remote sensing

MDPI

Article

An Autonomous Global Star Identification Algorithm Based on the Fast MST Index and Robust Multi-Order CCA Pattern

Zijian Zhu [1,2,3,4], Yuebo Ma [1,2,3], Bingbing Dan [1,2,4], Enhai Liu [1,2,3], Zifa Zhu [1,2,3,4], Jinhui Yi [1,2,3,4], Yuping Tang [1,2,3,4] and Rujin Zhao [1,2,3,*]

[1] National Key Laboratory of Optical Field Manipulation Science and Technology, Chinese Academy of Sciences, Chengdu 610209, China
[2] Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China
[3] Key Laboratory of Science and Technology on Space Optoelectronic Precision Measurement, Chinese Academy of Sciences, Chengdu 610209, China
[4] University of Chinese Academy of Sciences, Beijing 100149, China
[*] Correspondence: zhaorj@ioe.ac.cn

Abstract: Star identification plays a key role in spacecraft attitude measurement. Currently, most star identification algorithms tend to perform well only in a scene without noise and are highly sensitive to noise. To solve this problem, this paper proposes a star identification algorithm based on the maximum spanning tree (MST) index and multi-order continuous cycle angle (CCA) intended for the lost-in-space mode. In addition, a neighboring star selection method named dynamic eight-quadrant (DEQ) is developed. First, the DEQ method is used to select high-confidence neighboring stars for the main star. Then, the star image is regarded as a graph, and the Prim algorithm is employed to construct the MST pattern for each guide star, which is then combined with the K vector index to perform the main star candidate search. Finally, the Jackard similarity voting for the multi-order CCA of the main star is used to identify the main star, and the global neighboring star identification is conducted by the multi-order CCA of neighboring stars. The simulated and real star images test results show that compared with five mainstream algorithms, when the position noise is 1 pixel, the number of false stars is five, the magnitude noise is 0.5, and the identification accuracy of the proposed algorithm is higher than 98.5%. Therefore, the proposed algorithm has excellent anti-noise ability in comparison to other algorithms.

Keywords: star identification; lost-in-space mode; dynamic eight-quadrant method; maximum spanning tree; continuous cyclic angle

check for updates

Citation: Zhu, Z.; Ma, Y.; Dan, B.; Liu, E.; Zhu, Z.; Yi, J.; Tang, Y.; Zhao, R. An Autonomous Global Star Identification Algorithm Based on the Fast MST Index and Robust Multi-Order CCA Pattern. *Remote Sens.* **2023**, *15*, 2251. https://doi.org/10.3390/rs15092251

Academic Editors: Krzysztof Naus and Mieczysław Bakuła

Received: 1 March 2023
Revised: 19 April 2023
Accepted: 20 April 2023
Published: 24 April 2023

1. Introduction

The measurement of spacecraft attitude refers to determining the orientation of the spacecraft in the inertial coordinate system. This task plays a key role in the fields of astronomical navigation, remote sensing observation, and deep space exploration missions. In recent years, there have been many examples of fatal problems in satellites due to the incorrect operation of the attitude determination system, and a typical one is: "Lewis spacecraft spins out of control [1]." Therefore, the attitude determination system has been considered a vital subsystem in space missions [2]. The most critical component in the attitude determination system is an aerospace attitude sensor. At present, the commonly used aerospace attitude sensors mainly include sun sensors, infrared horizon sensors, gyroscopes, and star sensors. Compared with the other types of sensors, the star sensor provides more accurate three-axis attitude data in terms of arc seconds and can output attitude without knowing the prior attitude [3]. Due to the mentioned advantages, currently, a star sensor is the most widely used type of aerospace attitude sensor [4].

The workflow of a star sensor mainly includes two stages: offline and online. The offline stage is to load the star pattern database (SPD) on the ground. In contrast, the online stage mainly includes three steps: on-orbit centroid extraction, star identification, and attitude calculation. In addition, a star sensor has two operational modes: lost-in-space (LIS) mode and tracking mode [5]. The LIS mode occurs when there is no prior attitude or the attitude data are lost due to a malfunction [6]. In such a case, the star identification in the full-sky area is performed to determine the initial attitude, but this has high requirements for the robustness and real-time performance of a star identification algorithm. Then, a sensor enters the tracking mode and starts the star searching and identification process in a local area using the initial attitude data. Thus, star identification in the LIS mode is more challenging than that in the tracking mode.

In a realistic space deployment scenario, there can be certain deviations between the captured images and pre-stored guide stars in some boresight directions, which can have a severe impact on the star identification result. The main reasons for these deviations may be due to the following: the interference stars appear on the imaging plane due to radiation from nearby planets and space debris, causing the false star noise; the interference of the internal circuit of a star sensor might lead to high-magnitude star missing, which represents the magnitude noise; the focal length and principal point calibration errors of a star sensor and the jitter of a satellite platform can lead to a deviation in the star imaging position, resulting in the position noise. The listed cases have a high possibility of causing a star identification failure. Therefore, fast and accurate star identification in the LIS mode is extremely challenging, and many star identification algorithms have been proposed in recent years. Generally, the mainstream star identification algorithms can be divided into three categories: the subgraph-based class, pattern-based class, and learning class [7–11]. Typical, recently published star identification algorithms of three different categories are listed in Table 1.

Table 1. Classification of mainstream star identification algorithms.

Algorithm	Category	Methodology
Sun et al. [12]	Subgraph-based	Double-triangle with the star angle and distance
Liu et al. [13]	Subgraph-based	Triangle voting scheme
Kolomenkin et al. [14]	Subgraph-based	Geometric voting strategy
Mortari et al. [15]	Subgraph-based	Pyramid structure and K vector search
Cole et al. [16]	Subgraph-based	Area and polar moments of triangles
Lee et al. [17]	Pattern-based	Polar grid and multi-reference stars
Li et al. [18]	Pattern-based	Two-dimensional angular distances
Aghaei et al. [19]	Pattern-based	Optimization method
Zhang et al. [20]	Pattern-based	Radial and cyclic pattern
Samirbhai et al. [21]	Pattern-based	Rotation-invariant 2D vector
Jiang et al. [22]	Pattern-based	Redundant-coded star pattern
Liu et al. [23]	Pattern-based	A priori algorithm
Jiang et al. [4]	Learning-based	Hierarchical convolutional neural network (CNN)
Yang et al. [9]	Learning-based	1D-CNN

The so-called subgraph-based algorithms use a geometric structure between the stars to identify stars. The SPD is regarded as a graph, and the captured image is regarded as a subgraph, where graph vertices denote stars, and graph edges represent angular distances between the stars. Thereafter, the star identification is realized through the correspondence of vertices and edges between the subgraph and the SPD. The most classical algorithm of this type is the triangle algorithm [24], which regards the three brightest stars in the field of view (FOV) as vertices and angular distances between every two vertices as three edges of a triangle, thus constructing a unique triangle. This algorithm is simple and has been one of the most commonly used algorithms in engineering. However, it has the disadvantage of low feature dimensions, which makes redundant matching prone to occur. In addition, as the detection capability of a star sensor increases, the SPD memory increases exponentially, resulting in a significant decrease in search efficiency. Aiming to solve the aforementioned problems, recent research has proposed many triangle-derived algorithms [12–16], of which the most representative is the pyramid algorithm [15] that has been successfully applied to the High-Energy Transient Explorer (HETE) task. In this algorithm, four stars are used to form a pyramid structure, and the feature dimension is improved, reducing the probability of redundant matching. Furthermore, the $\boldsymbol{K}$ vector algorithm was introduced, which greatly improves search efficiency. In [14], the authors developed a voting strategy based on the relationship between stars to vote, but the identification accuracy decreased significantly with the number of false stars. In view of that, in [13], the authors proposed a multi-layer voting algorithm, which can slightly improve the anti-noise ability, but its SPD storage is large. Overall, the subgraph-based algorithms generally face the problem of large SPD storage memory.

Compared with the subgraph-based algorithms, the pattern-based algorithms typically have a smaller SPD. The first proposed algorithm of this type was the grid algorithm [25,26], which aims to construct a unique star pattern based on the positional relationship between the main star and the nearest neighboring star, called the reference star. However, due to over-reliance on the reference star, the identification accuracy decreases rapidly when there is noise interference. On this basis, many improved algorithms were derived [17–19,27]. For instance, in [17], a grid pattern vector in the polar grid was constructed, and a multi-reference stars method was proposed, but the problems of dependence on the reference star and high sensitivity to noise remained unsolved. In [18], the authors used a two-dimensional angular distance pattern to replace the original grid pattern and combined the traversal method to determine the reference star, which improved the robustness to noise. However, the traversal method has high computational complexity, resulting in reduced real-time performance. In addition, a series of algorithms for circular and radial patterns were developed [20,22,28]. In [22], an algorithm combining radial and neighboring star patterns was proposed, and redundant coding was used for storage, which reduced the storage capacity of the SPD. Further, in [23], companion stars were determined by the a priori algorithm, and a radial pattern image was generated, but only the selected neighboring stars could be identified. In [21], an innovative rotation-invariant two-dimensional vector pattern was designed; this design is robust against the positional noise, but its identification accuracy drops rapidly with the number of false stars. In [29], a combination of hamming distance and spearman correlation was adopted for star identification. However, this approach could identify only one to two stars adjacent to the main star. The general disadvantage of the pattern-based algorithms is that most of them perform poorly under the aforementioned three types of noises and cannot achieve global star identification.

Learning-based algorithms essentially represent a class of pattern-based algorithms [30]. Since CNN was applied to star identification [31], many CNN-based algorithms for star identification have been proposed, and the most representative ones use spider-web images and hierarchical CNN [4], mixed star patterns and multi-layer SOM neural networks [32], and 1D CNN [9]. They input the constructed patterns or star images into the network for training to classify and identify the guide stars in a captured star image. Although this kind of algorithm can improve the identification accuracy, the iterative updating of network weights results in a long running time. In addition, the memory consumption of learning-based algorithms is generally large.

Through the aforementioned analysis, it is found that star identification mainly has the disadvantages of a long running time, large memory usage, and sensitivity to noise. In this paper, with reference to pattern-based algorithms, a global star identification algorithm based on the maximum spanning tree fast index and continuous cycle angle is proposed, which greatly improves the search time and anti-noise ability of pattern-based star identification.

The main contributions of this study can be summarized as follows:

1. A dynamic eight-quadrant method for neighboring stars selection, which makes the guide stars in the FOV uniformly distributed and increases the identifiability of the

constructed pattern, is proposed. This provides a novel idea for the selection of neighboring stars;

2. The Prim algorithm is introduced into the field of star identification first, constructing the maximum spanning tree pattern for each main star, and is then combined with the ***K*** vector to define a fast index, which greatly improves the search efficiency of the main star;
3. A multi-order continuous cycle angle pattern is designed and used to perform the global identification of neighboring stars, which improves the anti-noise performance of the pattern-based algorithm;
4. Extensive experiments are conducted on simulated and real star images, and the experimental results show that the proposed algorithm is superior to most mainstream algorithms in terms of identification accuracy, memory, and time consumption.

The rest of this paper is organized as follows. Section 2 introduces the pre-knowledge and pattern framework. Section 3 presents the proposed algorithm, including the strategies for LUT, SPD generation, and star identification. Section 4 compares the proposed algorithm with five mainstream algorithms. Section 5 discusses the advantages and limitations of this study. Finally, Section 6 concludes this study. Appendix A summarizes all the abbreviations used in the study.

2. Pre-Knowledge and Pattern Framework

2.1. Pre-Knowledge

This section briefly introduces the terms involved in this study.

Guide star catalog: The original star catalog used in this study is the Smithsonian Astrophysical Observatory (SAO) star catalog, which contains 258,997 stars with a magnitude dimmer than 11 and includes data on the right ascension (α_i), declination (β_i), apparent magnitude, and other astronomical parameters of each star [33]. According to the detection capability of the star sensor used in this study, the stars with a magnitude below 6 are selected to form the guide star catalog, and a two-dimensional distribution graph is drawn according to the position coordinates (α_i,β_i) of the selected stars, as shown in Figure 1.

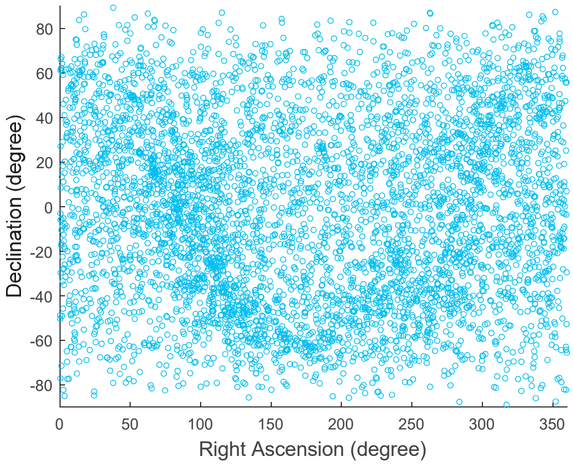


Figure 1. The two-dimensional distribution of the guide stars with a magnitude lower than 6.

