideas in designing the matching strategy. In view of the fact that no suitable neural network star map matching algorithm for dense star scenes has been found, such algorithms are not included in the comparison algorithms of this paper. During the test, because the original feature calculation process of RIAV is too cumbersome, this paper optimizes it, and the feature calculation results before and after the optimization are the same; in addition, in order to improve the problem that the robustness of RIAV is too poor in the dense star scenario, only relatively bright star images are input when the RIAV algorithm is running (accounting for 20% of the total number of stars extracted from the image).

3.2. Introduction to Test Data

This section carries out a comprehensive and rigorous test of the star map matching algorithm using both synthetic and measured data.

3.2.1. Synthetic Data

The synthetic data are mainly used for a comprehensive test of the algorithm's performance in a specific scene. The parameters of the synthetic data are field of view of 2.5 degrees, image size of $1\text{ k} \times 1\text{ k}$, image bit depth of 16 bits, limiting magnitude of 13 Mv, 90% of the energy of the star image is concentrated in the 3×3 pixel region, the maximum diameter of the out-of-focus blur is 7 pixels, the length of the star image streak is 15 pixels, and the streak angle is 45 degrees. The image background parameters were set by statistically counting 1000 measured image backgrounds at different times; see [42].

Random noise generated by cosmic rays is simulated using Gaussian-blur-processed salt-and-pepper noise, and high-frequency random noise such as bad points is simulated using salt-and-pepper noise. The brightness distribution of these noises is set to be consistent with the brightness distribution of stars in the catalog. The proportion of noise to total pixels, and the size of star image magnitude deviation and positioning deviation are set according to different testing scenarios. The data synthesis process is shown in Figure 6.

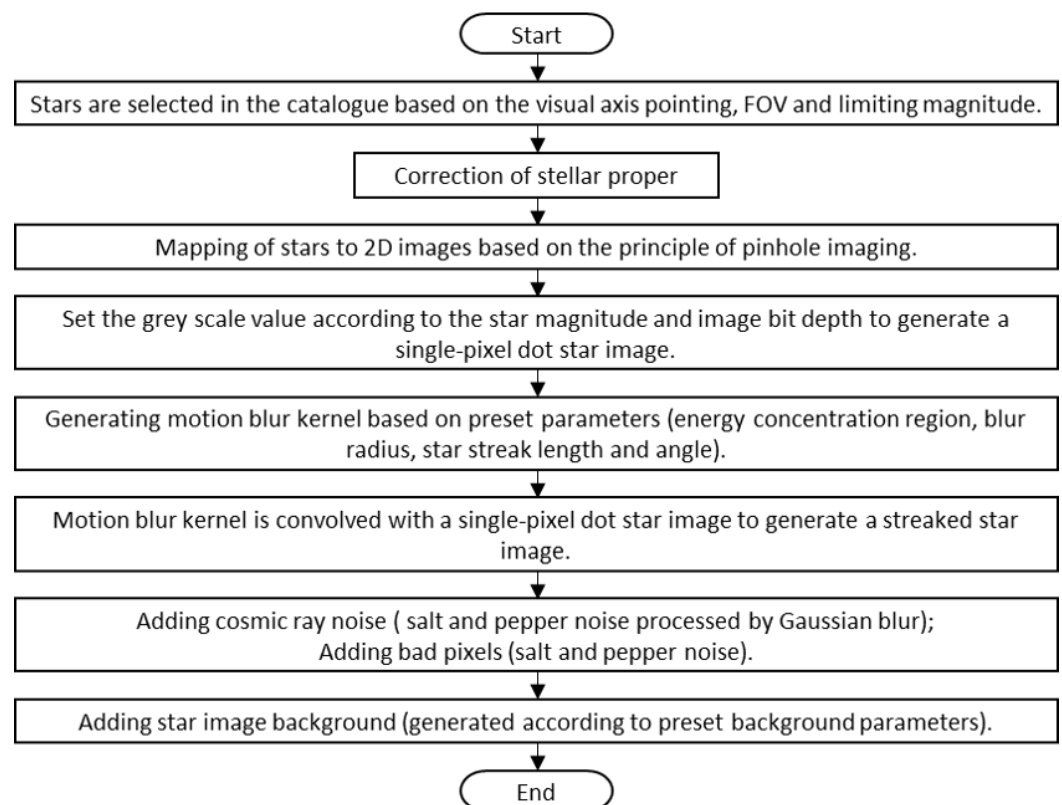


Figure 6. Block diagram of the data synthesis process.

In order to ensure the completeness of the catalogs as much as possible, the catalogs used in the simulation are the union of Tycho-2, UCAC5 (The fifth U.S. Naval Observatory CCD Astrograph Catalog) and BSC5 (Yale Bright Star Catalog 5th Edition). The center of the view axis points to the celestial equator (0 degrees of declination, 0 to 360 degrees of right ascension), which has a large variation in star density, and the algorithm's performance can be verified in different star density scenarios, with the number of stars in the field of view corresponding to the different right ascensions shown in Figure 8 of [42]. Figure 7 shows typical synthetic images for presentation.

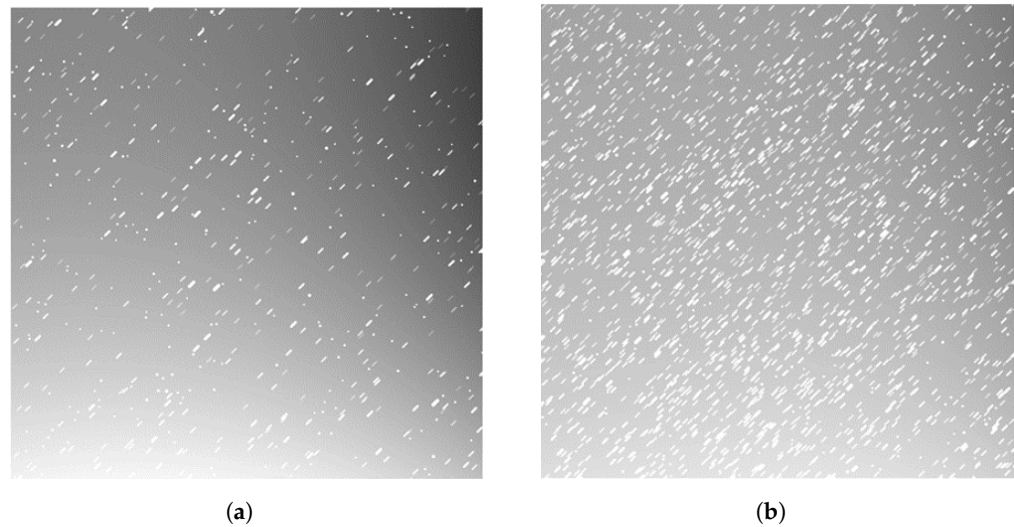


Figure 7. Typical synthetic star background image. (a) Synthetic image of a sparsely distributed star sky region (visual axis pointing: 0° declination, 0° right ascension). (b) Synthetic image of a densely distributed star sky region (visual axis pointing: 290° declination, 0° right ascension).

3.2.2. Measured Data

The measured data parameters are as follows: field of view, 2.5 degrees; image size, $1\text{ k} \times 1\text{ k}$; image bit depth, 16 bits; limiting magnitude about 13 Mv; and a total of 338,577 measured images included in 523 tasks over 31 days. The imaging environment of the measured data is more complex to better verify the comprehensive performance of the algorithm, especially the real complex background simulation algorithm is difficult to simulate realistically, so the measured images can be a good test of the algorithm's performance of the star map matching in the complex background.

3.3. Introduction to Evaluation Metrics

In this paper, matching rate, matching accuracy, and running time are used to evaluate the robustness, accuracy, and real-time performance of the star map matching algorithms.

Matching rate: This metric is defined as the ratio of the number of successfully matched image frames to the total number of image frames.

Matching accuracy: The deviation of the transformed position of the star image from the reference true value position. For the synthetic image, truth positions are given in the simulation, and the measured image truth is the celestial sphere coordinates of the star in the catalogs. It should be noted that the exact position of the star is rounded in the synthetic data, which results in a rounding error that is uniformly distributed over the interval $[-0.5, 0.5]$ with a mean value of about 0.38 pixels (for the synthetic data in this paper it is about 3.36 arcseconds), as shown in Equation (26).

$$E_s = \int_{-0.5}^{0.5} \int_{-0.5}^{0.5} \sqrt{x^2 + y^2} dx dy \approx 0.38(\text{pixels}). \quad (26)$$

3.4. Test Results

3.4.1. Synthetic Data Visual Axis Pointing Deviation Test Results

The purpose of this test is to test the performance of the star map matching algorithm under different sky regions and different visual axis pointing deviations. Specific configuration: visual axis pointing deviation 0 to 1 degree, step size 0.03 degrees; each deviation angle test 360 images (reference center pointing: 0 degrees of declination, 0 to 360 degrees of right ascension, step size 5 degrees, select 5 points as the center of the image to be tested in the concentric circle centered on the reference center pointing), a total of 12,240 images; do not add the noise, magnitude deviation and positioning deviation, and the rest of the

parameters for the default values. Figures 8–10 show the statistics of the matching rate, average matching error, and average running time of each algorithm in this test scenario with different visual axis pointing deviations.

