An accurate star identification approach based on spectral graph matching for attitude measurement of spacecraft

Zhiyuan You¹ · Junzheng Li¹ · Hongcheng Zhang² · Bo Yang^{2,3,4} · Xinyi Le^{2,3,4}

Received: 25 April 2021 / Accepted: 3 December 2021 / Published online: 4 January 2022
© The Author(s) 2021

Abstract

Star identification is the foundation of star trackers, which are used to precisely determine the attitude of spacecraft. In this paper, we propose a novel star identification approach based on spectral graph matching. In the proposed approach, we construct a feature called the neighbor graph for each main star, transforming the star identification to the problem of finding the most similar neighbor graph. Then the rough search and graph matching are cooperated to form a dynamic search framework to solve the problem. In the rough search stage, the total edge weight in the minimum spanning tree of the neighbor graph is selected as an indicator, then the k-vector range search is applied for reducing the search scale. Spectral graph matching is utilized to achieve global matching, identifying all stars in the neighbor circle with good noise-tolerance ability. Extensive simulation experiments under the position noise, lost-star noise, and fake-star noise show that our approach achieves higher accuracy (mostly over 99%) and better robustness results compared with other baseline algorithms in most cases.

Keywords Star tracker · Star identification · Graph matching · Pattern recognition

Introduction

Motivation and related works

Star tracker, a precise device used to determine the attitude of spacecraft, is one of the most extensively adopted celestial navigation instruments [33,34]. Star tracker completes the attitude measure through three procedures: star centroiding, star identification, and attitude calculation, among which star identification is the fundamental component for attitude measurement. In terms of the applied field, there are two types of star identification tasks, recursive case (some prior attitude information is available) and lost-in-space case (no prior atti-

tude information is available) [33]. In this paper, we focus on the more general lost-in-space case.

Star identification is faced with two major difficulties. First, the large scale of catalog stars brings a heavy burden to star identification [33]. Second, the position noise, lost-star noise, and fake-star noise in stellar images are caused by imaging noise, optical distortion, and unexpected interference light, making star identification more challenging [23]. To overcome these difficulties, lots of star identification algorithms have been proposed. These algorithms could be roughly classified into two classes, subgraph-isomorphism-based ones and pattern-based ones.

The subgraph-isomorphism-based algorithms establish a graph by viewing each star as a vertex and the angular separation between each star-pair as an edge. The graph obtained from the stellar image is a subgraph of the entire graph, thus star identification could be completed by seeking the most similar subgraph. The triangle algorithm [24,30] and its modifications [5,14] utilize different search techniques to search the triangle subgraph. The pyramid algorithm [26] uses a permutation approach to select four stars to form a pyramid subgraph, decreasing the possibility of mis-identification. Wang et al. [35] encodes the triangle angular separation to integers and uses the hash map to search the database directly. However, existing subgraph-isomorphism-based algorithms

✉ Xinyi Le
lexinyi@sjtu.edu.cn

¹ School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China

² Department of Automation, Shanghai Jiao Tong University, Shanghai, China

³ Key Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai, China

⁴ Shanghai Engineering Research Center of Intelligent Control and Management, Shanghai, China

are susceptible to the position noise. Also, these algorithms complete the identification by integrating the matching of star-pairs. Hence, these algorithms could not achieve global matching. Here global matching means that all observed stars should be viewed as a whole and be identified through only one operation.

The pattern-based algorithms extract a pattern for a main star, then identify the main star by searching the most similar pattern in the database. The grid algorithm [29] and its modifications [17,22,28] divide the stellar image into grid cells to form a pattern. These algorithms rotate the stellar image using the alignment star to keep the consistency of the grid pattern. However, the choice of the critical alignment star could be easily confused by the noise, resulting in identification failure even mis-identification. Singular value decomposition (SVD) based star identification algorithms [12,13] treat the singular values as the pattern. K–L transform is used in [42] for rough matching, and a star walk step is performed to identify stars. [32] combines the hamming distance and Spearman correlation, increasing the identification efficiency. [20] uses image normalization and the Zernike moment to extract more invariant pattern features. [8,31] adopt neural networks to classify the extracted grid patterns. References [11,36] directly view the stellar image as a pattern and identify the main star through the convolution neural network (CNN). However, the above-mentioned approaches highly rely on the measured star magnitude whose accuracy is relatively low, increasing the variance of matching results.