Star sensor: The model diagram of the star sensor imaging the stars on the imaging plane is shown in Figure 2, where the reddish-brown star represents the main star (MS) in the current FOV, the blue star represents the star in the FOV but not selected as neighboring star (NS), the orange star represents the star selected as NS, and the number of NSs is denoted by N; the orange stars represent other stars in the celestial sphere, and A_{ij} is the angular distance between stars i and j.

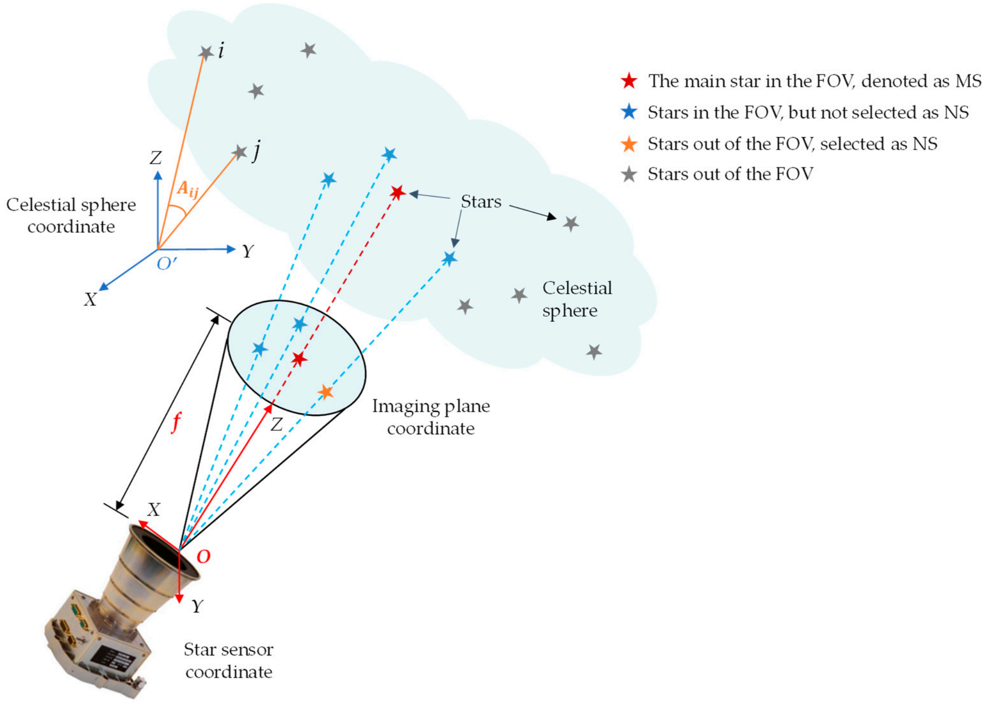

i
j
Z
Y
X
O′
A_{ij}
Celestial sphere coordinate
Stars
Celestial sphere
f
Z
Imaging plane coordinate
X
O
Y
Star sensor coordinate
★ The main star in the FOV, denoted as MS
★ Stars in the FOV, but not selected as NS
★ Stars out of the FOV, selected as NS
★ Stars out of the FOV

Figure 2. The illustration of the star imaging process by the star sensor.

Continuous cyclic angle: The maximum FOV radius is denoted by PR_{max}, and the continuous cyclic angle (CCA) is defined as an angle between the lines connecting the MS and any two NSs in the FOV. The CCA for N = 8 is presented in Figure 3; Figure 3a shows the angle between the adjacent NSs, denoted by θ_i (i = 1, 2, ..., N), which represents the first-order CCA pattern of an NS; Figure 3b shows the multi-order CCA pattern of NS_1, denoted by $\theta_{1,m}$ (m = 2, 3, . . . , N). Among them, $\theta_{1,2} = \theta_1$, $\theta_{1,3} = \theta_1 + \theta_2$, . . . , $\theta_{1,8} = \theta_1 + \theta_2 + \theta_3 + \theta_4 + \theta_5 + \theta_6 + \theta_7$. The CCAs of other orders of NS_1 are calculated in the same way. Similarly, the multi-order CCA patterns of other NSs are constructed.

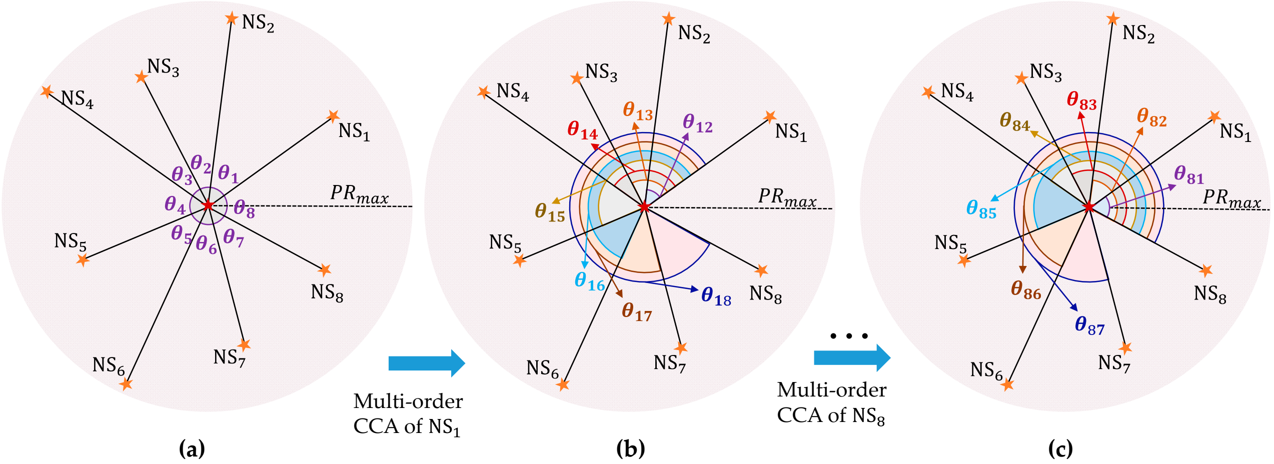


Figure 3. Schematic diagram of the multi-order CCA construction process. (**a**) First-order CCA of all NSs; (**b**) Multi-order CCA of NS_1; (**c**) Multi-order CCA of NS_8.

Graph structure: A star image is regarded as an undirected graph in this study; namely, all stars are regarded as a vertex set (V) of graphs (G), and all inter-star connections are regarded as an edge set (E); the Euclidean distance of E is regarded as an edge weight. The maximum spanning tree (MST) represents the set of the maximum edge weights that connect all vertices without loops, and it can be represented by a set $\mathrm{T} = \{d(u,v)|(u,v) \ \in$

$V\}$, where $d(u,v)$ represents the edge weight for the MST. Therefore, the value of MST can be calculated as $w(\mathrm{T}) = \sum_{d(u,v)\in \mathrm{T}} d(u,v)$.

2.2. Proposed Rotation-Invariant Pattern Frames

Ideally, the star sensor should be imaged in the same position as a star when the SPD is constructed. However, in the real imaging process, it is common that a star rotates on the imaging plane, and constructing a rotation-invariant pattern is the primary task in the star identification algorithm design.

In this study, a multi-order CCA pattern star image identification framework based on the MST fast index is proposed. The construction and identification principles of the proposed framework are explained in detail in Section 3, and in this section, the rotation invariance of the proposed framework is explained.

First, the projection transformation steps of the star sensor are analyzed. According to (α_i, β_i) of the guide star, the coordinates of the ith guide star in the earth-centered inertial (ECI) coordinate system can be expressed as follows:

$$\boldsymbol{v}_i^{ECI} = [X_i^{ECI}; Y_i^{ECI}; Z_i^{ECI}] = [\cos\alpha_i \cos\beta_i; \sin\alpha_i \cos\beta_i; \sin\beta_i]. \tag{1}$$

Then, the coordinates of the ith guide star in the star sensor coordinate system are defined by:

$$\boldsymbol{v}_i^{sen} = [X_i^{sen}; Y_i^{sen}; Z_i^{sen}] = \boldsymbol{C} \cdot \boldsymbol{v}_i^{ECI}, \tag{2}$$

where $\boldsymbol{C}$ is the attitude rotation matrix [1], which converts the ECI coordinates $\left[X_i^{ECI},\ Y_i^{ECI},\ Z_i^{ECI}\right]$ to the star sensor coordinates $\left[X_i^{sen},\ Y_i^{sen},\ Z_i^{sen}\right]$.

Finally, the coordinates of the ith guide star on the imaging plane $(x_i,\ y_i)$ are calculated by:

$$\begin{cases} x_i = \frac{fX_i^{sen}}{p_x Z_i^{sen}} + \frac{D_x}{2} \\ y_i = \frac{fY_i^{sen}}{p_y Z_i^{sen}} + \frac{D_y}{2} \end{cases}, f = \frac{D_x p_x}{2\tan(\frac{FOV_x}{2})} = \frac{D_y p_y}{2\tan(\frac{FOV_y}{2})}, \tag{3}$$

where p_x, D_x, and FOV_x denote the pixel size, the number of pixels, and the degree of FOV in the x-dimension, respectively; the meaning of the parameters in the y-dimension is the same.

Next, the rotation invariance analysis of the CCA pattern is performed. The first-order CCA of all NSs is expressed as follows:

$$\theta_i = \begin{cases} \left[\arccos\left(\frac{dot((x_i - x_m, y_i - y_c),(x_{i+1} - x_m, y_{i+1} - y_m))}{||x_i - x_m, y_i - y_m||_2 \cdot ||x_{i+1} - x_m, y_{i+1} - y_m||_2}\right)\right], i = 1, \ldots, N-1, \\ \left[\arccos\left(\frac{dot((x_N - x_m, y_N - y_m),(x_1 - x_m, y_1 - y_m))}{||x_N - x_m, y_N - y_m||_2 \cdot ||x_1 - x_m, y_1 - y_m||_2}\right)\right], i = N, \end{cases} \tag{4}$$

where $(x_i,\ y_i)$ are the coordinates of the ith NS on the imaging plane, and $(x_m,\ y_m)$ are the coordinates of the MS of the imaging plane; function $dot(\boldsymbol{b}_1, \boldsymbol{b}_2)$ returns the dot product of vectors $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$; $||\cdot||_2$ represents the two-norm of a vector; $[\cdot]$ indicates rounding to the nearest integer.

Assume that $\boldsymbol{S} = [\theta_1, \theta_2, \ldots, \theta_N]$, which represents the first-order CCA set; then, according to the CCA definition and Equation (4), the ith NS multi-order CCA pattern of the kth MS can be expressed as follows:

$$\begin{aligned} &\boldsymbol{A}^k[i] = [\boldsymbol{S}_i, \boldsymbol{S}_i + \boldsymbol{S}_{i>>1}, \ldots, \boldsymbol{S}_i + \boldsymbol{S}_{i>>1} + \ldots + \boldsymbol{S}_{i>>(N-1)}], \\ &s.t.\ 1 < i \leq N \end{aligned} \tag{5}$$

where the symbol ">>" means that vector $\boldsymbol{S}$ is shifted circularly to the right.

Further, define the matrix consisting of $\boldsymbol{A}^k[i]$ $(i = 1,\ 2, \ldots,\ N)$ as a multi-order CCA pattern of the kth MS as follows:

$$\boldsymbol{A}^k = [\boldsymbol{A}^k[1]; \boldsymbol{A}^k[2]; \ldots; \boldsymbol{A}^k[N]], \tag{6}$$

where the rows in $\boldsymbol{A}^k$ represent the $\boldsymbol{A}^k[i]$ $(i = 1,\ 2, \ldots,\ N)$ of NSs, and the columns denote the zth-order CCA patterns, where $\mathrm{z} = 1,\ 2, \ldots,\ N-1$.

As a star sensor rotates, the imaging plane coordinates of stars change. According to Equation (4), the rotated first-order CCA can be expressed by:

$$\theta_i^{R_\delta} = \begin{cases} \left[\arccos\left(\frac{R_\delta \cdot dot((x_i - x_m, y_i - y_m),(x_{i+1} - x_m, y_{i+1} - y_m))}{\sqrt{R_\delta} \cdot \sqrt{R_\delta} \cdot \left|\left|x_i - x_m, y_i - y_m\right|\right|_2 \cdot \left|\left|x_{i+1} - x_m, y_{i+1} - y_m\right|\right|_2}\right)\right], i = 1, \ldots, N-1, \\ \left[\arccos\left(\frac{R_\delta \cdot dot((x_N - x_m, y_N - y_m),(x_1 - x_m, y_1 - y_m))}{\sqrt{R_\delta} \cdot \sqrt{R_\delta} \cdot \left|\left|x_N - x_m, y_N - y_m\right|\right|_2 \cdot \left|\left|x_1 - x_m, y_1 - y_m\right|\right|_2}\right)\right], i = N, \end{cases} \tag{7}$$

where R_δ denotes the rotation matrix obtained after the rotation by angle δ, and it is given by:

$$R_\delta = \begin{bmatrix} \cos\delta & -\sin\delta \\ \sin\delta & \cos\delta \end{bmatrix}. \tag{8}$$

Based on Equations (4) and (7), it holds that $\theta_i = \theta_i^{R_\delta}$. This shows that the first-order CCA set $\boldsymbol{S} = [\theta_1, \theta_2, \ldots, \theta_N]$ does not change with rotation, while the multi-order CCA matrix $\boldsymbol{A}^k$ is constructed based on the set $\boldsymbol{S}$, so its value is also unchanged. Thus, the rotation changes only the row order of the pattern matrix $\boldsymbol{A}^k$, while its value remains unchanged and can be considered rotation-invariant. The row alignment is described in Section 3.2.