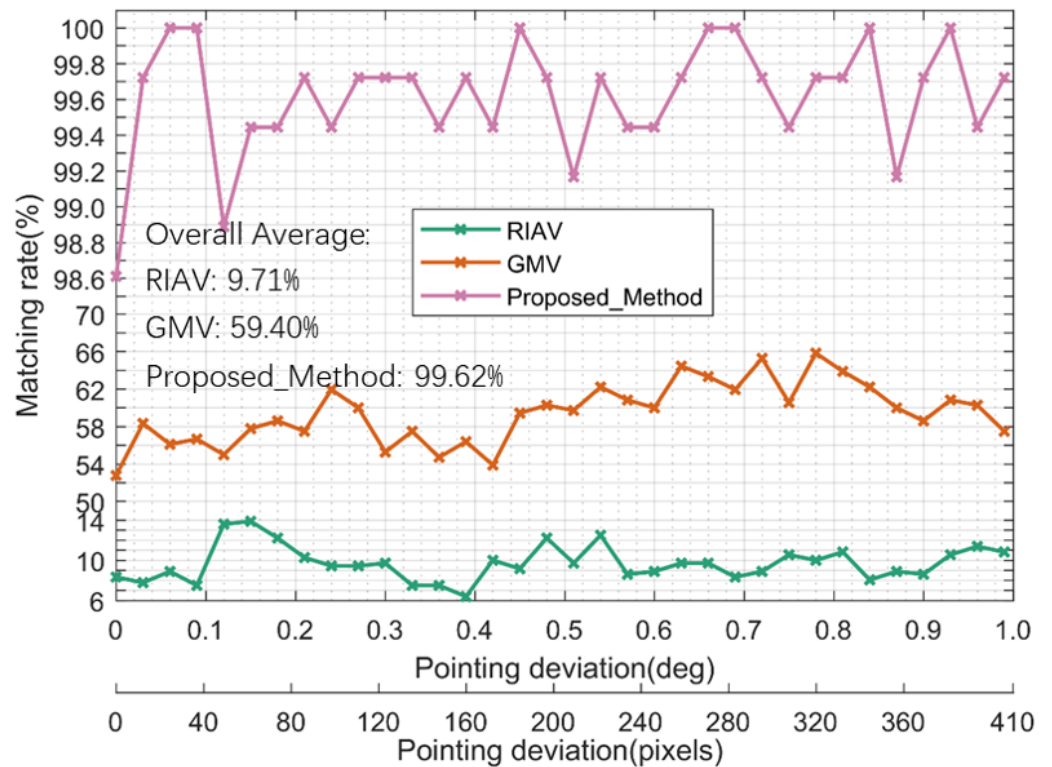


Figure 8. Matching rates for different visual axis pointing deviations.

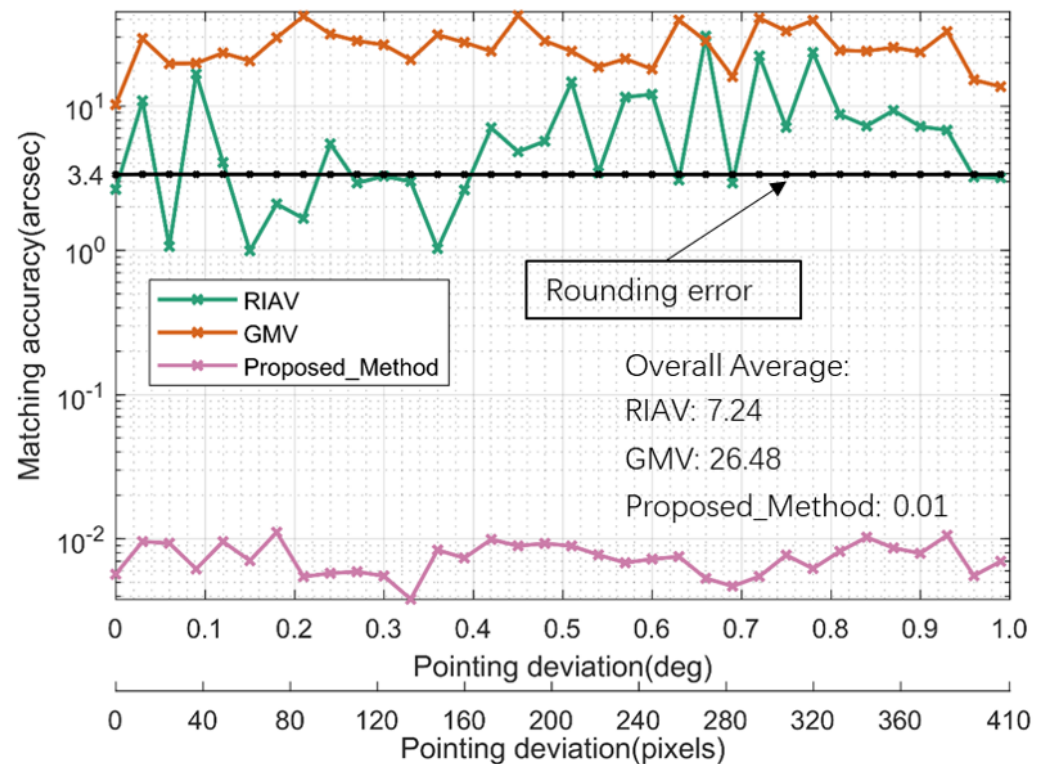


Figure 9. Average matching accuracy for different visual axis pointing deviations (black line is the error introduced by the synthetic data).

According to the test results, overall the proposed algorithm performs significantly better than the RIAV and GMV algorithms. Specifically for the matching rate, the proposed algorithm maintains a matching rate of over 98.6%, indicating that it is robust to visual axis pointing deviation; the matching rate of GMV in the comparison algorithms fluctuates between 52% and 66%, indicating that the GMV algorithm is insensitive to visual axis pointing deviation but is not robust; the worst matching rate of the RIAV algorithm is below 14%, indicating that the algorithm is very poor in robustness. In terms of matching accuracy, the proposed algorithm maintains a high matching accuracy (overall average error is only 0.01 arcseconds); RIAV and GMV are not very accurate and fluctuate a lot, especially the GMV algorithm has an overall average error of 26.48 arcseconds. In terms of running time, since the RIAV algorithm only tests 20% of the total number of stars its real-time performance is optimal (the rest of the test scenarios are in the same situation, and the running time of the RIAV algorithm will not be explained subsequently); the GMV algorithm takes the longest time at about 2.5 s, and the overall average running time of the proposed algorithm is 0.13 s, which is a very good real-time performance.

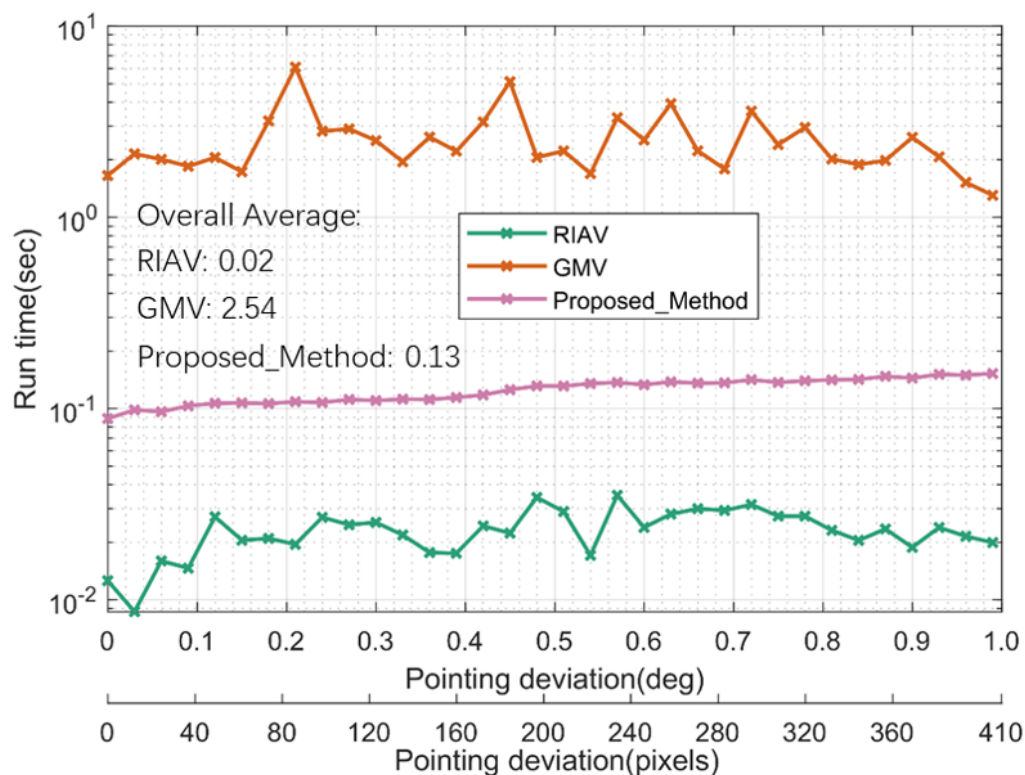


Figure 10. Average running time for different visual axis pointing deviations.

3.4.2. Synthetic Data False Star Test Results

The purpose of this test is to test the performance of the star map matching algorithm in different sky regions and with different numbers of false stars. Here, false stars refer to noise and bad points, which are simulated by adding patchy noise (salt-and-pepper noise after Gaussian blurring) and highlighted individual pixels (salt-and-pepper noise). The specific configuration is as follows: the proportion of patchy noise to the total pixels is from 0 to 5×10^{-4} , with a step size of 1.613×10^{-5} ; the proportion of highlighted independent pixels to the total pixels is from 0 to 5×10^{-5} , with a step size of 1.613×10^{-6} ; i.e., the proportion of total false stars to the total pixels is from 0 to 5.5×10^{-4} , with a step size of 1.774×10^{-5} ; each sub-scene was tested with 360 images (center pointing: 0 degrees of declination, 0 to 359 degrees of right ascension, 1 degree step), a total of 11,160 images; no magnitude deviation and positioning deviation were added, and the rest of the parameters were default values. Figures 11–13 show the statistics of the matching rate, average matching

error, and average running time of each algorithm in this test scenario when the proportion of false stars in the total pixels varies; where the horizontal axis percentage of false stars is the ratio of the number of false stars to the total number of stars (the sum of the number of real stars and false stars).