Finally, the MST construction only uses the Euclidean distance of stars on the imaging plane. According to the MST definition given in Section 2.1 and the construction process presented in Section 3.1.2, the MST pattern is also rotation-invariant. Therefore, both patterns constructed in this study are rotation-invariant. In addition, the proposed dynamic eight-quadrant NS selection method ensures the invariance of the star pattern, which is explained in Section 3.1.1.

3. Proposed Methodology

Inspired by the pattern-based algorithms, this study proposes an innovative star identification algorithm based on the MST fast index and multi-order CCA pattern. In addition, this study develops a guide star selection method named the DEQ method, which can make the guide stars in the FOV more uniform, thus ensuring the pattern is more identifiable. The proposed algorithm includes two main tasks: offline LUT and SPD generation and online star identification. The specific steps of the proposed algorithm are as follows. In the offline step, each star at the center of the FOV is denoted as MS, a unique MST look-up table (LUT) index is constructed, and a multi-order CCA pattern is generated to obtain the SPD. In the online step, both patterns are constructed from the captured or simulated images. First, the candidate MS set is obtained using the $\boldsymbol{K}$ vector value from the LUT, and then the unique MS and NSs in the image are identified based on the CCA pattern. The flowchart of the proposed algorithm is shown in Figure 4, where DMS and DNS represent the MS and NS of the LUT and SPD, respectively; SMS and SNS denote the MS and NS of the captured and simulated images, respectively. In Figure 4, the upper and lower structures represent two major steps, and the right side presents the rough flow of the proposed global star identification.

3.1. LUT and SPD Generation

Before constructing the offline LUT and SPD, it is necessary to preprocess the guide star catalog, which includes performing magnitude filtering and double star merging [34] and then obtaining N_s = 4956 stars. Next, the LUT and SPD construction steps are conducted. Generally, if the number of stars in the FOV is too small, the star identification will fail; in contrast, if the number of stars in the FOV is too large, the star pattern will be redundant, which will also cause identification errors. Therefore, it is essential to select an optimal number of stars before constructing the star pattern. The number of stars in different FOVs is presented in Figure 5, where it can be seen that there are redundant stars in most of the boresight for FOV = 20° × 20°. To solve this problem, this study proposes a dynamic eight-quadrant NS selection method named the DEQ method. The NSs selected by the DEQ method are more uniformly distributed in the FOV, which makes the constructed patterns more identifiable.

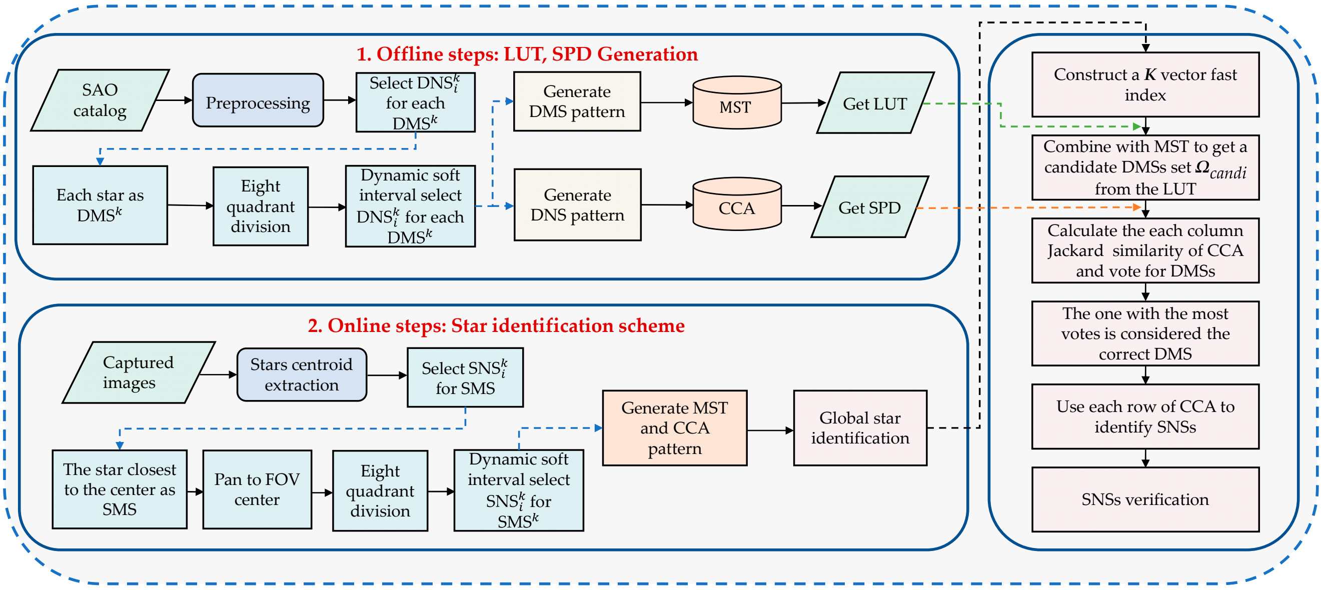

1. Offline steps: LUT, SPD Generation
SAO catalog
Preprocessing
Select DNS_i^k for each DMS^k
Generate DMS pattern
MST
Get LUT
Each star as DMS^k
Eight quadrant division
Dynamic soft interval select DNS_i^k for each DMS^k
Generate DNS pattern
CCA
Get SPD
2. Online steps: Star identification scheme
Captured images
Stars centroid extraction
Select SNS_i^k for SMS
Generate MST and CCA pattern
Global star identification
The star closest to the center as SMS
Pan to FOV center
Eight quadrant division
Dynamic soft interval select SNS_i^k for SMS^k
Construct a K vector fast index
Combine with MST to get a candidate DMSs set Ω_{candi} from the LUT
Calculate the each column Jackard similarity of CCA and vote for DMSs
The one with the most votes is considered the correct DMS
Use each row of CCA to identify SNSs
SNSs verification

Figure 4. The flowchart of the proposed algorithm consisting of the offline and online steps.

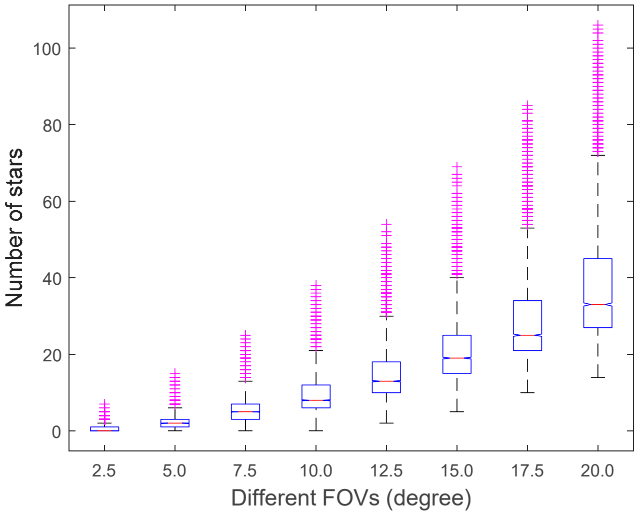


Figure 5. The distribution of star numbers for different FOVs.

3.1.1. DEQ Method

The DEQ method involves two main steps: eight-quadrant division and dynamic soft interval select DNS_i^k, which are explained in the following.

Step 1: Eight-quadrant division: In the guide star catalog, place each star in the center of the FOV in turn and select the center star as the DMS^k $(k = 1, 2, \ldots, N_s)$ each time, which is denoted by reddish-brown in Figure 6a. Then, set the minimum field radius $PR_{min} = 0.1°$, avoiding selecting a DNS_i^k $(i = 1, \ldots, N)$ that is too close to the DMS^k. Further, initialize the dynamic radius PR_d, as shown in Figure 6b. Based on the x and y axes and the diagonal direction, the eight quadrants are divided, as shown in Figure 6c;

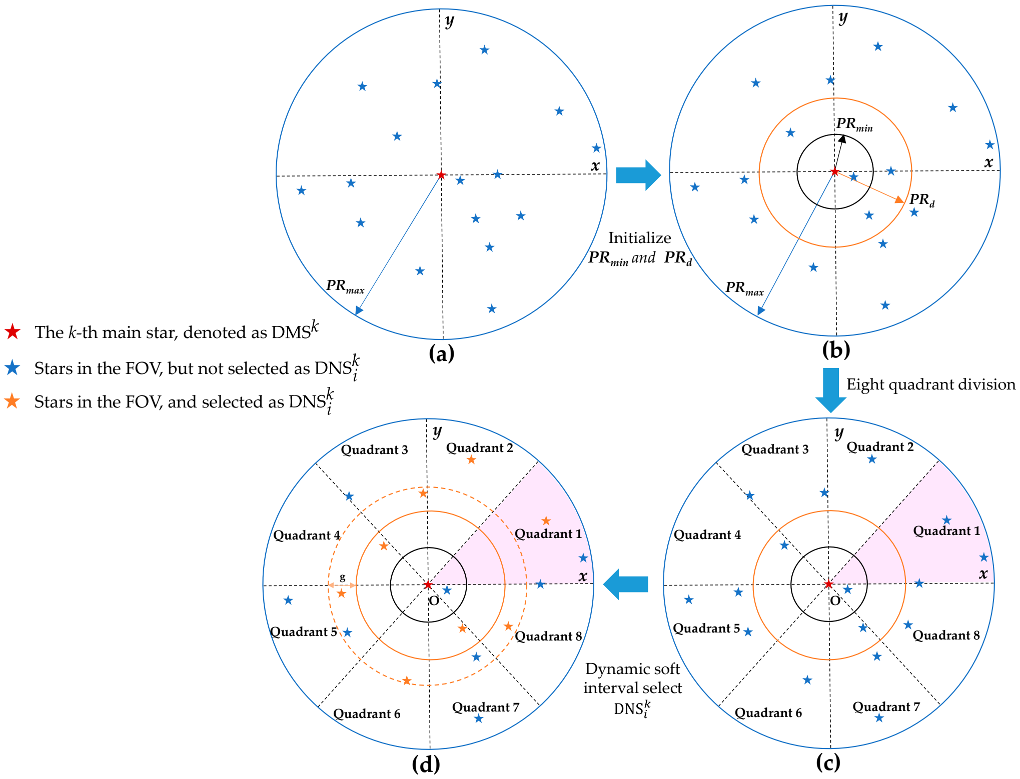


Figure 6. The NS selection rules in the proposed DEQ method. (**a**) Distribution of redundant NSs in the FOV; (**b**) Parameter settings; (**c**) Eight-quadrant division based on the imaging plane coordinate system; (**d**) According to λ_i of the stars in each quadrant, a dynamic NS selection is performed.

Step 2: Dynamic soft interval select DNS_i^k: First, set an initial value of g to make PR_d change dynamically within (PR_{min}, PR_{max}), where PR_{max} is $FOV/2$, as shown in Figure 6d. Moreover, denote the maximum number of DNSs by N_{max}. The values of N_{max} and g and the initial value of PR_d are given in Table 2. The parameter N_{max} is to ensure that there is a guide star in each quadrant in most boresight directions; therefore, the value is set to eight. By randomly generating 10,000 boresight directions, most of the guide stars under the FOV = 20° × 20° are distributed within the FOV with a radius of 5°, so PR_d = 5° is set; moreover, it is set to g = 1.25°. When the FOV radius of 5° does not meet the aforementioned distributions, then it can be satisfied after one or two iterations.

Table 2. Parameter settings of the DEQ method.

Parameters	Illustration	Value
N_{max}	Maximum number of NSs	8
g	Dynamic step size	1.25 (deg)
P_d	Initial radius	5 (deg)

Next, the process of the DEQ method is explained in detail.

In this study, the confidence factor λ_i is defined for each DNS_i^k by Equation (9), which served as an NS selection criterion for each quadrant.

$$\lambda_i = -\left(\frac{Mag_i - Mag_{min}}{Mag_{max} - Mag_{min}} + \frac{r_i}{d_{max}} * 2\right), \tag{9}$$

where Mag_i and r_i are the magnitude and the Euclidean distance from the center of the ith star, respectively; Mag_{min} and Mag_{max} are the minimum and maximum magnitudes, respectively; $d_{max} = \sqrt{{D_x}^2 + {D_y}^2}$ is the maximum imaging plane distance.

Next, select a star from each of the eight quadrants in the range (PR_{min}, PR_d). The process is as follows: according to the definition of λ_i, the lower the Mag_i and r_i, the larger the λ_i value is, and the easier it is for a star to be selected as DNS_i^k. Thus, the DNS with the largest λ_i value in each quadrant is selected, as shown in Figure 6d. When the number of selected DNSs is less than $N_{\max}$, PR_d is incremented by g to increase the field range; this process is repeated until PR_d = PR_{max} to meet the condition that there is a star in all eight quadrants, which is defined as the eight-quadrant distribution. However, for sky areas with a small number of stars, if the number of stars does not meet the eight-quadrant distribution, the stars are directly arranged according to the descending order of λ_i, and the top $N_{\max}$ stars are selected as DNSs. Compared to the direct selection of the brightest $N_{\max}$ stars in the FOV, which represents the BR method, or $N_{\max}$ stars closest to the DMS, which is the ED method, the proposed DEQ method can effectively reduce the noise interference caused by missing and false stars and position deviation. The performance of the proposed DEQ method is demonstrated in Section 4.1.

3.1.2. MST and CCA Pattern Construction

In the proposed DEQ method, a uniform distribution and a relatively fixed number of DNSs are selected for each DMS^k, which paves the way for the construction of more identifiable MST and CCA patterns.

The first step is based on the LUT of MST, which is to index the candidate MSs to speed up the star identification process. As mentioned in Section 2.1, stars captured by a star sensor are regarded as a vertex of **G**, and the weight between two vertices (u,v) is defined as their Euclidean distance on the imaging plane, which is calculated by:

$$\begin{aligned} &d(u,v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}, \\ &s.t. \quad u,v \in \mathrm{V} \end{aligned} \tag{10}$$

where (x_u, y_u) and (x_v, y_v) are the coordinates of the two stars on the imaging plane.

Next, the MST pattern is constructed. There are two commonly used algorithms for MST construction: the Kruskal algorithm and the Prim algorithm. Among them, the Kruskal algorithm is more suitable for sparse graphs, whereas the Prim algorithm is more appropriate for dense graphs [35]. The general judgment principle for dense and sparse graphs is defined as follows:

$$\begin{cases} L_E \geq L_V * \log L_V, & \text{dense graphs}, \\ \text{otherwise}, & \text{sparse graphs}, \end{cases} \tag{11}$$

where L_E represents the number of edges, and L_V represents the number of vertices.

In this study, the number of vertices represents the number of stars in the FOV after screening by the DEQ method, that is, $L_V = N + 1 = 9$. Considering the edge weights between all stars in the FOV, there are nine vertices, with $L_E = \mathrm{C}_9^2 = 36$ edges. After substituting L_V and L_E into Equation (11), it can be concluded that this problem belongs to the MST construction problem of dense graphs. Therefore, this study uses the Prim algorithm for the MST pattern construction of each guide star.

The detailed steps of the MST index construction process are as follows. First, define V = {NS_1, NS_2, NS_3, NS_4, MS, NS_5, NS_6, NS_7, NS_8} as a set of all vertices and U as a set of MST vertices. Randomly select a star from the guide stars selected in the eight quadrants as a starting point (i.e., NS_3 in Figure 7a), use Equation (10) to calculate the distance between it and other stars as an edge weight, and select the maximum edge weight value as r_1, as shown in Figure 7b. In addition, the vertex of MST is denoted by U = {NS_3, NS_6}. Then, select the edge with the largest weight from all edges connected with NS_3 and NS_6 as the second edge r_2, as shown in Figure 7c; then, the MST vertex is expressed by U = {NS_2, NS_3, NS_6}. According to the aforementioned steps and the rule that any three vertices cannot form a loop, when all nine vertices are selected as MST vertices, the MST pattern construction of the DMS^k is completed, as shown in Figure 7d. The final MST pattern calculation is given by:

$$w_D(T^k) = \sum_{i=1}^{N} d_i(u,v). \tag{12}$$

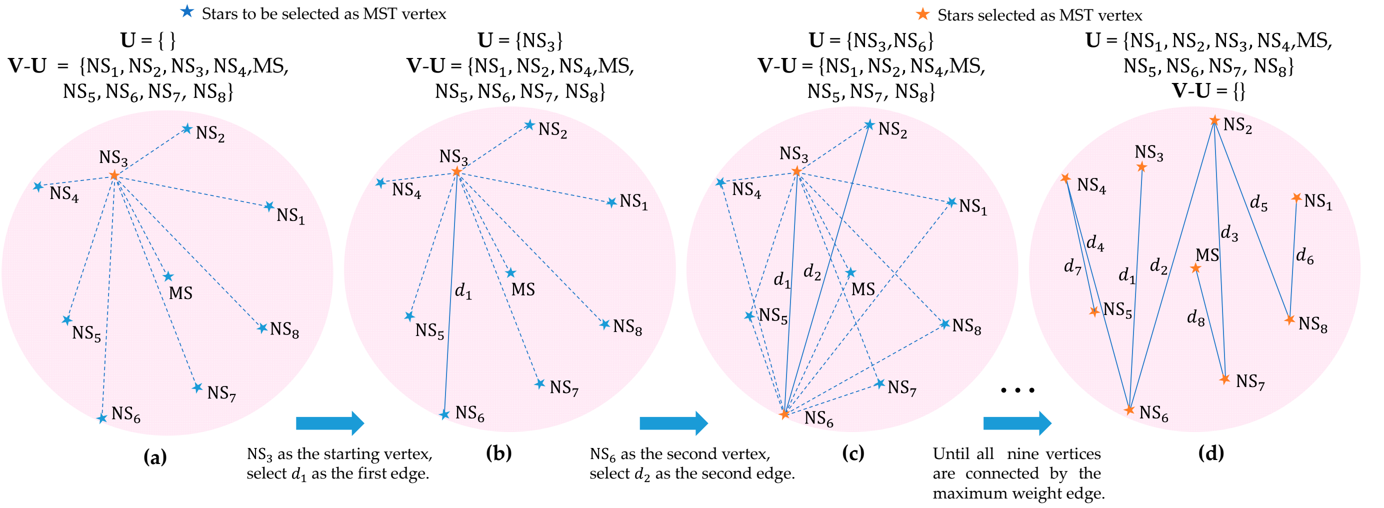

★ Stars to be selected as MST vertex
★ Stars selected as MST vertex
U = { }
V-U = {NS$_1$, NS$_2$, NS$_3$, NS$_4$,MS, NS$_5$, NS$_6$, NS$_7$, NS$_8$}
U = {NS$_3$}
V-U = {NS$_1$, NS$_2$, NS$_4$,MS, NS$_5$, NS$_6$, NS$_7$, NS$_8$}
U = {NS$_3$,NS$_6$}
V-U = {NS$_1$, NS$_2$, NS$_4$,MS, NS$_5$, NS$_7$, NS$_8$}
U = {NS$_1$, NS$_2$, NS$_3$, NS$_4$,MS, NS$_5$, NS$_6$, NS$_7$, NS$_8$}
V-U = {}
NS$_1$
NS$_2$
NS$_3$
NS$_4$
NS$_5$
NS$_6$
NS$_7$
NS$_8$
MS
d_1
d_2
d_3
d_4
d_5
d_6
d_7
d_8
(a)
NS$_3$ as the starting vertex, select d_1 as the first edge.
(b)
NS$_6$ as the second vertex, select d_2 as the second edge.
(c)
. . .
Until all nine vertices are connected by the maximum weight edge.
(d)

Figure 7. The MST pattern construction steps when the Prim algorithm is used. (**a**) Randomly select the starting vertex (NS$_3$) and calculate the Euclidean distance to the other stars; (**b**) Select the edge with the largest distance as a weight of the first edge (r_1); (**c**) Select another vertex (NS$_6$) connected to r_1 as the second vertex; similarly, select the weight of the second edge (r_2); (**d**) Construct the eight maximum edge weights.

After obtaining the MST pattern of each MS, store the generated patterns in a one-dimensional vector ***Y*** with a length of N_s, denoted by $\mathbf{Y}(k) = \{\mathrm{w_D}(\mathrm{T}^1), \mathrm{w_D}(\mathrm{T}^2), \ldots, \mathrm{w_D}(\mathrm{T}^{N_s})\}$. In addition, to increase the search efficiency of the MST candidate MS, the K vector algorithm is used to locate the search interval rapidly. Namely, it takes only a few simple steps to perform interval indexing fast, and the corresponding time complexity is O(1).

The specific steps in combining the K vector with the MST are as follows:

(1) Arrange vector Y in ascending order, that is, Y(1) = Y$_{\text{min}}$, Y(N_s)= Y$_{max}$. The average step size occupied by each element in vector Y is L_d= $(\mathrm{Y}(N_s) - \mathrm{Y}(1))/(N_s - 1)$. Mortari et al. [15] pointed out that the straight line connecting (1, Y(1) − L_d/2), (N_s, Y(N_s) + L_d/2) can ensure that each step contains Y(k). Therefore, a straight line y(x) is drawn according to the vector Y, which can be expressed as follows:

$$\begin{cases} y(x) = a_1 x + a_0, \\ a_1 = \frac{\mathrm{Y}(N_s) - \mathrm{Y}(1) + L_d}{N_s - 1}, \\ a_0 = \mathrm{Y}(1) - a_1 - L_d/2. \end{cases} \tag{13}$$

(2) Define the K vector index function K(x) as follows:

$$\begin{cases} \mathrm{K}(x) = k, \\ \mathrm{Y}(k) < y(x) < \mathrm{Y}(k+1), \end{cases} \tag{14}$$

where k is the serial number of DMS^k in the LUT, and x is the MST index value.

Next, the CCA pattern in the SPD is constructed. The CCA definition is introduced in Section 2.1; therefore, in this section, only the construction process is presented. First, the first-order CCA set that is shown in Figure 8a is denoted as $\mathbf{S} = [\theta_1, \theta_2, \ldots, \theta_N]$, and the corresponding calculation formula is given in Equation (4). The DMS^k multi-order CCA of the ith DNS$_i^k$ (i = 1, 2, . . . , N) is calculated by Equations (5) and (6). The multi-order CCA construction process of eight NSs is presented in Figure 8b,c, where θ_{uv} represents the counterclockwise angle between the vertices u and v; for instance, $\theta_{85} = \theta_8 + \theta_1 + \theta_2 + \theta_3 + \theta_4$. The multi-order CCA pattern of the DMS^k is presented in Figure 8d, and it is used to select the correct MS from the candidate DMS set obtained by the MST if the candidate star obtained by the MST is not unique, as well as to identify NSs. The multi-order CCA of each DMS^k is an $N \times (N - 1)$ matrix, where each column denotes the CCA pattern of DMS^k from the first order to the (N – 1)th order, which is used to obtain the correct MS from the candidate DMS set; each row in the matrix represents the multi-order CCA of DNS$_i^k$, which is used to identify NSs. After obtaining the correct MS, the row identification can be directly performed to obtain the correct NS. The identification process is described in detail in Section 3.2.