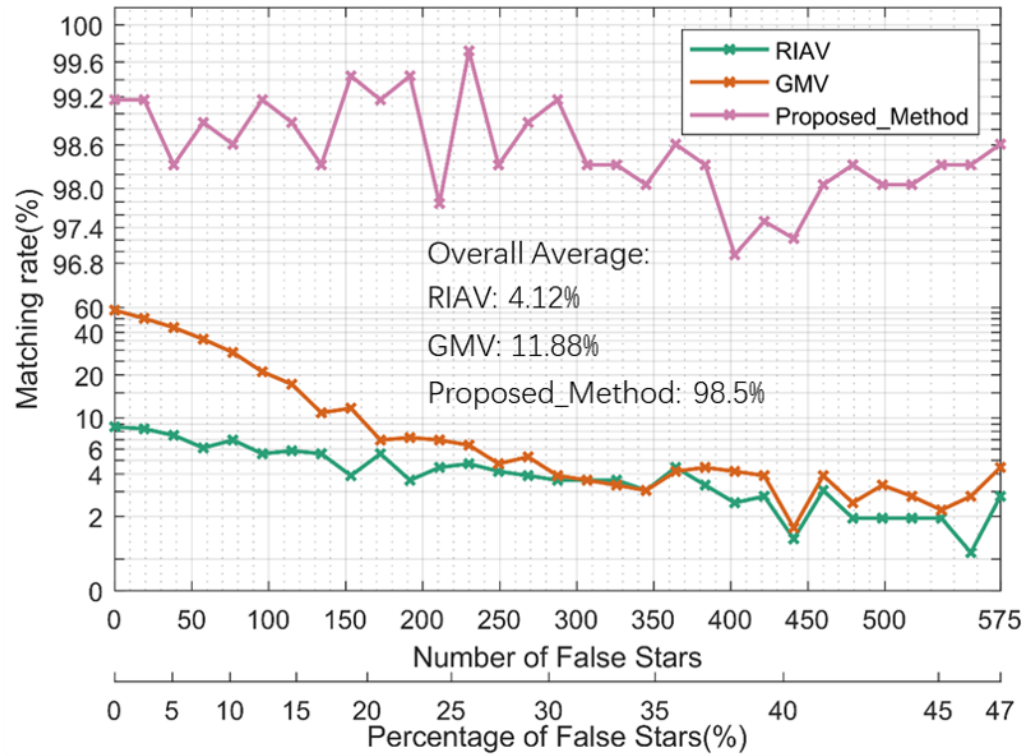


Figure 11. Matching rates for different numbers of false stars.

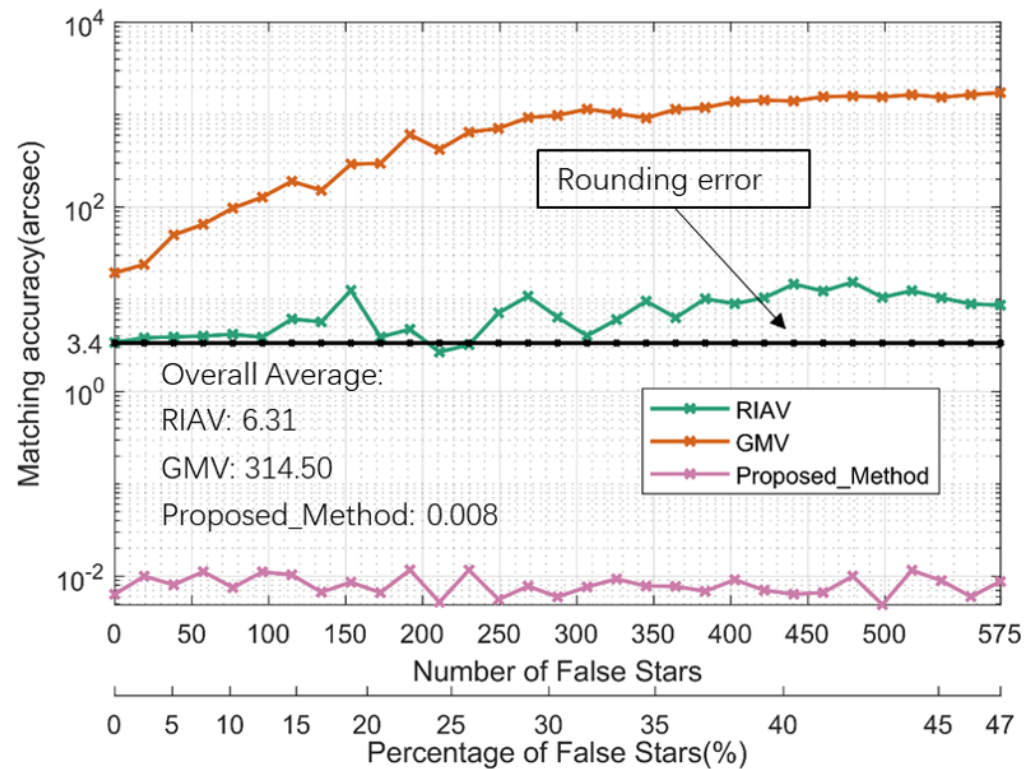


Figure 12. Average matching accuracy for different numbers of false stars (black line is the error introduced by the synthetic data).

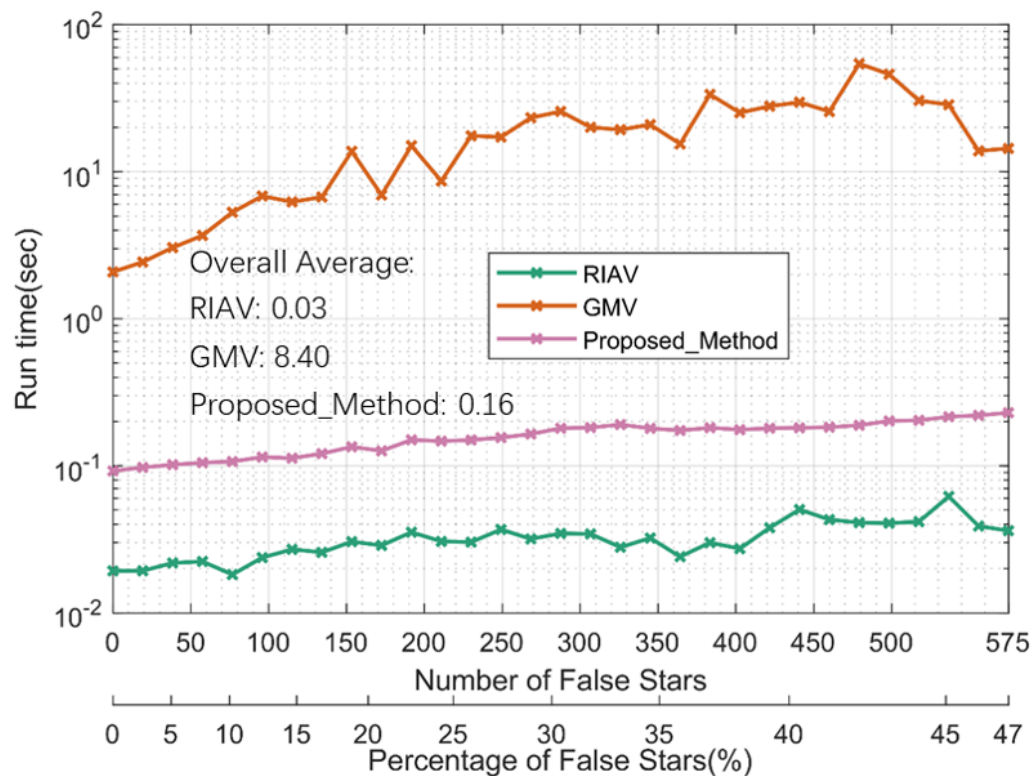


Figure 13. Average running time for different numbers of false stars.

According to the test results, overall the proposed algorithm performs significantly better than the RIAV and GMV algorithms. Specifically, the proposed algorithm maintains a matching rate of more than 96.8% with only slight fluctuations, indicating that the algorithm is robust to the number of false stars; its average matching error is the smallest and hardly fluctuates with the number of false stars, and its average running time is around 0.16 s. The matching rate, matching accuracy, and running time of the GMV algorithm deteriorate with the increase in the number of false stars, indicating that the algorithm is more sensitive to false stars. The RIAV algorithm has the worst overall performance and its performance is also sensitive to false stars.

3.4.3. Synthetic Data Positioning Deviation Test Results

The purpose of this test is to test the performance of the star map matching algorithm in different sky regions and with different positioning deviations. The process is simulated by adding Gaussian noise with a mean value of 0 and a standard deviation of σ pixels in the row and column directions of the stellar coordinates, respectively. Specific configurations: $0 \leq \sigma \leq 1.5$, with a step size of 0.1 pixels, 360 images are tested for each value of σ (center pointing: 0 degrees of declination, 0 to 359 degrees of right ascension, with a step size of 1 degree), a total of 5760 images; no noise and magnitude deviation are added, and the rest of the parameters are default values. Figures 14–16 show the statistics of the matching rate, average matching error, and average running time of each algorithm in this test scenario with different positioning deviations.

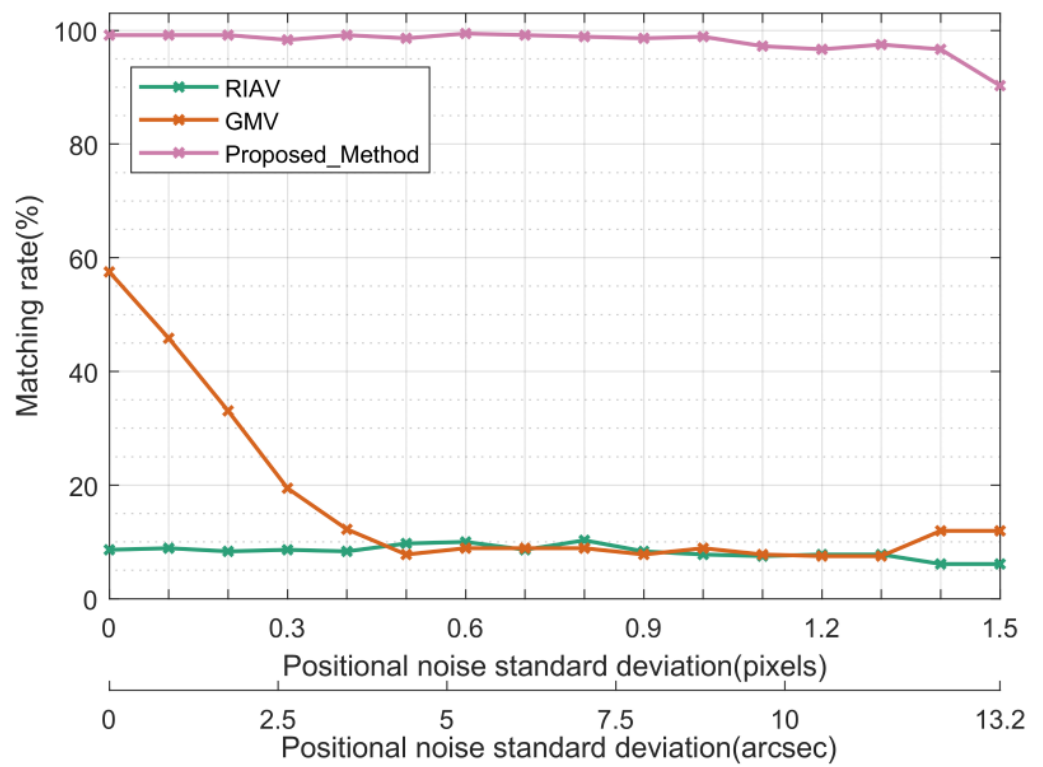


Figure 14. Matching rate for different positioning deviations.

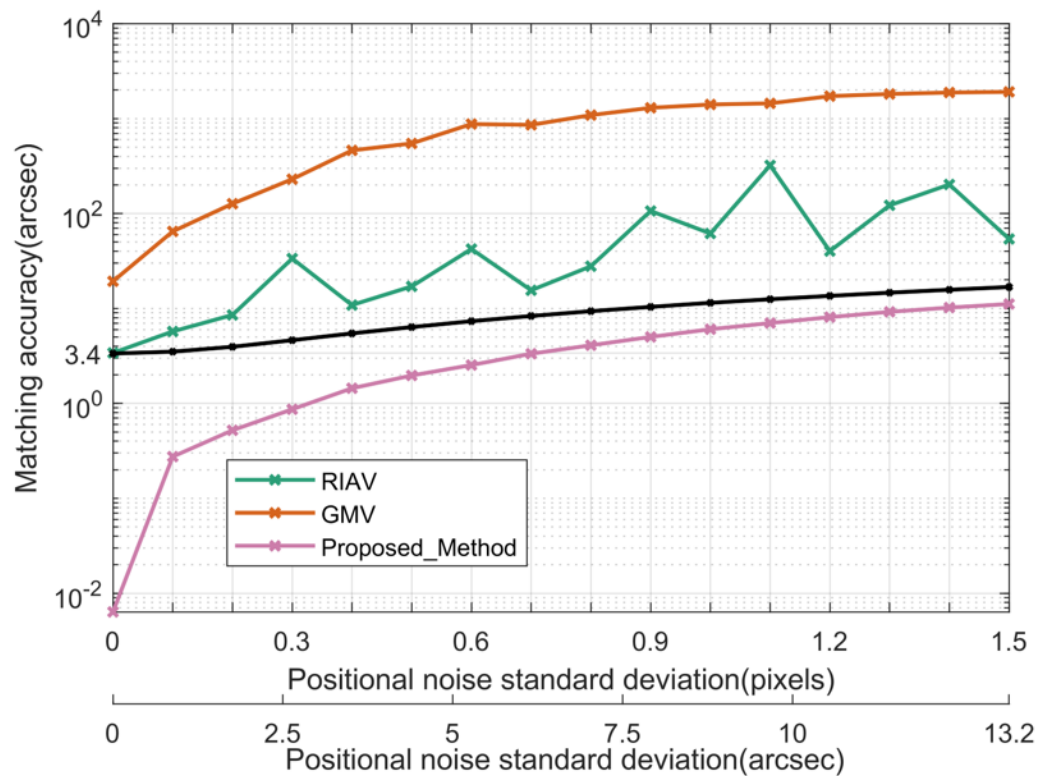


Figure 15. Average matching accuracy for different positioning deviations (black line is the error introduced by the synthetic data).

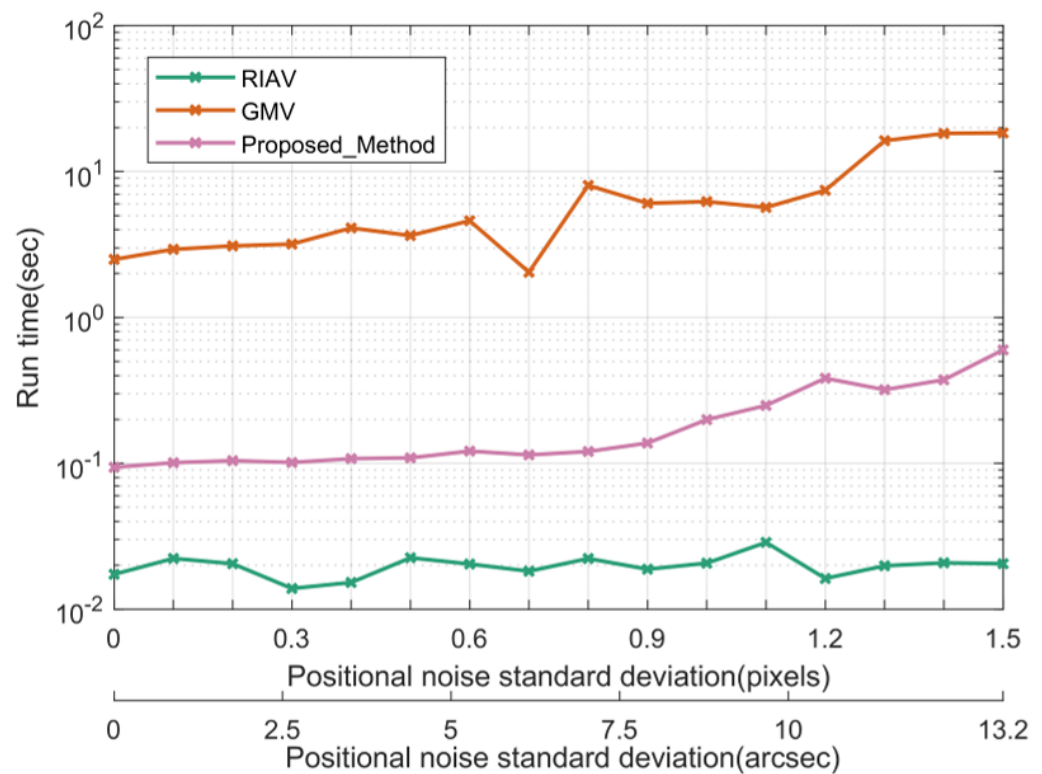


Figure 16. Average running time for different positioning deviations.

According to the test results, overall the proposed algorithm has the best performance, and the RIAV and GMV algorithms are worse. Specifically, for the matching rate, the proposed algorithm basically stays above 95%, and only decreases when the positioning deviation is 1.5 pixels, which indicates that it is robust to positioning deviation; the matching rate of the GMV algorithm in the comparison algorithms decreases drastically with the increase in the deviation, which indicates that it is sensitive to the positioning deviation; and the matching rate of the RIAV algorithm is the worst, around 10%, which indicates that the robustness of the algorithm is very poor. In terms of matching accuracy, the proposed algorithm can eliminate the outliers with large errors as soon as possible, and its performance is optimal, and the overall error is lower than that introduced by the synthetic data; the accuracy of RIAV and GMV is not high and fluctuates a lot, and the total average matching error is more than 50 arcseconds. In terms of running time, the GMV algorithm takes the longest time at about 5.3 s, and the proposed algorithm has an average running time of about 0.2 s for a single frame, which meets the task requirements.

3.4.4. Synthetic Data Magnitude Deviation Test Results

The purpose of this test is to test the performance of the star map matching algorithm in different sky regions and with different magnitude deviations. The process is simulated by adding Gaussian noise with mean value 0 and standard deviation σ Mv to the actual magnitude values. Specific configurations: $0 \leq \sigma \leq 1.5$, with a step size of 0.05 Mv; 360 images are tested for each value of σ (center pointing: 0 degrees of declination, 0 to 359 degrees of declination, with a step size of 1 degree), for a total of 11,160 images; no noise and positioning deviation are added, and the rest of the parameters are default values. Figures 17–19 show the statistics of the matching rate, average matching error, and average running time of each algorithm in this test scenario with different magnitude deviations.

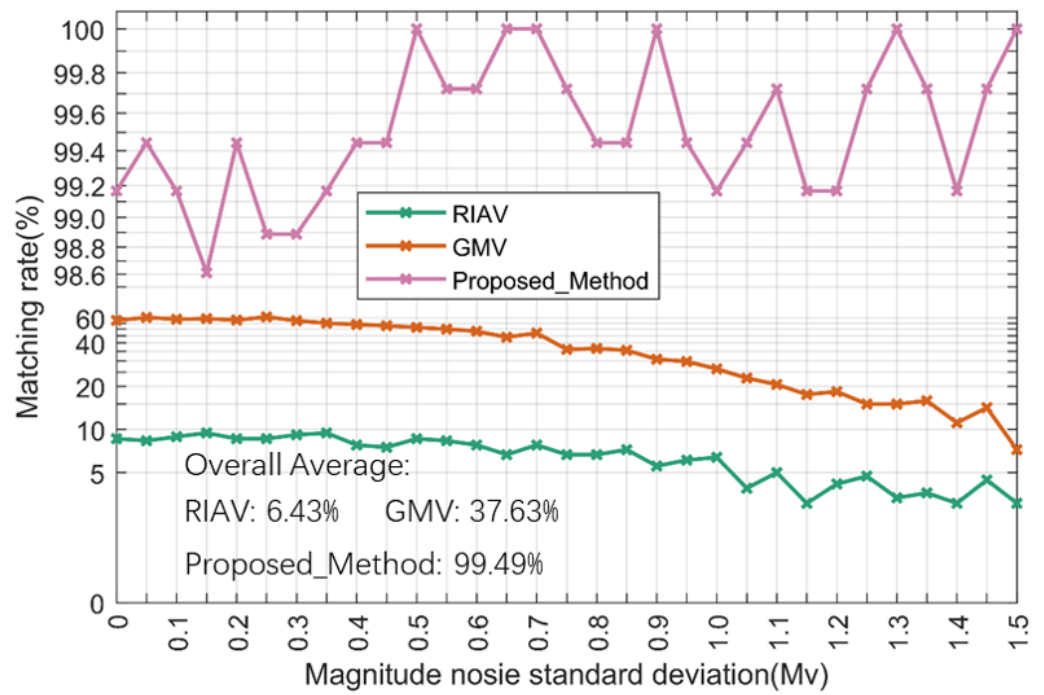


Figure 17. Matching rate for different magnitude deviations.