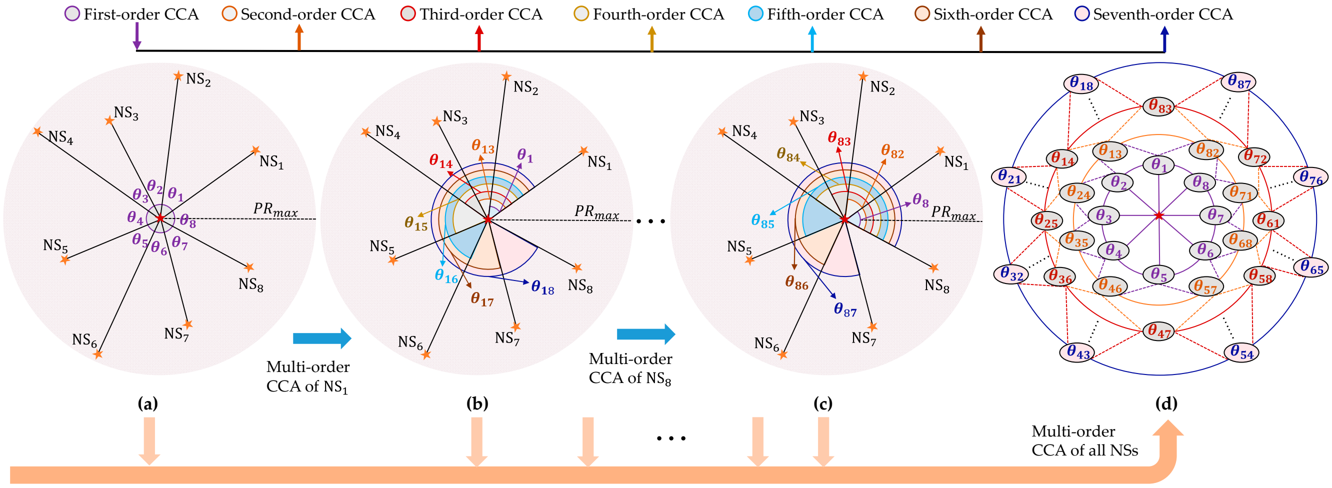

○ First-order CCA
○ Second-order CCA
○ Third-order CCA
○ Fourth-order CCA
○ Fifth-order CCA
○ Sixth-order CCA
○ Seventh-order CCA
NS$_1$
NS$_2$
NS$_3$
NS$_4$
NS$_5$
NS$_6$
NS$_7$
NS$_8$
PR_{max}
θ_1 θ_2 θ_3 θ_4 θ_5 θ_6 θ_7 θ_8
θ_{13} θ_{14} θ_{15} θ_{16} θ_{17} θ_{18}
θ_{82} θ_{83} θ_{84} θ_{85} θ_{86} θ_{87}
θ_{21} θ_{24} θ_{25} θ_{32} θ_{35} θ_{36} θ_{43} θ_{46} θ_{47} θ_{54} θ_{57} θ_{58} θ_{61} θ_{65} θ_{68} θ_{71} θ_{72} θ_{76}
Multi-order CCA of NS$_1$
Multi-order CCA of NS$_8$
Multi-order CCA of all NSs
(a)
(b)
(c)
(d)

Figure 8. The multi-order CCA pattern construction process of DMS^k. (**a**) The first-order CCA of NS$_1$–NS$_8$; (**b**) The multi-order CCA of NS$_1$; (**c**) The multi-order CCA of NS$_8$; (**d**) The multi-order CCA pattern of DMS^k.

Finally, the storage format of the guide star catalog, including the LUT and SPD, is shown in Figure 9. The LUT is stored in an increasing order of MST, and the multi-order CCA of each DMS^k is stored in the DNS$_i^k$ counterclockwise order. The storage content mainly includes the star number (k), right ascension (α_k), declination (β_k), and magnitude of stars. In addition, MST and CCA patterns for each star are included, which are stored in LUTs and SPDs, respectively. Among them, each row in the LUT corresponds to a main star, which is used to rapidly search for candidate DMSs; the SPD stores the CCA pattern matrix of each star, and the meaning of its row and column has been indicated in Figure 9, which is used to realize the determination of the unique DMS and the global SNSs identification.

3.2. Star Identification Scheme

To identify the SMS and SNSs rapidly and accurately after screening by the DEQ method, this study proposes a two-step identification method. First, the star coordinates in the captured image are extracted and shifted to the image center, and the SNSs are selected using the DEQ method. Then, the selected guide stars are used to generate the MST pattern and combined with the K vector algorithm to establish an indexing mechanism for the rapid locating of DMS candidates from the LUT. Next, the correct SMS and its SNSs are identified from the DMS candidate set by voting using the Jaccard similarity coefficients of each order CCA pattern. Finally, the identified stars are verified to eliminate mismatched stars. The identification framework is shown in Figure 10.

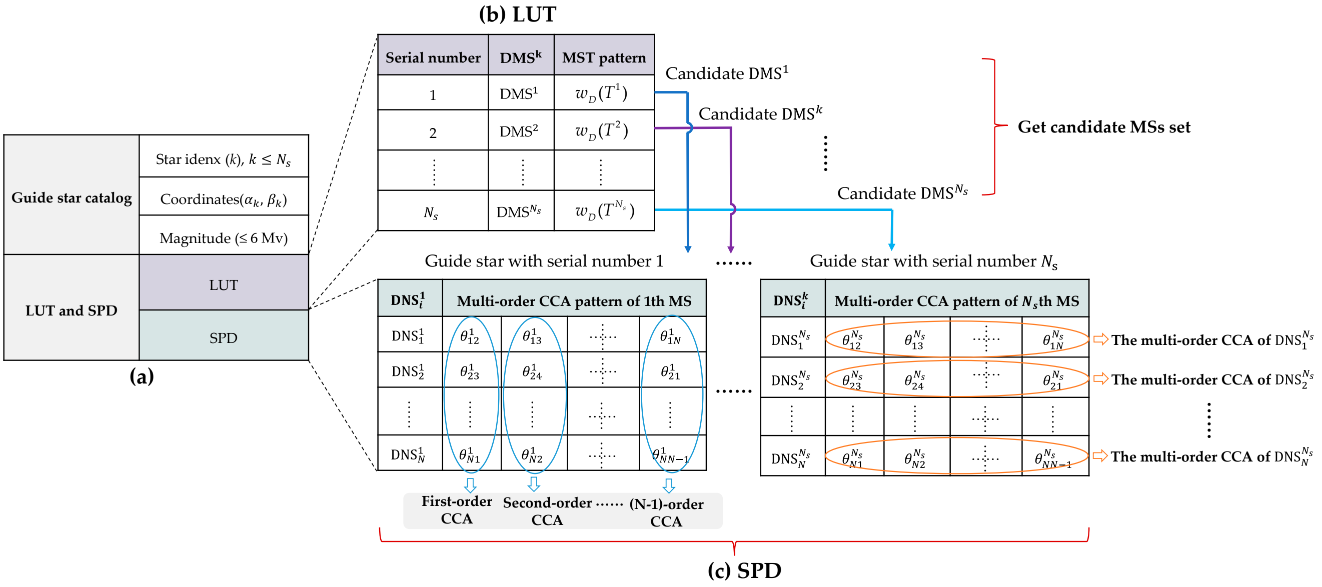


Figure 9. The storage format of the guide star catalog, including the LUT and SPD. (**a**) Parameters stored in the guide star catalog; (**b**) The LUT storage format used to obtain the correct DMS set; (**c**) The SPD storage format used to obtain the correct DMS^k and identify SNS$_i^k$.

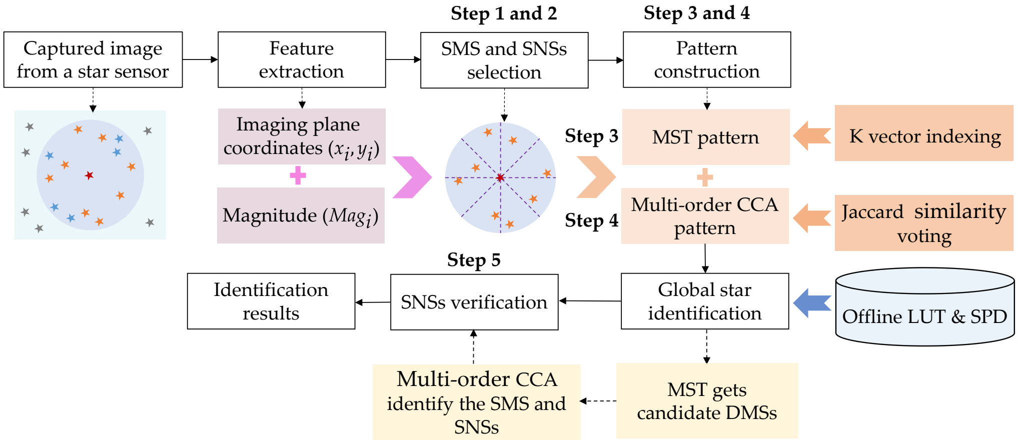


Figure 10. The overall identification framework.

The specific steps of the identification algorithm are as follows:

Step 1: SMS selection

In real scenes, it cannot be guaranteed that there is a guide star in every FOV center. Therefore, when a star image frame is to be identified and captured, the centroid extraction algorithm is used first to calculate the 2D coordinates of each star in the FOV. Then, the Euclidean distance between them and the image center (x_c, y_c) is calculated. Finally, the star with the smallest value is selected as an SMS and denoted by (x_m, y_m).

Step 2: SNS selection for SMS

The DEQ method is employed to perform SNSs selection. First, the 2D coordinate deviations between (x_c, y_c) and (x_m, y_m) in the x and y dimensions are calculated and denoted by $\Delta x = x_m - x_c$ and $\Delta y = y_m - y_c$, respectively. Then, translate (x_m, y_m) to coincide with (x_c, y_c) following the translation rule of:

$$\begin{cases} x'_m = x_m - \Delta x \\ y'_m = y_m - \Delta y \end{cases} . \tag{15}$$

The coordinates (x_s, y_s) of other stars in the FOV are also translated according to the rule defined by Equation (15), and stars whose coordinate value is greater than PR_{max} after translation are eliminated. Next, the same parameters, PR_d, N_{max}, and g, are used to define a DEQ SNS selection rule, and the confidence factor λ_i of each star in each quadrant is calculated by Equation (9); the star with the maximum λ_i value in each quadrant is selected as an SNS. Finally, eight SNSs with a relatively uniform distribution and high confidence are selected.

Step 3: MST fast index definition and candidate DMS search

Similarly, regard the nine stars, including SMS, as graph vertices and select a starting vertex randomly; next, use the Prim algorithm to construct an MST pattern for SMS until all vertices are involved in the tree denoted by $w_S(T)$. Due to the rounding errors in the tree construction process, the initial threshold D_{th} and the dynamic change parameter ΔD are used to search for the candidate DMS from the LUT roughly and rapidly. The DMS that meets the following conditions will be stored in the candidate DMS set Ω_{candi}:

$$\begin{aligned} &\Omega_{candi} = \left\{\mathrm{DMS}^k, (k = 1, \ldots, N_{candi}) | w_S(T) - D_{th} \le w_D(T) \le w_S(T) + D_{th}\right\}, \\ &s.t.\ D_{th} = D_{th} + l \cdot \Delta\mathrm{D}, l = l + 1, l \ge 0 \end{aligned} \tag{16}$$

where l is the iteration number, which is used to increase the threshold interval when the correct DMS is not involved in Ω_{candi}, and N_{candi} is the number of candidate DMSs.

The advantage of the dynamic threshold is that it not only effectively avoids a long search time under the high threshold but also reduces the probability of missing the correct candidate star due to the too-small threshold. In this way, the search efficiency is significantly improved, while the length of Ω_{candi} is effectively reduced. In this study, set D_{th} = 1 pixel and ΔD = 16 pixels. This is because the maximum position noise in real scenes is approximately 1 pixel; in this case, the maximum MST pattern difference caused by the position noise is about 16 pixels.

Simultaneously, to increase the search efficiency, using the K vector algorithm to create a fast index mechanism, rapidly locate the starting and ending candidate DMS intervals. First, calculate the index value x_{star} and x_{end} as follows:

$$\begin{cases} x_{start} = \frac{w_S(T) - D_{th} - a_0}{a_1}, \\ x_{end} = \frac{w_S(T) + D_{th} - a_0}{a_1}. \end{cases} \tag{17}$$

Then, substitute the index value into the $\boldsymbol{K}$ vector function, Equation (14), to obtain the serial number (k_{start}, k_{end}) of the candidate DMSs in the LUT. The obtained row corresponds to a candidate DMS; finally, store them in Ω_{candi}.Step 4: Multi-order CCA pattern construction for SMS and SNS identification

First, randomly select a star as a starting star and use Equations (4)–(6) to obtain the multi-order CCA pattern of the captured star image. If the number of candidate DMSs N_{candi} is less than one, a dynamic threshold will adjust the range to obtain a new set of candidate DMSs; if $N_{candi} = 1$, the candidate DMS is considered to be the correct MS; otherwise, the Jackard similarity coefficient of the multi-order CCA (i.e., each column of the matrix $\boldsymbol{A}_{capt}[\mathrm{z}]$ and $\boldsymbol{A}^k_{candi}[\mathrm{z}]$) between the SMS and the candidate DMSs in Ω_{candi} is calculated as follows:

$$\begin{aligned} &\boldsymbol{J}^k_{sim}[z] = Jackard(\boldsymbol{A}_{capt}[z], \boldsymbol{A}^k_{candi}[z]) = \frac{\left|\boldsymbol{A}_{capt}[z] \cap \boldsymbol{A}^k_{candi}[z]\right|}{\left|\boldsymbol{A}_{capt}[z] \cup \boldsymbol{A}^k_{candi}[z]\right|}, \\ &s.t.\ z = 1, \ldots, N-1, k = 1, \ldots, N_{candi} \end{aligned} \tag{18}$$

where $\boldsymbol{J}^k_{\mathrm{sim}}[\mathrm{z}]$ represents the zth order CCA Jackard similarity coefficient between the SMS and the kth candidate DMS; $\boldsymbol{A}_{capt}[\mathrm{z}]$ and $\boldsymbol{A}^k_{candi}[\mathrm{z}]$ are the zth order CCA of the SMS and the zth order CCA of the kth candidate DMS, respectively.

The Jaccard similarity offers a significant advantage in that it effectively disregards any order differences between DNS and SNS by treating them as two unordered sets, eliminating the step of determining the reference star or edge in most pattern-based algorithms. This is also the motivation for choosing it as a similarity measure.