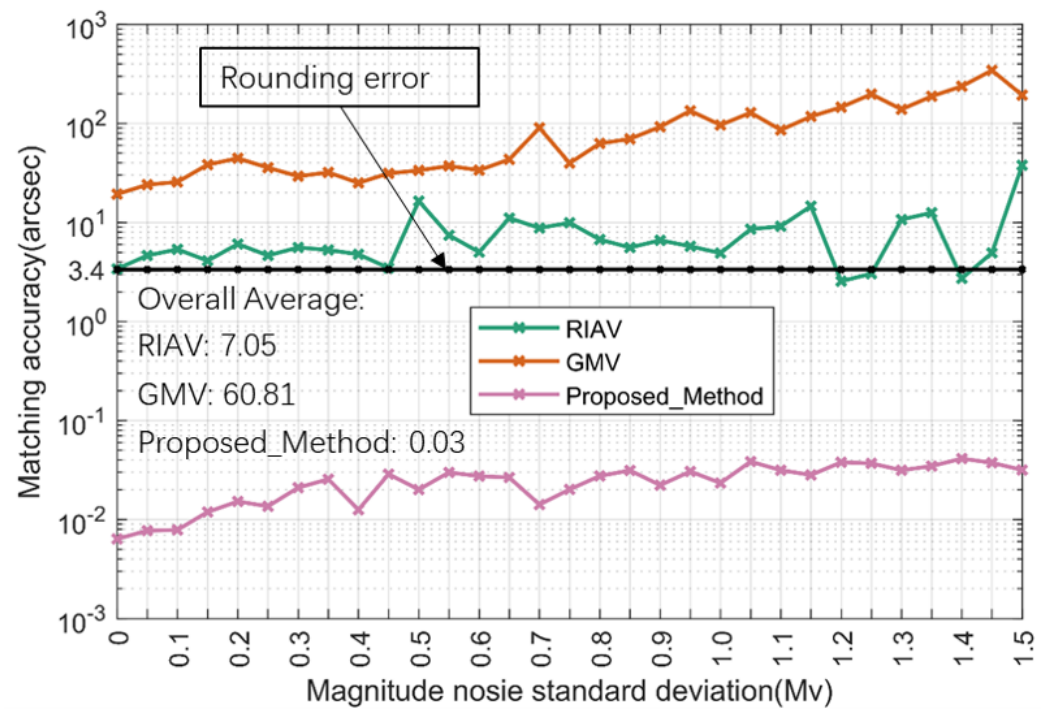


Figure 18. Average matching accuracy for different magnitude deviations (black line is the error introduced by the synthetic data).

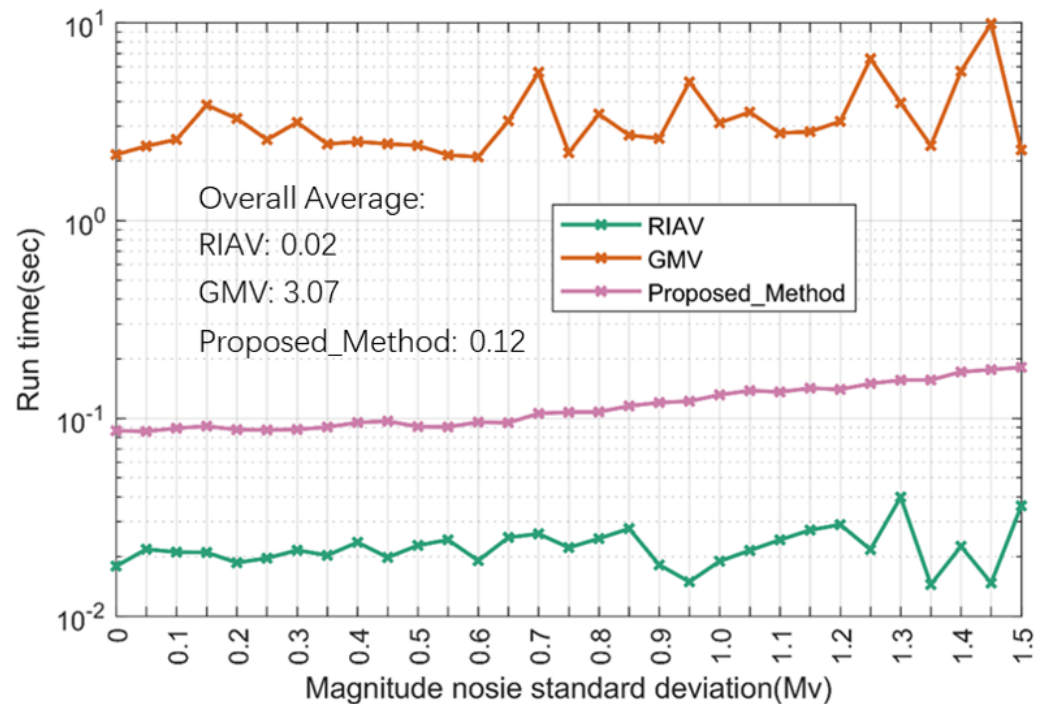


Figure 19. Average running time for different magnitude deviations.

According to the test results, the proposed algorithm has a better matching rate, matching accuracy, and runtime performance with little fluctuation, indicating that it is insensitive to the magnitude deviation; the GMV algorithm basically stabilizes the matching rate at about 60%, with little fluctuation in the matching error and runtime when the magnitude deviation is less than 0.5 Mv, but the performance gradually deteriorates with the increase in the magnitude deviation; and the RIAV algorithm still performs the worst and is sensitive to the magnitude deviation.

3.4.5. Measured Data Test Results

The performance of the star map matching algorithms on measured images of dense stars with complex backgrounds was tested using measured data. Figures 20–22 show the statistics of the matching rate, average matching error, and average running time of each algorithm for different tasks when tested using measured data.

According to the test results, overall the proposed algorithm outperforms the RIAV and GMV algorithms. Specifically, the proposed algorithm maintains a matching rate of 100% for most of the tasks, and only a few tasks have a slightly lower matching rate due to poor image quality and a large deviation in the visual axis pointing, indicating that it is suitable for almost all scenes in the measured images; the GMV algorithm in the comparison algorithm has a slightly lower matching rate than the proposed algorithm, maintaining a matching rate of more than 94%, which reflects the high robustness characteristics of the GMV algorithm; the RIAV algorithm has the worst matching rate and fluctuates greatly, indicating that the algorithm cannot adapt to most of the scenes in the measured images. In terms of matching accuracy, the proposed algorithm maintains a matching error of 0.5 to 1 arcseconds for most of the tasks, which is significantly better than the comparison algorithms; the errors of both RIAV and GMV fluctuate greatly between 1 and 15 arcseconds. In terms of running time, the proposed algorithm maintains an overall time of about 0.03 s, which fully meets the real-time requirements; the GMV algorithm is still the most time consuming, with a total average time of 0.92 s.

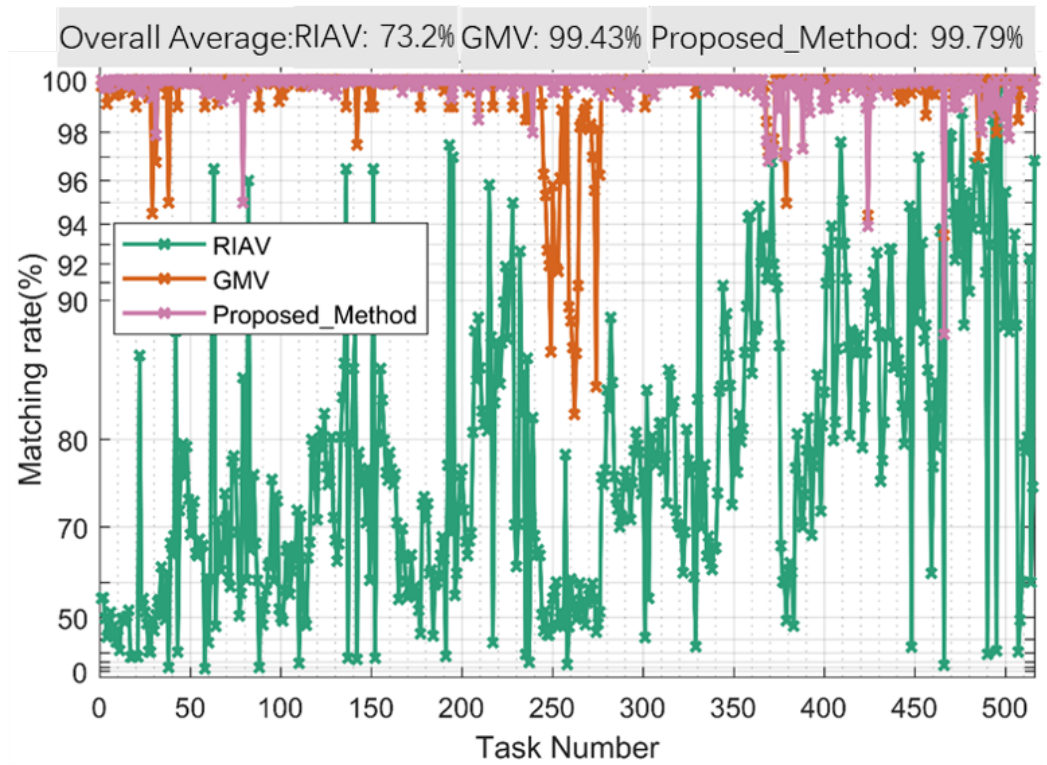


Figure 20. Matching rate for different tasks.

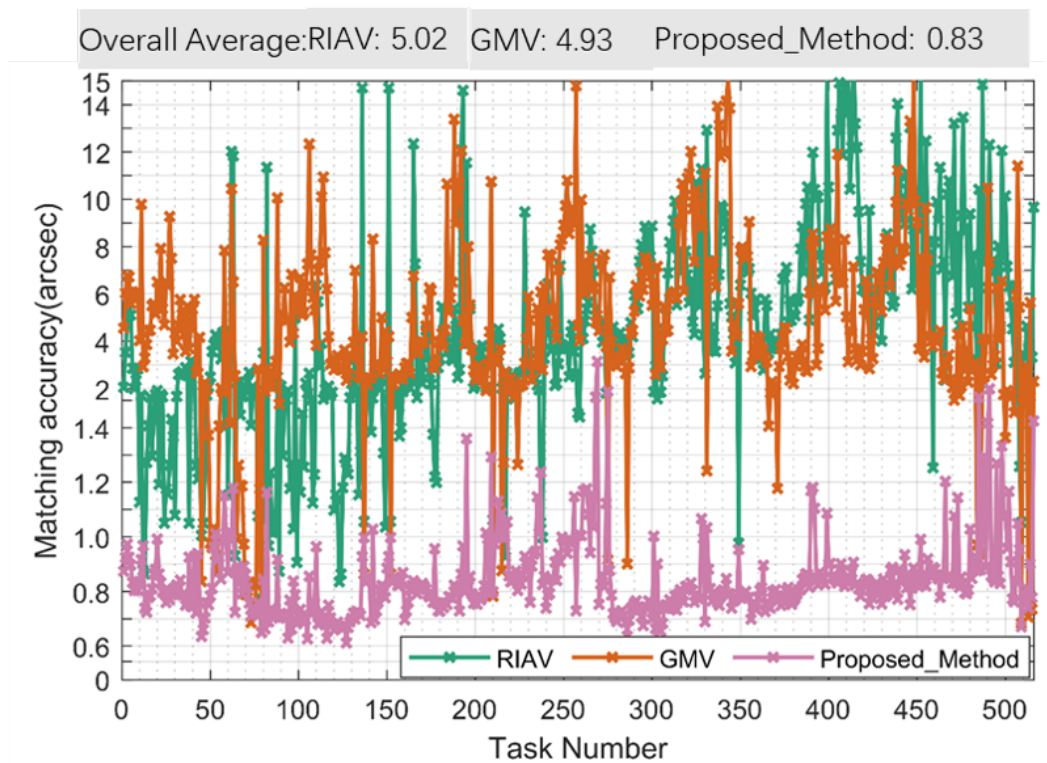


Figure 21. Average matching accuracy for different tasks.

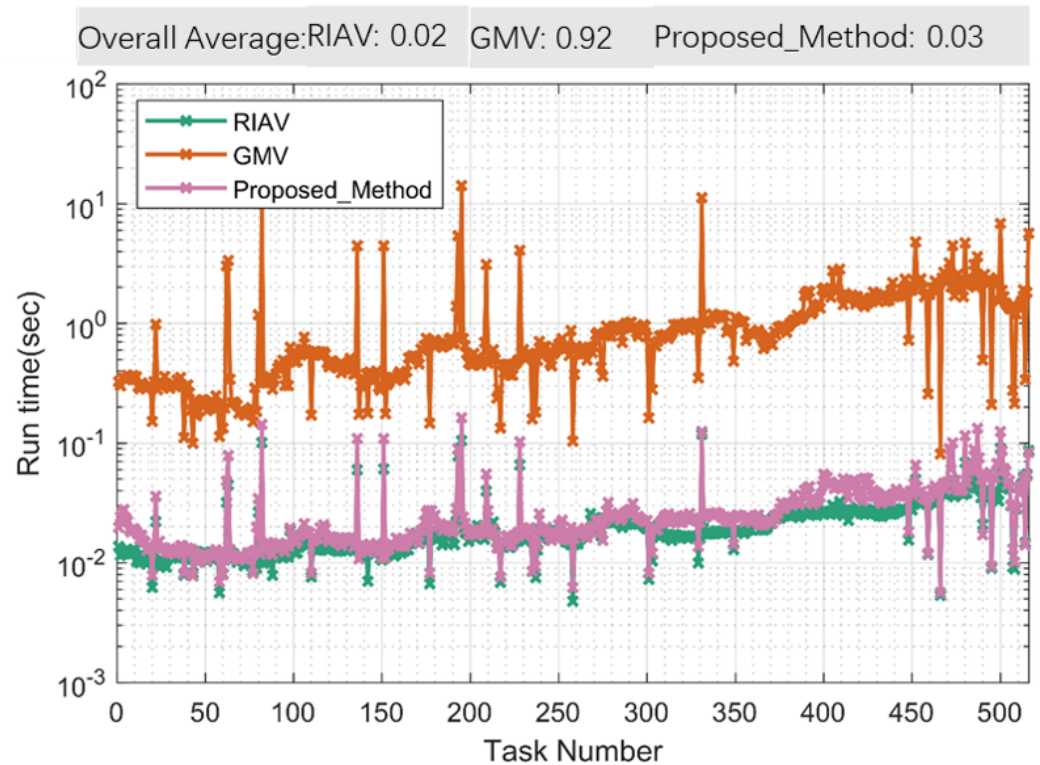


Figure 22. Average running time for different tasks.

For a more in-depth analysis of the performance of the star map matching algorithms, the matching errors of each algorithm are plotted in Figures 23–25, with the horizontal and vertical axes showing the errors in the right ascension ($\Delta\alpha\cos\delta$) and declination ($\Delta\delta$) directions, respectively. Each matched star corresponds to an error point on the figure, the frequency distributions of the errors in the right ascension and declination directions are given on the lower and right sides of the figure, respectively, and the statistical results are given in the upper right corner. “Num” is the total number of matched stars, which reflects the robustness of the algorithm, and the larger the value is, the better the algorithm is; “mean $\Delta\angle$ ” is the mean value of the error angle of all the star images (error angle: the angle between the celestial sphere coordinates based on the transformation parameters provided by the matching algorithm and the reference coordinates of the same star in the catalog; it should be noted that this is different from the mean value of the error calculated according to the number of images in Figure 21); the smaller the value is, the higher the algorithm’s accuracy is; “mean $\Delta\alpha\cos\delta$ ” and “mean $\Delta\delta$ ” denote the mean values of the errors in the right ascension and declination directions, respectively, and “std $\Delta\alpha\cos\delta$ ” and “std $\Delta\delta$ ” denote the standard deviation of the errors in the right ascension and declination directions, respectively. Ideally, the right ascension and declination errors should be normally distributed, so the mean values of the right ascension and declination errors should be close to 0, and the smaller the standard deviation is, the better the performance of the algorithm is.

The error statistics show that the total number of matched stars of the proposed algorithms is significantly more than that of the comparison algorithms, and the average error angle of 0.77 arcsec (equivalent to about 0.1 pixel) is the smallest, which indicates that the proposed algorithms are optimal in terms of robustness and accuracy. In terms of error distribution, the mean error values of all the algorithms are basically around 0, but the standard deviations of the errors in the right ascension and declination directions of the proposed algorithm are significantly smaller than those of the comparison algorithm, which indicate that the errors of the proposed algorithm in both the right ascension and declination directions are more concentrated near 0.

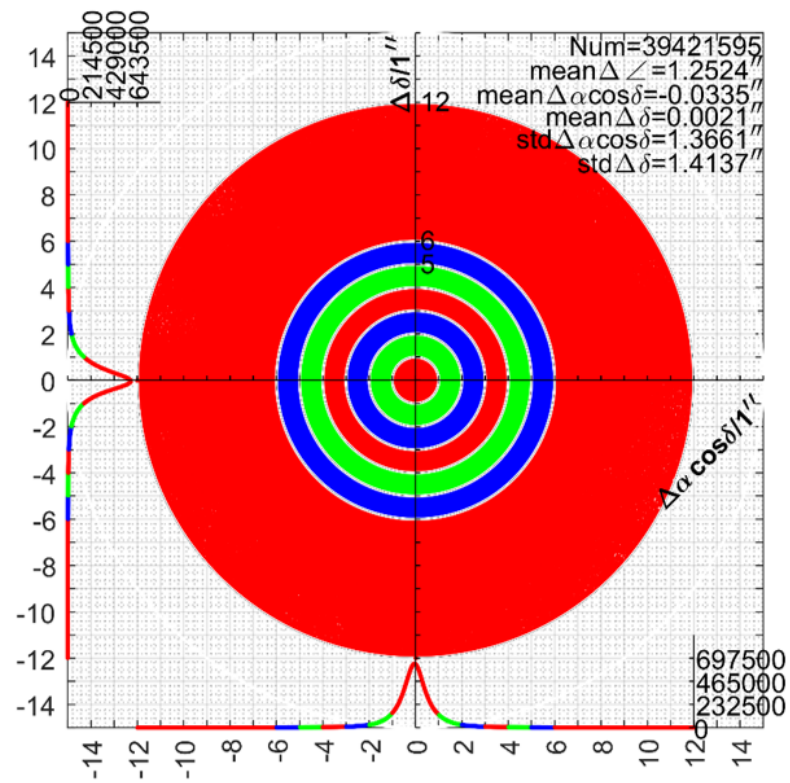


Figure 23. Matching error of RIAV.