Further, a repeated voting strategy is employed to identify the unique MS. Equation (18) is used to calculate the CCA Jackard similarities of the first to (*N*− 1)th order, and the number of votes for the candidate DMSs with the maximum similarity in each order is increased by one. After the (*N*− 1)th order CCAs are calculated, the star with the largest number of repeated votes is denoted as the correct DMS, which can be expressed as follows:

$$\begin{aligned} &\underset{\mathrm{DMS}^{k}_{candi} \in \boldsymbol{\Omega}_{candi}}{\operatorname{argmax}} \sum_{z=1}^{N-1} Vote\left(\mathrm{DMS}^{k}_{candi}\right) = \left\{ \, k \middle| max(\boldsymbol{J}^{k}_{sim}[\mathrm{z}]) \, \right\}, \\ &s.t. \quad 1 < k < N_{candi} \end{aligned} \tag{19}$$

where function $Vote(\cdot)$ adds one to the number of votes, and function $max(\cdot)$ returns the maximum value.

Finally, the global identification of SNSs is performed using the multi-order CCA pattern of each SNS (i.e., each row of $\boldsymbol{A}_{capt}$). The specific process is as follows. First, an SNS is randomly selected from the captured image, and the CCA pattern of the selected SNS is denoted as $\boldsymbol{A}_{capt}[i]$ (1 < *i* < *N*). Next, the CCA pattern of the correctly identified MS is recorded as $\boldsymbol{A}^{k}_{corr}$. Then, each row of the *N* × (*N*− 1) matrix $\boldsymbol{A}^{k}_{corr}$ is subtracted from the 1 × (*N*− 1) vector $\boldsymbol{A}_{capt}[i]$ to obtain the difference matrix $\boldsymbol{A}_{diff}$, which can be expressed by:

$$\begin{aligned} &\boldsymbol{A}_{diff} = \boldsymbol{A}^{k}_{corr} - \boldsymbol{A}_{capt}[i], \\ &s.t. \; 1 < i < N \end{aligned} \tag{20}$$

where *k* is the serial number of the correctly identified MS.

Next, the number of elements less than η in each row in $\boldsymbol{A}_{diff}$ is obtained by:

$$\begin{aligned} &\boldsymbol{N}_{zeros}[i] = Count(\boldsymbol{A}_{diff}[i] < \eta), \\ &s.t. \; i = 1, \ldots, N \end{aligned} \tag{21}$$

where η is an infinitesimal number close to zero; function $Count(\cdot)$ is used to determine the number of zeros in each row; the row with the largest number of elements close to zero is considered the guide star identity ID_{alig} corresponding to the aforementioned randomly selected SNS, and it is called the alignment star and is obtained by:

$$ID_{alig} = \underset{i=1,\ldots,N}{\operatorname{argmax}} \, \boldsymbol{N}_{zeros}[i]. \tag{22}$$

As mentioned in Section 2.2, since the rotation of the captured star image changes only the row sequence of the CCA pattern matrix of the SMS ($\boldsymbol{A}^{k}_{corr}$), after determining the ID_{alig} of the correct DMS, it only needs to be shifted cyclically to achieve an $\mathrm{NS}_{ID_{alig}}$ alignment in the matrix of $\boldsymbol{A}^{k}_{corr}$ and $\boldsymbol{A}_{capt}$. In this way, SNS corresponds to the DNS of the correct DMS one by one, so the identity of other SNSs can be identified.

Step 5: SNSs verification

First, compare the difference in multi-order CCA between the $\mathrm{DNS}_{ID_{alig}}$ of the correct DMS ($\boldsymbol{A}^{k}_{corr}\left[ID_{alig}\right]$) and the alignment star in the captured image ($\boldsymbol{A}_{capt}\left[ID_{alig}\right]$) as follows:

$$[\boldsymbol{D}_{v}, \boldsymbol{Ind}] = \boldsymbol{A}^{k}_{corr}[ID_{alig}] - \boldsymbol{A}_{capt}[ID_{alig}] \, , \tag{23}$$

where $\boldsymbol{D}_{v}$ is the difference vector in the identified alignment star between the captured image and the SPD, and ***ind*** denotes the SNSs index vector; if there are elements larger than one in $\boldsymbol{D}_{v}$, they are considered misidentified neighboring stars and are eliminated by the index vector ***Ind***.

In general, if three or more stars within the FOV are identified, and neither false stars nor correct stars are misidentified, the identification of the current frame is considered successful.

4. Experimental Results

Compared with most pattern-based algorithms, the main advantage of the proposed algorithm is that it does not rely on the reference star, which improves algorithm robustness against noise significantly. The effectiveness of the proposed algorithm was verified by experimental tests on simulated and real captured star images using the star sensor, whose parameters are listed in Table 3.

Table 3. The main parameters of the star sensor used in the experimental verification.

Parameters	Value
FOV ($FOV_x \times FOV_y$)	20° × 20°
Imaging plane ($D_x \times D_y$)	1536 × 1536 (pixels)
Single-pixel size ($p_x \times p_y$)	0.0055 × 0.0055 (mm)
Focal length (f)	24.03 (mm)

At present, most star identification algorithms use the ED method or the BR method to select NSs, but they both have certain disadvantages in real applications. Among them, the ED method is more significantly affected by position noise and false star noise, whereas the BR method is extremely sensitive to magnitude noise, which may result in wrong NS selection. In this study, combining the advantages of the ED and BR methods, a confidence factor (Equation (9)) is defined for stars, and the DEQ method is proposed for NS selection. Therefore, to illustrate the advantages of the proposed DEQ method better, this study compares the identification accuracy of the three methods under the effect of position noise, false star noise, and magnitude noise.

Then, the identification accuracy and speed of the proposed algorithm were compared with the corresponding performances of five state-of-the-art algorithms. In the experiments, the Monte Carlo method was employed to generate 15,000 simulated star images; also, all guide stars were projected separately to the center of the FOV, and an additional 4956 simulated star images were obtained, which included the scenes where every guide star was MS. Finally, 199, 546 simulated star images were obtained ($N_{\mathrm{T}}= 199,546$) and used as a test dataset. The CPU used the Core i7-6500U manufactured by Intel in Santa Clara, California, USA, with a main frequency of 2.5 GHz and a memory of 12 GB, and the program was written in Matlab2018a.

4.1. Comparison Results of Different Star Selection Strategies

First, the tests were performed in the absence of noise interference, and the results indicated that the three methods performed well, achieving similar identification accuracy values, particularly ED (99.95%), BR (99.94%), and DEQ (99.95%).

Then, the three methods' performances were tested under positional noise. In [36], the authors pointed out that the maximum position deviation in real scenes is approximately 1 pixel. Therefore, a Gaussian position noise with a mean of zero and a standard deviation of 0.1–1 pixels was used in the experiment, and the obtained experimental results are presented in Figure 11a. As shown in Figure 11a, when the position noise was 0–0.2 pixels, the identification accuracy of the three methods was maintained at a relatively high level; it was higher than 99%. When the position noise reached the value of 1 pixel, the identification accuracy of the ED method decreased significantly to 97.56%, but the BR and DEQ methods could maintain high accuracy in this case. The results indicated that the proposed DEQ method was robust to position noise.

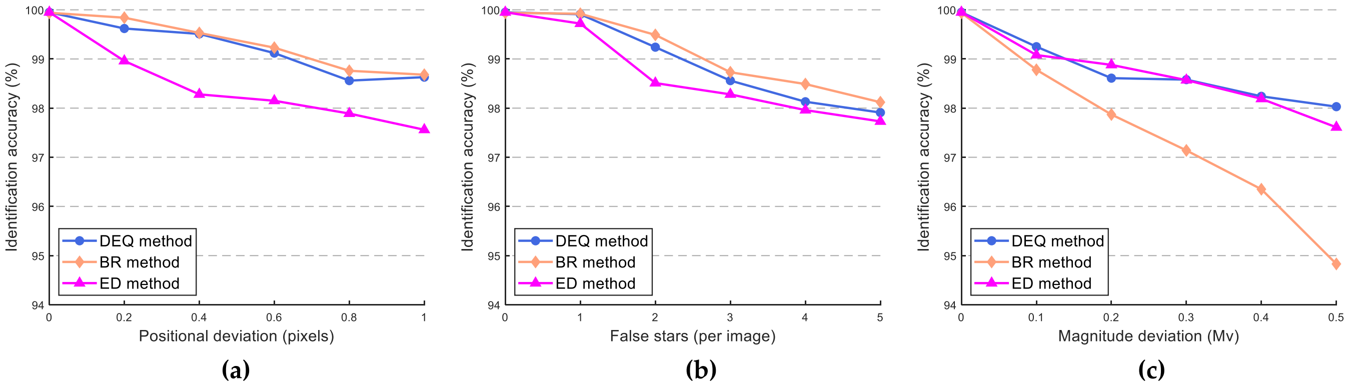


Figure 11. The identification accuracy results of the three star selection methods under the effect of three types of noises. (**a**) Scenes with a deviation of 0–1 pixels in positional noise; (**b**) Scenes with 0–5 false stars; (**c**) Scenes with a deviation of 0–0.5 in magnitude noise.

The comparison results of the three methods obtained under the effect of the false star noise are presented in Figure 11b, where it can be seen that when the number of false stars was one, the identification accuracy values of the three methods were basically the same. However, when the number of false stars increased, the identification accuracy values of the DEQ and BR were less affected than that of the ED method. This was because the ED method used only the Euclidean distance for NS selection, so the probability of selecting false stars closer to the MS was high, which affected the identification accuracy.

Finally, magnitude noise with a mean value of zero and a standard deviation of 0–0.5 was used to test the three methods' accuracy, as shown in Figure 11c. The results indicated that with the increase in magnitude noise, the identification accuracy of the BR method obviously decreased, and when the magnitude noise reached the value of 0.5, its accuracy decreased to 94.83%. This was because this method relied only on the magnitude in the NS selection, so the probability of a wrong NS selection increased with the magnitude noise. The ED method achieved better performance than the BR method. Namely, when the magnitude noise reached the value of 0.5, the identification accuracy of the ED method was still high, reaching 97.61%. Finally, the proposed DEQ method defined the confidence of each NS and thus could achieve a high identification accuracy of 98.03% under the 0.5 magnitude noise.

Consequently, it can be concluded that the proposed DEQ method has good robustness against different noises in star identification.

4.2. Comparison and Analysis Results of Identification Algorithms

Next, the identification accuracy, memory, and running time consumption of the proposed algorithm and five state-of-the-art algorithms were compared. The five algorithms used for comparison included: the optimized grid pattern algorithm (OGP) [19], radial and cyclic pattern algorithm (RCP) [20], geometric voting algorithm (GMV) [14], pyramid algorithm [15], and discrete HMM-Based algorithm (HMM) [37]. These selected comparison algorithms were mainly pattern-based and subgraph-based, and all claim to have achieved good performance. The reason why the learning-based algorithm was not selected is that it did not seem to have reported any examples of on-orbit applications.

In this study, the identification accuracy was defined as follows:

$$Accuracy = \frac{N_T - N_E}{N_T} \times 100\%, \tag{24}$$

where N_{T} is the number of stars in the test dataset, and N_E is the number of identification failures.

4.2.1. Identification Accuracy Comparison

First, the test was performed in an ideal scene without any noise, and the obtained identification accuracy results of the six algorithms are shown in Table 4. Next, the accuracy comparison experiment was conducted under three types of noises, in turn, and the results are shown in Table 4, where AIA represents the average identification accuracy, and MIA represents the identification accuracy under the maximum noise interference scenes.

Table 4. Comparison results of the identification accuracy of the six algorithms in different scenes.

	No Noise	Position Noise		False Star Noise		Magnitude Noise	
	AIA(%)	**AIA(%)**	**MIA(%)**	**AIA(%)**	**MIA(%)**	**AIA(%)**	**MIA(%)**
Proposed	99.95	99.23	98.63	98.95	97.91	98.95	98.03
OGP	98.63	98.01	97.39	95.92	91.29	95.92	96.42
RCP	97.59	95.09	92.42	89.63	80.18	89.63	92.17
GMV	97.54	93.30	88.35	88.58	76.46	88.58	93.29
Pyramid	99.12	97.53	94.89	97.30	95.33	97.30	96.84
HMM	98.72	98.22	97.81	96.89	94.26	98.10	97.13

(1) Positional noise effect

The comparison results of the identification accuracy of the six algorithms under the position noise are presented in Figure 12. In the experiment, the added noise was the Gaussian noise with a mean value of zero and a standard deviation of 0.1–1 pixels. When the position noise's standard deviation was in the range of 0.1–0.2, the identification accuracy of all algorithms was greater than 96%. When the standard deviation of the position noise increased to 0.8, the accuracy values of the RCP, GMV, and Pyramid algorithms decreased to 93.61%, 90.64%, and 96.96%, respectively. The OGP, HMM, and proposed algorithms could still maintain high accuracy. When the standard deviation of the position noise increased to one, the identification accuracy of the RCP and GMV algorithms decreased significantly to 92.42% and 88.35%, respectively. This was because the GMV algorithm used only the distance voting scores of NSs for identification, while the RCP algorithm had deviations in binary codes in radial and circular directions due to star position deviations. However, the identification accuracy of the proposed algorithm was still high, above 98.5%. This was due to the dynamic threshold used in the proposed algorithm, and the constructed MST and CCA could make full use of the global characteristics of NSs. Therefore, the proposed algorithm showed high robustness against the position noise.