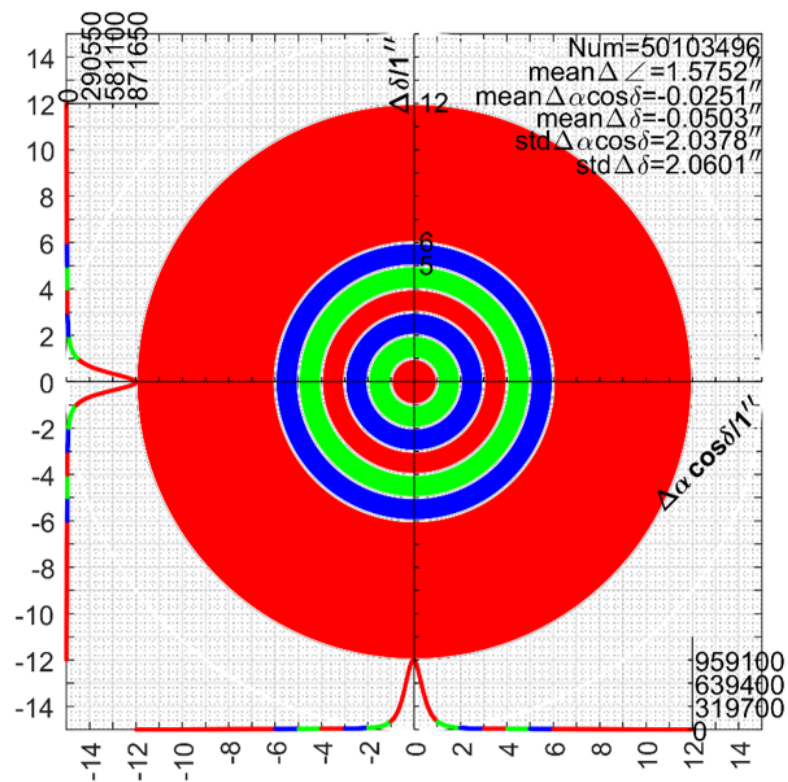


Figure 24. Matching error of GMV.

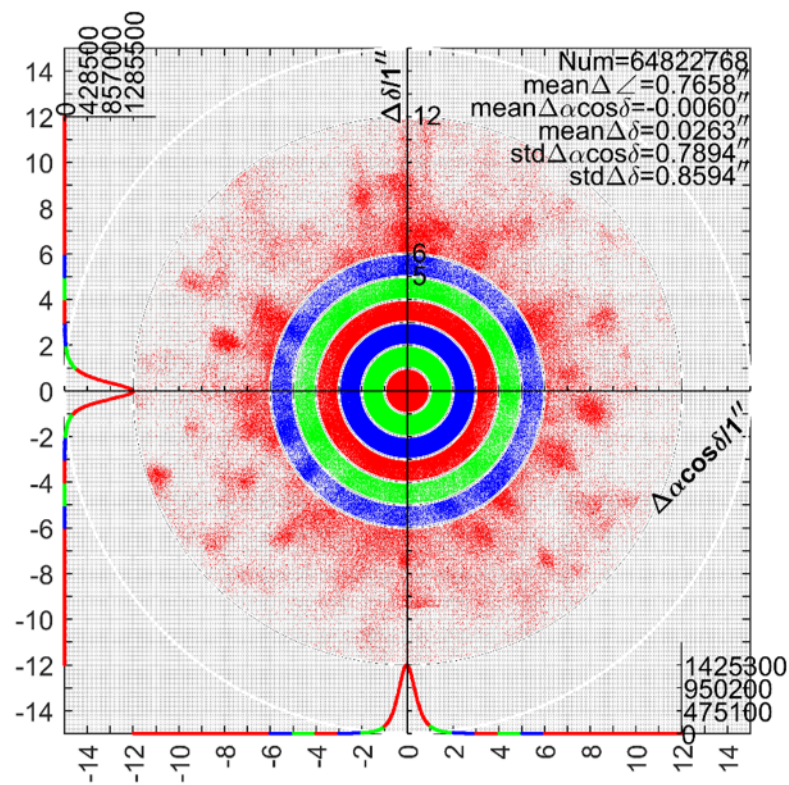


Figure 25. Matching error of the proposed algorithm.

4. Discussion

In this section, the comprehensive performance of the algorithms proposed in this paper and the comparison algorithms are discussed. In order to evaluate the comprehensive performance of the star map recognition algorithms, a scoring function of the following equation is defined in reference [42]:

$$S = \frac{1}{N} \sum_{i=1}^N S_i^s (w_a * S_i^a + w_t * S_i^t), \quad (27)$$

where N is the number of matched image frames, S_i^s , S_i^a , and S_i^t are the performance scores of matching rate, accuracy, and running time, respectively; $w_a = 0.8$ and $w_t = 0.2$ are the weights of accuracy and running time, respectively, considering that the accuracy is more reflective of the core performance of the algorithm than the running time in the practical application, the weight of the accuracy is bigger here. S_i^s , S_i^a , and S_i^t are defined as follows:

$$S_i^s = \begin{cases} 1 & \text{Success} \\ 0 & \text{Fall} \end{cases}, \quad (28)$$

$$S_i^a = 100 \times \max\left(0, 1 - \frac{\max(0, a_i - Ra_i)}{2 \times \max(f, Ra_i)}\right), \quad (29)$$

where a_i is the accuracy of the algorithm, Ra_i is the reference accuracy (for synthetic data it is the introduced error, measured data is 0), f is the adjustment factor (3.4 arcseconds).

$$S_i^t = \begin{cases} 100 & 0 \leq t_i \leq 0.2s \\ 90 & 0.2 < t_i \leq 0.5s \\ 70 & 0.5 < t_i \leq 1s \\ 50 & 1 < t_i \leq 3s \\ 30 & t_i > 3s \end{cases}, \quad (30)$$

where t_i is the running time of the algorithm.

Table 2 shows the statistics of the comprehensive performance of the algorithm, where “score” is the average score of all test scenarios to indicate the comprehensive performance of the algorithm. The radar chart of the comprehensive performance of the star map matching algorithm is shown in Figure 26.

Table 2. Comprehensive performance statistics of star map matching algorithms.

	Pointing Deviation	False Star	Positioning Deviation	Magnitude Deviation	Measured Data	Score
RIAV	8.46	3.33	6.48	5.61	55.48	15.87
GMV	38.43	3.75	5.08	19.18	68.17	26.92
Proposed Method	98.97	97.75	96.39	98.91	90.08	96.42

According to the comprehensive performance evaluation results, the proposed algorithm has significantly better comprehensive performance than the comparison algorithms, and its performance is more balanced in each scene. Among the comparison algorithms, the GMV algorithm performs reasonably well in the measured data, but its performance is obviously insufficient in the synthetic data scenario not covered by the measured data, indicating that the GMV algorithm is less suitable for the dense star scenario; the RIAV algorithm has the worst performance overall, and according to the test results, this algorithm is not suitable for star map matching in the dense star scenario.

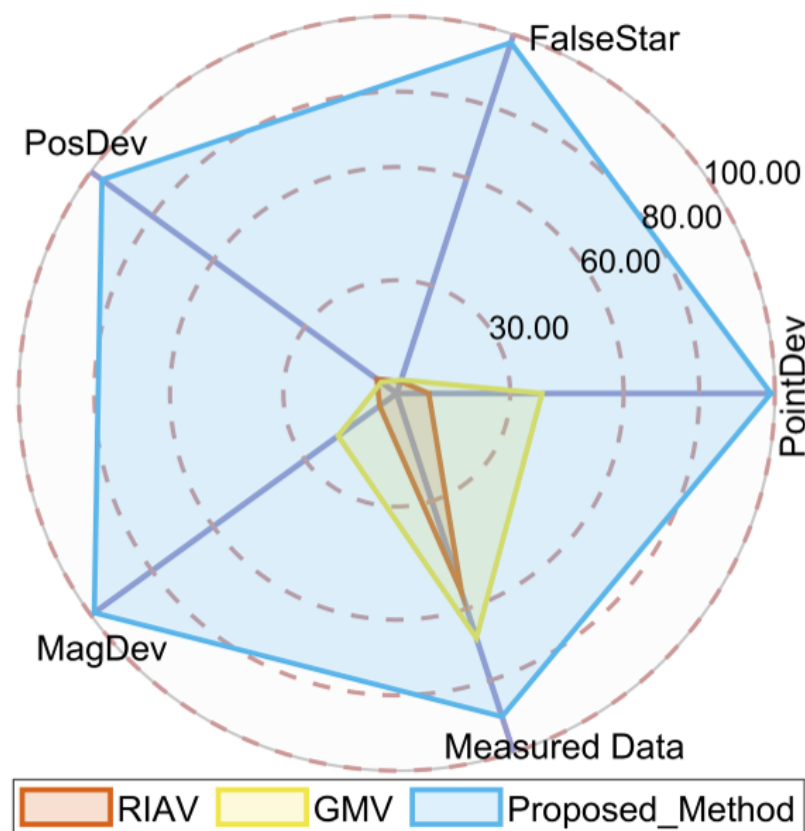


Figure 26. Comprehensive performance radar charts for star map matching algorithms.

5. Conclusions

In this paper, we propose a robust star map matching algorithm for dense star scenes, which is derived from the geometric voting star map matching technique, relies on angular features, includes two steps of rough matching and fine matching, and has the characteristics of high accuracy, robustness, and good real-time performance. The synthetic data test results show that the proposed algorithm is robust to stellar positioning error, false stars, visual axis pointing deviation, and magnitude uncertainty in a dense star application scenario. Its matching rate and matching accuracy are significantly better than the comparison algorithm, and the average running time fully meets the task requirements. The excellent performance of the algorithm is further verified by the results of processing a large amount of real measurement data. Overall its comprehensive performance score reaches 96.42, indicating that it is a relatively ideal star map matching algorithm for dense star scenes. Finally, we present future related work. Given that the matching rate of the proposed algorithm has not yet reached 100%, further research on more robust improvement algorithms will be carried out; in addition, the development of a dense star scene star map matching algorithm that does not rely on pointing information will also be a part of future work.