(2) False star noise effect

The impact of a different number of false stars on the identification accuracy of the six algorithms is presented in Figure 13. In this experiment, the number of false stars was set from zero to five. When the number of false stars reached the value of five, the identification accuracy of the RCP algorithm decreased to 80.18%. This was because the false stars made the wrong selection of the minimum central angle, which resulted in a failure of the cycle pattern. Due to the increase in the number of false stars appearing in the grid, the accuracy of the OGP algorithm was reduced to 91.29%. Namely, the erroneous voting of false stars significantly affected the identification accuracy of the GMV algorithm, reducing its accuracy to 76.46%. The HMM eigenvectors were disturbed by false stars, causing matching errors, so the identification accuracy of the HMM algorithm decreased to 94.26%. The proposed algorithm was based on the DEQ method, using the defined confidence factor of the NSs, which could effectively suppress the interference of false stars. In addition, the global influence of false stars on the graph was not great, and the topology structure of a multi-order CCA pattern could also accurately eliminate the interference of false stars. Therefore, even in scenes with false stars, the proposed algorithm could achieve a high identification accuracy.

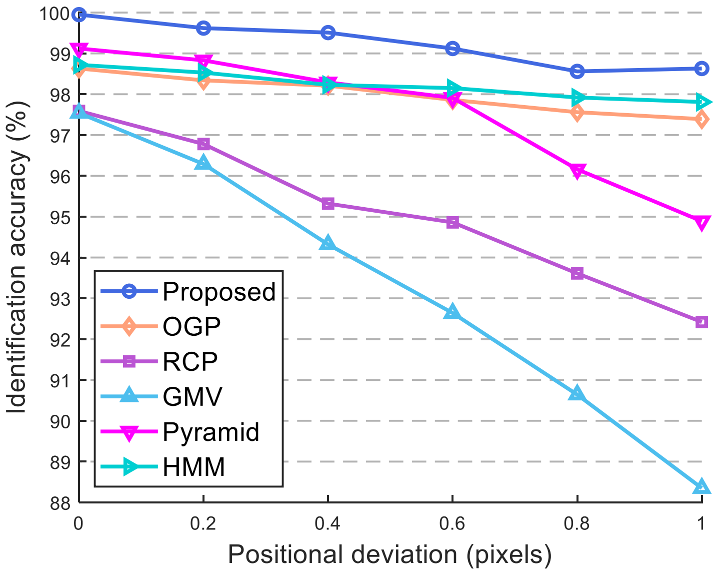


Figure 12. The identification accuracy results of the six algorithms under the positional noise.

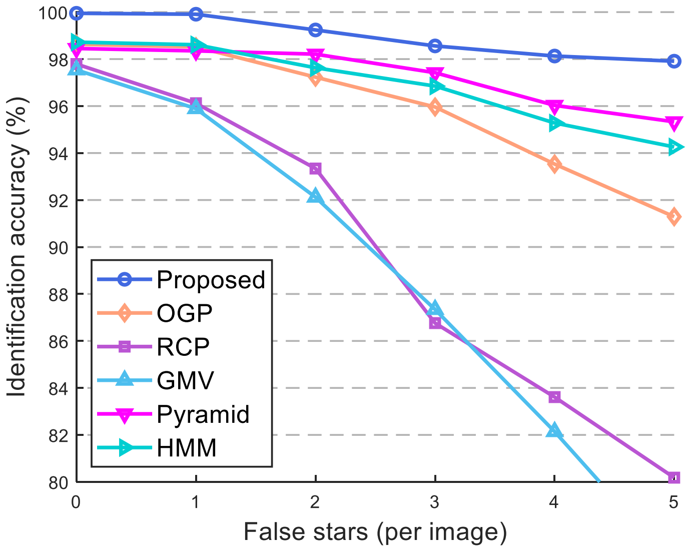


Figure 13. The identification accuracy results of the six algorithms under the false star noise.

(3) Magnitude noise effect

The comparison results of the identification accuracy of the six algorithms under the deviation of 0–0.5 caused by the magnitude noise are presented in Figure 14. As the deviation increased from 0 to 0.5, the identification accuracy of the proposed algorithm decreased from 99.95% to 98.03%. When the magnitude noise reached 0.5, the accuracy of the proposed algorithm was 4.74%, 5.86%, 1.61%, 1.19%, and 0.9% higher than the accuracy of the GMV, RCP, OGP, Pyramid, and HMM algorithms, respectively. This was because the GMV algorithm relied on the votes of NSs, and the circular pattern of the RCP algorithm depended on the starting edge. However, the magnitude noise led to the loss of NSs and caused the starting edge selection error. Namely, due to the influence of the magnitude noise, the guide star was missing in the OGP grid cells. Therefore, the GMV, RCP, and OGP algorithms were sensitive to magnitude noise. However, the proposed algorithm did not need to determine the starting star like the RCP algorithm, and both patterns selected the starting star randomly and considered the global characteristics between stars, thus effectively reducing the interference caused by missing stars. Thus, the proposed algorithm had a high tolerance to the magnitude noise.

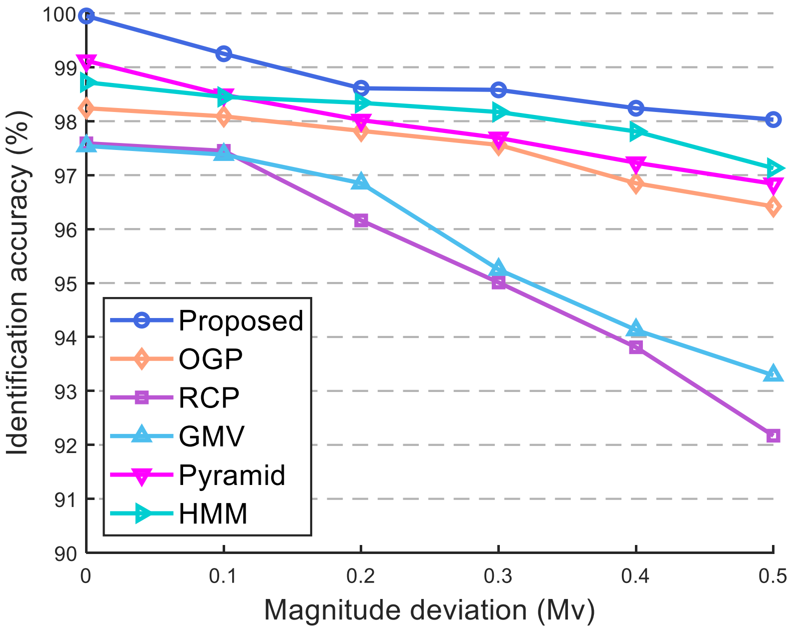


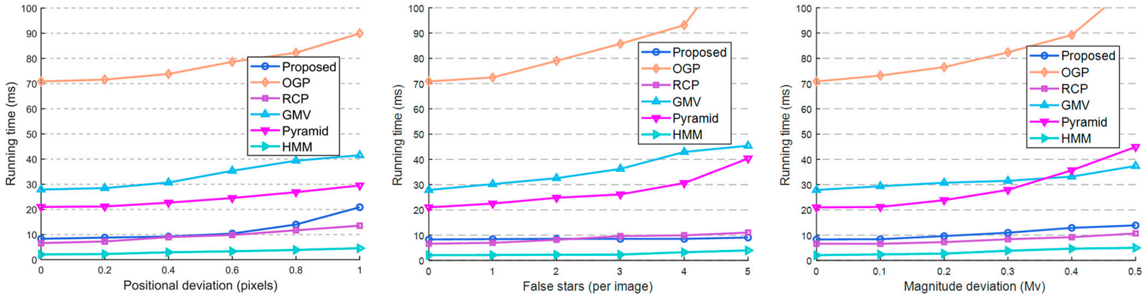
Figure 14. The identification accuracy results of the six algorithms under magnitude noise.

4.2.2. Memory and Running Time Test

The running time of a star identification algorithm mainly refers to the time consumed from the centroid extraction of a star image to achieve the correct identification of stars in the FOV. The average running time (AIT) results obtained under no-noise and different noisy conditions and the running time values obtained at the maximum noise (MIT) are presented in Table 5. The running time of the OGP algorithm was relatively long; this was because of a relatively complex optimization process of the OGP algorithm. Thus, to ensure identification accuracy, it is necessary to sacrifice the running time. The HMM algorithm had the shortest running time among all algorithms, which was due to the simplicity of its pattern and identification process. The running time of the RCP algorithm was shorter than those of the OGP, GMV, and Pyramid algorithms, but as the noise level increased, the RCP algorithm required performing multi-step identification to realize the star identification, which increased the running time significantly. The running time of the proposed algorithm changed slightly under the influence of magnitude noise and false star noise. Under the position noise, to ensure high identification accuracy, the dynamic threshold search process became longer, and the running time increased accordingly, but this increase could be considered acceptable. The running time comparison results of different algorithms obtained under three types of noises are shown in Figure 15. The memory usage of a star identification algorithm is mainly defined by the LUT and SPD. The comparison results of the memory usage of different algorithms are shown in Table 5. The results demonstrated that the proposed algorithm with $N_{\max} = 8$ was superior to most mainstream algorithms in terms of memory and running time consumption.

Table 5. Comparison results of the identification accuracy of the six algorithms in different scenes.

	Memory (KB)	Position Noise		False Star Noise		Magnitude Noise	
		AIT (ms)	MIT (ms)	AIT (ms)	MIT (ms)	AIT (ms)	MIT (ms)
Proposed	1103.48	11.92	20.87	8.56	9.05	10.68	13.87
OGP	1638.43	77.86	89.95	83.29	126.78	81.41	110.36
RCP	573.44	9.65	13.53	8.71	11.03	8.12	10.68
GMV	1863.68	33.72	41.52	35.87	45.39	31.68	37.39
Pyramid	2068.48	24.75	29.49	27.64	10.35	29.09	44.92
HMM	2283.52	3.18	4.56	2.66	3.96	3.44	4.98

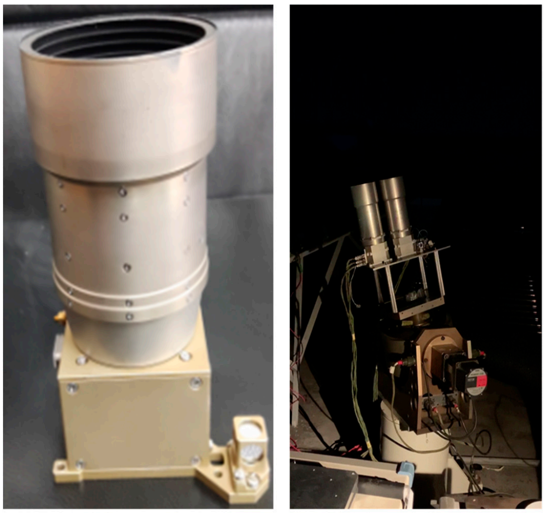


(a) (b) (c)

Figure 15. The running time results of different star identification algorithms under three types of noises. (**a**) Scenes with a deviation of 0–1 pixels in positional noise; (**b**) Scenes with 0–5 false stars; (**c**) Scenes with a deviation of 0–0.5 in magnitude noise.

4.3. Real Star Image Test

The performance of the proposed algorithm was verified by real-scene experiments using the real star images collected by a star sensor, as shown in Figure 16a. The images were collected in Lijiang City, Yunnan Province, China; this area was selected because of good weather and thin clouds, which could better simulate actual space scenes. Under the rotation of the three-axis turntable, a total of 3128 real star images were collected, as illustrated in Figure 16b. In this experiment, the key parameters of the star sensor were set, as shown in Table 3. An example of an identification result record of a one-frame real star image is shown in Table 6, where the star catalog serial number of the SMS is 233.



(a) (b)

Figure 16. Real star image experiments. (**a**) Star sensor used in the experiments; (**b**) Picture of the star sensor placed on the three-axis turntable to collect real star images.

Table 6. The identification results of the one-frame real star image.