Author Contributions: Conceptualization, Q.S.; methodology, Q.S.; software, Q.S.; validation, Q.S.; formal analysis, Q.S.; investigation, Q.S.; resources, Z.N.; data curation, Y.L.; writing—original draft preparation, Q.S.; writing—review and editing, Z.N. and Z.W.; visualization, Q.S.; supervision, Z.N.; project administration, Z.N.; funding acquisition, Z.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The synthetic data supporting the results of this study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors thank the Xi'an Satellite Control Center for providing real data to fully validate the performance of the algorithm, and are very grateful to all the editors and reviewers for their hard work on this paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Christian, J.A.; Crassidis, J.L. Star Identification and Attitude Determination with Projective Cameras. *IEEE Access* **2021**, *9*, 25768–25794. [[CrossRef](#)]
2. Liu, H.; Wei, X.; Li, J.; Wang, G. A star identification algorithm based on simplest general subgraph. *Acta Astronaut.* **2021**, *183*, 11–22. [[CrossRef](#)]
3. Zhang, G. *Star Identification: Methods, Techniques and Algorithms*; Springer: Berlin/Heidelberg, Germany; New York, NY, USA, 2017; Chapter 1; pp. 24–32. [[CrossRef](#)]
4. Cole, C.L.; Crassidis, J.L. Fast Star-Pattern Recognition Using Planar Triangles. *J. Guid. Control. Dyn.* **2006**, *29*, 64–71. [[CrossRef](#)]
5. Somayeh, F.; Nikkhah, A.A.; Roshanian, J.; Salahshoor, S. Blind Star Identification Algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 547–557. [[CrossRef](#)]
6. Liu, M.; Wei, X.; Wen, D.; Wang, H. Star Identification Based on Multilayer Voting Algorithm for Star Sensors. *Sensors* **2021**, *21*, 3084. [[CrossRef](#)] [[PubMed](#)]
7. Zhang, G.J.; Wei, X.G.; Jiang, J. Star Map Identification Based on a Modified Triangle Algorithm. *Acta Aeronaut. Astronaut. Sin.* **2006**, *27*, 1150–1154.
8. Zhao, Y.; Wei, X.; Li, J.; Wang, G. Star Identification Algorithm Based on K–L Transformation and Star Walk Formation. *IEEE Sens. J.* **2016**, *16*, 5202–5210. [[CrossRef](#)]
9. Wang, G.; Li, J.; Wei, X. Star Identification Based on Hash Map. *IEEE Sens. J.* **2018**, *18*, 1591–1599. [[CrossRef](#)]
10. Leake, C.; Arnas, D.; Mortari, D. Non-Dimensional Star-Identification. *Sensors* **2020**, *20*, 2697. [[CrossRef](#)]
11. Mortari, D. Search-Less Algorithm for Star Pattern Recognition. *J. Astronaut. Sci.* **1997**, *45*, 179–194. [[CrossRef](#)]
12. Hernández, E.A.; Alonso, M.A.; Chávez, E.; Covarrubias, D.H.; Conte, R. Robust polygon recognition method with similarity invariants applied to star identification. *Adv. Space Res.* **2017**, *59*, 1095–1111. [[CrossRef](#)]
13. Toloei, A.; Zahednamazi, M.; Ghasemi, R.; Mohammadi, F. A comparative analysis of star identification algorithms. *Astrophys. Space Sci.* **2020**, *365*, 63. [[CrossRef](#)]

14. Niu, Y.; Wei, X.; Li, J. Fast and Robust Star Identification Using Color Ratio Information. *IEEE Sens. J.* **2022**, *22*, 20401–20412. [[CrossRef](#)]
15. Yuan, X.; Zhu, J.; Zhu, K.; Li, X. A Star-Identification Algorithm Based on Global Multi-Triangle Voting. *Appl. Sci.* **2022**, *12*, 9993. [[CrossRef](#)]
16. Spratling, B.B.; Mortari, D. A Survey on Star Identification Algorithms. *Algorithms* **2009**, *2*, 93–107. [[CrossRef](#)]
17. Liebe, C. Pattern recognition of star constellations for spacecraft applications. *IEEE Aerosp. Electron. Syst. Mag.* **1993**, *8*, 31–39. [[CrossRef](#)]
18. Liebe, C. Star trackers for attitude determination. *IEEE Aerosp. Electron. Syst. Mag.* **1995**, *10*, 10–16. [[CrossRef](#)]
19. Sun, L.; Jiang, J.; Zhang, G.; Wei, X. A Discrete HMM-Based Feature Sequence Model Approach for Star Identification. *IEEE Sens. J.* **2016**, *16*, 931–940. [[CrossRef](#)]
20. Padgett, C.; Kreutz-Delgado, K. A grid algorithm for autonomous star identification. *IEEE Trans. Aerosp. Electron. Syst.* **1997**, *33*, 202–213. [[CrossRef](#)]
21. Clouse, D.; Padgett, C. Small field-of-view star identification using Bayesian decision theory. *IEEE Trans. Aerosp. Electron. Syst.* **2000**, *36*, 773–783. [[CrossRef](#)]
22. Silani, E.; Lovera, M. Star identification algorithms: Novel approach & comparison study. *IEEE Trans. Aerosp. Electron. Syst.* **2006**, *42*, 1275–1288. [[CrossRef](#)]
23. Aghaei, M.; Moghaddam, H.A. Grid star identification improvement using optimization approaches. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2080–2090. [[CrossRef](#)]
24. Li, J.; Wei, X.; Wang, G.; Zhou, S. Improved Grid Algorithm Based on Star Pair Pattern and Two-dimensional Angular Distances for Full-Sky Star Identification. *IEEE Access* **2020**, *8*, 1010–1020. [[CrossRef](#)]
25. Lee, H.; Bang, H. Star Pattern Identification Technique by Modified Grid Algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 1112–1116. [[CrossRef](#)]
26. Wei, X.; Wen, D.; Song, Z.; Xi, J.; Zhang, W.; Liu, G.; Li, Z. A star identification algorithm based on radial and dynamic cyclic features of star pattern. *Adv. Space Res.* **2019**, *63*, 2245–2259. [[CrossRef](#)]
27. Liu, H.; Wei, X.; Li, J.; Wang, G. A Star Identification Algorithm Based on Recommended Radial Pattern. *IEEE Sens. J.* **2022**, *22*, 8030–8040. [[CrossRef](#)]
28. Schiattarella, V.; Spiller, D.; Curti, F. A novel star identification technique robust to high presence of false objects: The Multi-Poles Algorithm. *Adv. Space Res.* **2017**, *59*, 2133–2147. [[CrossRef](#)]
29. Schiattarella, V.; Spiller, D.; Curti, F. Star identification robust to angular rates and false objects with rolling shutter compensation. *Acta Astronaut.* **2020**, *166*, 243–259. [[CrossRef](#)]
30. Mehta, D.S.; Chen, S.; Low, K.S. A robust star identification algorithm with star shortlisting. *Adv. Space Res.* **2018**, *61*, 2647–2660. [[CrossRef](#)]
31. Mehta, D.S.; Chen, S.; Low, K.S. A Rotation-Invariant Additive Vector Sequence Based Star Pattern Recognition. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *55*, 689–705. [[CrossRef](#)]
32. Kolomenkin, M.; Pollak, S.; Shimshoni, I.; Lindenbaum, M. Geometric voting algorithm for star trackers. *IEEE Trans. Aerosp. Electron. Syst.* **2008**, *44*, 441–456. [[CrossRef](#)]
33. Li, J.; Wei, X.; Zhang, G. Iterative algorithm for autonomous star identification. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 536–547. [[CrossRef](#)]
34. Bubeck, S.; Sellke, M. A Universal Law of Robustness via Isoperimetry. *J. ACM* **2023**, *70*, 1–18. [[CrossRef](#)]
35. Rijlaarsdam, D.; Yous, H.; Byrne, J.; Oddenino, D.; Furano, G.; Moloney, D. Efficient Star Identification Using a Neural Network. *Sensors* **2020**, *20*, 3684. [[CrossRef](#)] [[PubMed](#)]
36. Wang, H.; Wang, Z.; Wang, B.; Yu, Z.; Jin, Z.; Crassidis, J.L. An artificial intelligence enhanced star identification algorithm. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 1661–1670. [[CrossRef](#)]
37. Han, J.; Yang, X.; Xu, T.; Fu, Z.; Chang, L.; Yang, C.; Jin, G. An End-to-End Identification Algorithm for Smearing Star Image. *Remote Sens.* **2021**, *13*, 4541. [[CrossRef](#)]
38. Wang, B.; Wang, H.; Jin, Z. An Efficient and Robust Star Identification Algorithm Based on Neural Networks. *Sensors* **2021**, *21*, 7686. [[CrossRef](#)] [[PubMed](#)]
39. Jiang, J.; Liu, L.; Zhang, G. Star Identification Based on Spider-Web Image and Hierarchical CNN. *IEEE Trans. Aerosp. Electron. Syst.* **2020**, *56*, 3055–3062. [[CrossRef](#)]
40. Yang, S.; Liu, L.; Zhou, J.; Zhao, Y.; Hua, G.; Sun, H.; Zheng, N. Robust and Efficient Star Identification Algorithm based on 1-D Convolutional Neural Network. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 4156–4167. [[CrossRef](#)]
41. Huang, B.; Li, Z.H.; Tian, X.Z.; Yang, L.; Zhang, P.J.; Chen, B. Modeling and correction of pointing error of space-borne optical imager. *Optik* **2021**, *247*, 167998. [[CrossRef](#)]
42. Sun, Q.; Liu, L.; Niu, Z.; Li, Y.; Zhang, J.; Wang, Z. A Practical Star Image Registration Algorithm Using Radial Module and Rotation Angle Features. *Remote Sens.* **2023**, *15*, 5146. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.