Scheme	Centroids		Guide Star Catalog			
	x (pixel)	y (pixel)	Serial Number in Star Catalog	Right Ascension (deg)	Declination (deg)	Star Magnitude
SMS	710.68	840.13	233	20.8790	−30.9456	5.83
SNS$_1$	1052.82	891.73	177	15.6101	−31.5520	5.51
SNS$_2$	1048.15	1458.62	176	15.3261	−38.9165	5.59
SNS$_3$	569.81	1295.78	256	22.2336	−36.8652	5.50
SNS$_4$	411.44	954.22	288	25.5358	−32.3270	5.26
SNS$_5$	501.81	765.63	265	24.0355	−29.9073	5.70
SNS$_6$	590.52	481.08	251	22.5954	−26.2079	5.93
SNS$_7$	575.54	792.00	255	22.9301	−30.28	5.79
SNS$_8$	1123.05	727.28	174	14.6515	−29.3574	4.31

5. Discussion

Star identification is to find the correspondence between a captured star image and the pre-stored guide star catalog and obtain the identity of the guide star in the FOV in preparation for the high-precision spacecraft attitude measurement. However, most existing algorithms perform poorly in terms of identification accuracy and running time under noisy conditions, which has been the main motivation of this study for developing an efficient star identification algorithm that can perform well under different noisy conditions. The proposed algorithm was compared with five state-of-the-art algorithms under different conditions, and the results have indicated that the proposed algorithm has certain advantages over the existing algorithm, which could be summarized as follows:

1. The confidence factor is defined for each guide star, and the proposed DEQ method is used to select NSs. Compared with the BR and ED methods, in the proposed method, the probability of wrong NS selection is reduced, and the selected NSs are more evenly distributed, which increases pattern identifiability;
2. Compared with the existing pattern-based algorithms, neither the MST nor the CCA pattern in the proposed algorithm depends on the reference star or edge. Therefore, the proposed algorithm is more robust to noise than the existing algorithms;
3. The MST and CCA patterns fully consider the global characteristics of stars, thus significantly enhancing the anti-noise ability of the identification algorithm. In addition, the combination of MST and the K vector can significantly improve the efficiency of the candidate DMS search. Moreover, the dynamic threshold adopted in the proposed algorithm not only ensures the identification accuracy but also reduces the interval search time. The CCA pattern requires determining only one NS, and then it completes the global identification of other NSs by simply shifting to achieve an alignment.

Despite the aforementioned advantages, the proposed algorithm has certain limitations, which are as follows:

1. The proposed DEQ method can be used only in combination with a large-FOV star sensor. Namely, under the small-FOV conditions, with the increase in detection capability, the number of faint stars in the FOV increases significantly, and the confidence discrimination of guide stars decreases, resulting in an NS selection error. In the future, we will try to build a virtual FOV through multi-frame star images stitching and combine the incomplete star catalog to make it also suitable for small-FOV.
2. The MST pattern is constructed based on the Euclidean distance between stars, but it is susceptible to positional noise. To address this shortcoming, in the proposed algorithm, an iterative dynamic threshold interval is defined to ensure identification accuracy, but this can be achieved at the cost of an increase in the running time. In

the future, we will try to use optimization methods to determine an optimal static threshold to eliminate the running time of iterations.

6. Conclusions

The practical significance of the proposed algorithm is that when the star sensor is in the LIS mode, the algorithm can identify stars accurately and rapidly and provide accurate attitude information for the satellite.

In this study, a global star identification algorithm based on the MST index and multi-order CCA is developed using the Euclidean distance and star angle pattern with a rotation invariance. First, aiming at the shortcomings of the BR and ED methods in neighboring star selection, a confidence factor is defined for stars. The proposed DEQ method can more effectively reduce the loss and mis-selection rate of neighboring stars caused by different noises than the BR and ED methods. In addition, the starting star vertex is randomly selected, and the Prim algorithm is employed to construct the MST pattern for each main star and combine it with the K vector algorithm; a dynamic threshold interval matching method is used to search for main star candidates rapidly. Finally, the Jackard similarity coefficients of the multi-order CCA (with $1 - (N - 1)$ columns in each matrix) between the captured image and main star candidates are calculated, and a voting strategy is used to identify the correct main star. Simultaneously, the neighboring stars are identified by using a multi-order CCA ($1 - N$ rows in the matrix) cyclic shift of the neighboring star of the correct main star. The robustness and rapidity of the proposed algorithm are verified by comparison experiments with five state-of-the-art algorithms, considering the identification accuracy and running time under position noise, false star noise, and magnitude noise. Moreover, the experimental results obtained on 3128 real star images verify the excellent performance of the proposed algorithm.

In the future, combined with the research of the high-precision centroid extraction algorithm, the performance of the algorithm under high dynamic conditions should be verified; in addition, researchers should continue to optimize the proposed algorithm and complete the on-orbit applications.

Author Contributions: Conceptualization, Z.Z. (Zijian Zhu) and Y.M.; methodology, Z.Z. (Zijian Zhu); software, Z.Z. (Zijian Zhu); validation, Z.Z. (Zijian Zhu), B.D., Z.Z. (Zifa Zhu) and J.Y.; investigation, Z.Z. (Zijian Zhu) and R.Z.; resources, E.L.; writing—original draft, Z.Z. (Zijian Zhu); writing—review and editing, Y.M. and R.Z.; visualization, Z.Z. (Zijian Zhu), B.D. and Y.T.; supervision, E.L.; project administration, R.Z.; funding acquisition, R.Z. and Y.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Department of Science and Technology of Sichuan Province (No. 2022JDRC0065) and the West Light Foundation of the Chinese Academy of Sciences (No. YA21K001).

Data Availability Statement: The data presented in this paper are not publicly available at this time but may be obtained from the authors upon reasonable request.

Acknowledgments: We thank the researchers who cooperated with us at the Institute of Optics and Electronics, Chinese Academy of Sciences. Meanwhile, we would like to thank the reviewers for their valuable comments.

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

Appendix A

Table A1. Definition of the abbreviations used in this study.

Abbreviations	Definitions
DMS/SMS	Main star in database/image
DNS/SNS	Neighboring star of DMS/SMS in database/image
LIS	Lost-in-space
LUT	Look-up table
SPD	Star pattern database
ED	Euclidean distance method used to screen DNS/SNS for DMS/SMS
BR	Brightness method used to screen DNS/SNS for DMS/SMS
DEQ	Proposed method used to screen DNS/SNS for DMS/SMS
MST	Maximum spanning tree
CCA	Continuous cycle angle
AIT	Average running time
MIT	Running time under the maximum noise
AIA	Average identification accuracy
MIA	Identification accuracy under the maximum noise

References

1. Samirbhai, M.D. A Reliable and Fast Lost-in-Space Mode Star Tracker. Ph.D. Dissertation, Nanyang Technological University, Singapore, 2019.
2. Du, J.; Wei, X.; Li, J.; Wang, G.; Zang, C. Star identification based on radial triangle mapping Matrix. *IEEE Sens. J.* **2022**, *22*, 8795–8807. [CrossRef]
3. Pham, M.D.; Low, K.-S.; Chen, S. An autonomous star recognition algorithm with optimized database. *IEEE Trans. Aerosp. Electron. Syst.* **2013**, *49*, 1467–1475. [CrossRef]
4. Jiang, J.; Liu, L.; Zhang, G. Star identification based on spider-web image and hierarchical CNN. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *56*, 3055–3062. [CrossRef]
5. Lu, K.; Liu, E.; Zhao, R.; Tian, H.; Zhang, H. A Fast Star Identification Algorithm of Star Sensors in the LIS Mode. *Remote Sens.* **2022**, *14*, 1739. [CrossRef]
6. Mehta, D.S.; Chen, S.; Low, K.S. A robust star identification algorithm with star shortlisting. *Adv. Space Res.* **2018**, *61*, 2647–2660. [CrossRef]
7. Wang, G.; Li, J.; Wei, X. Star identification based on hash map. *IEEE Sens. J.* **2017**, *18*, 1591–1599. [CrossRef]
8. Li, J.; Wei, X.; Zhang, G. Iterative algorithm for autonomous star identification. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 536–547. [CrossRef]
9. 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]
10. Wang, H.; Wang, Z.-y.; Wang, B.-d.; Yu, Z.-q.; Jin, Z.-h.; Crassidis, J.L. An artificial intelligence enhanced star identification algorithm. *Front. Inf. Technol. Electron. Eng.* **2020**, *21*, 1661–1670. [CrossRef]
11. Zhang, G. *Star Identification: Methods, Techniques and Algorithms*; Springer: Berlin/Heidelberg, Germany, 2016.
12. Sun, L.; Zhou, Y. Mvdt-si: A multi-view double-triangle algorithm for star identification. *Sensors* **2020**, *20*, 3027. [CrossRef]
13. Liu, M.; Wei, X.; Wen, D.; Wang, H. Star identification based on multilayer voting algorithm for star sensors. *Sensors* **2021**, *21*, 3084. [CrossRef] [PubMed]
14. Kolomenkin, M.; Pollak, S.; Shimshoni, I.; Lindenbaum, M. Geometric voting algorithm for star trackers. *IEEE Trans. Aerosp. Electron. Syst.* **2008**, *44*, 441–456. [CrossRef]
15. Mortari, D.; Junkins, J.L.; Samaan, M. Lost-in-space pyramid algorithm for robust star pattern recognition. In Proceedings of the Guidance and Control 2001, Breckenridge, CO, USA, 31 January–4 February 2001; pp. 49–68.
16. Cole, C.L.; Crassidis, J.L. Fast star-pattern recognition using planar triangles. *J. Guid. Control Dyn.* **2006**, *29*, 64–71. [CrossRef]
17. Lee, H.; Bang, H. Star pattern identification technique by modified grid algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2007**, *43*, 1112–1116.
18. 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* **2019**, *8*, 1010–1020. [CrossRef]
19. Aghaei, M.; Moghaddam, H.A. Grid star identification improvement using optimization approaches. *IEEE Trans. Aerosp. Electron. Syst.* **2016**, *52*, 2080–2090. [CrossRef]
20. Zhang, G.; Wei, X.; Jiang, J. Full-sky autonomous star identification based on radial and cyclic features of star pattern. *Image Vis. Comput.* **2008**, *26*, 891–897. [CrossRef]

21. Mehta, D.S.; Chen, S.; Low, K.-S. A rotation-invariant additive vector sequence based star pattern recognition. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *55*, 689–705. [CrossRef]
22. Jiang, J.; Ji, F.; Yan, J.; Sun, L.; Wei, X. Redundant-coded radial and neighbor star pattern identification algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2015**, *51*, 2811–2822. [CrossRef]
23. 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]
24. Liebe, C.C. Pattern recognition of star constellations for spacecraft applications. *IEEE Aerosp. Electron. Syst. Mag.* **1993**, *8*, 31–39. [CrossRef]
25. Padgett, C.; Kreutz-Delgado, K.; Udomkesmalee, S. Evaluation of star identification techniques. *J. Guid. Control Dyn.* **1997**, *20*, 259–267. [CrossRef]
26. Padgett, C.; Kreutz-Delgado, K. A grid algorithm for autonomous star identification. *IEEE Trans. Aerosp. Electron. Syst.* **1997**, *33*, 202–213. [CrossRef]
27. Na, M.; Zheng, D.; Jia, P. Modified grid algorithm for noisy all-sky autonomous star identification. *IEEE Trans. Aerosp. Electron. Syst.* **2009**, *45*, 516–522. [CrossRef]
28. 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]
29. Samirbhai, M.D.; Chen, S.; Low, K.S. A hamming distance and spearman correlation based star identification algorithm. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *55*, 17–30. [CrossRef]
30. Zhang, G.; Zhang, G. *Star Identification Utilizing Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 153–176.
31. Alvelda, P.; San Martin, A. Neural network star pattern recognition for spacecraft attitude determination and control. *Adv. Neural Inf. Process. Syst.* **1989**, 314–322.
32. Wang, Y.; Zhang, H. Star recognition based on mixed star pattern and multilayer SOM neural network. In Proceedings of the 2017 IEEE Aerospace Conference, Big Sky, MT, USA, 4–11 March 2017; pp. 1–6.
33. Myers, J.; Sande, C.; Miller, A.; Warren, W., Jr.; Tracewell, D. SKY2000-master star catalog-star catalog database. *Bull. Am. Astron. Soc.* **1997**, *191*, 128.
34. Zhu, Z.; Ma, Y.; Dan, B.; Zhao, R.; Liu, E.; Zhu, Z. ISSM-ELM–a guide star selection for a small-FOV star sensor based on the improved SSM and extreme learning machine. *Appl. Opt.* **2022**, *61*, 6443–6452. [CrossRef]
35. Ayegba, P.; Ayoola, J.; Asani, E.; Okeyinka, A. A comparative study of minimal spanning tree algorithms. In Proceedings of the 2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS), Lagos, Nigeria, 18–21 March 2020; pp. 1–4.
36. Jun, Z.; Yuncai, H.; Li, W.; Da, L. Studies on dynamic motion compensation and positioning accuracy on star tracker. *Appl. Opt.* **2015**, *54*, 8417–8424. [CrossRef]
37. Sun, L.; Jiang, J.; Zhang, G.; Wei, X. A discrete HMM-based feature sequence model approach for star identification. *IEEE Sens. J.* **2015**, *16*, 931–940. [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.