Recent years have witnessed the growing attention on graph-based approaches [15,16,38,41]. Graph-based star identification algorithms have also emerged. Reference [9] uses the 2D Voronoi space partition to separate observed stars into its own polygon to generate a feature graph for star identification. However, the algorithm could only achieve star identification when the matched two feature graphs have the same number of vertices. Reference [10] adopts the ant colony algorithm to generate an optimal path as a feature for star identification, but it does not utilize more useful graph structure information directly.

Graph matching algorithms [4,6,7,18,37,39,40], as a kind of graph-based approaches, are developed to establish the correspondences between two sets of features. Each set of features is viewed as a graph by taking features as vertices and regarding the relationships of features as edges. Graph matching algorithms establish vertex correspondences using the affinity matrix that encodes both vertex similarity and edge similarity between two graphs. The graph matching could be formulated to a quadratic assignment program [1], subjected to one-by-one correspondence constraints (each vertex in one graph corresponds at most one vertex in another graph). This is known to be NP-hard, so it is often solved approximately by relaxing the constraints and finding local optima. Typically, there are two types of approxima-

tion methods, double-stochastic approximation methods and spectral approximation methods. Double-stochastic approximation methods [7,37,39] achieve quite accurate matching results with relatively more time consumption, while spectral approximation methods [6,18,40] are faster with a little sacrifice of accuracy.

Graph matching is promising for star identification because of its two properties. First, all vertex correspondences between two graphs could be obtained through only one operation. Second, graph matching could match two graphs with different vertex numbers. Therefore, graph matching could bring two advantages to star identification. (a) global matching. All observed stars could be viewed as a whole and be identified through only one operation. (b) noise-tolerance ability. The lost-star noise and fake-star noise change the star number in the stellar image, which is solvable using graph matching. Thus the noise-tolerance ability could be improved.

Contribution

To overcome the weaknesses of existing star identification algorithms, we propose a novel star identification algorithm based on graph matching. We summarize our contributions as follows:

- We propose a novel star identification algorithm based on graph matching. To our best knowledge, we are the first to introduce the graph matching algorithm to star identification. Our algorithm achieves global matching in that all observed stars are viewed as a whole and identified through only one operation, while traditional approaches could only match a few stars like a triangle or a pyramid. Our approach also has a good noise-tolerance ability.
- We propose a dynamic search framework that views the accuracy as the first target, and minimizes the time consumption as much as possible with the premise of high accuracy. Essentially, this framework formulates star identification as a bi-level optimization problem where the lower level is concerning the accuracy, and the upper level is the minimization of time consumption.
- Our approach achieves quite high accuracy (>99% mostly), and significantly outperforms classical algorithms including the pyramid algorithm [26], modified grid algorithm [17] and one-dimensional vector pattern algorithm [22] in simulation experiments under the position noise, lost-star noise, and fake-star noise.

Organization

In the next section, we formulate the star identification problem into a graph matching optimization. The proposed approach including database generation, rough search, graph

matching, and dynamic search framework is described in detail in the following section, followed by the simulations and experiment results in the next section. The characteristics of the proposed algorithm are discussed in the following section. Finally, the conclusion is given in the last section.

Preliminaries and problem formulation

Preliminaries

Star catalog The star catalog we use in this paper is Smithsonian Astrophysical Observatory (SAO) [27] star catalog, which contains visual magnitude and position information of catalog stars. For a catalog star i , the visual magnitude m_i measures its brightness, and its position is expressed as (α_i, β_i) , where α_i is the right ascension and β_i is the declination in the celestial sphere reference frame (Fig. 1).

Angular separation The angular separation is used to measure the distance between two stars. We denote the angular separation between star i and j as $\theta_{i,j}$ (Fig. 1).

Stellar image A stellar image is generated from the star tracker equipped in spacecraft (Fig. 14) or a star tracker simulation platform (Fig. 3a).

Weighted graph A graph is defined as $G = (V, E)$, where V is the vertex set and E is the edge set. If edges in E have been assigned weights, then G is a weighted graph. If every vertex-pair in V is connected by an edge in E , then G is a complete graph.

Adjacency matrix The adjacency matrix is a square matrix used to represent a graph. For a specified weighted graph G_I ,

the adjacency matrix is denoted as $A_I(i, j) = a_{ij}$, where a_{ij} is the weight of edge ij .

Minimum spanning tree The minimum spanning tree (MST) is a subset of edges of a connected, weighted, undirected graph that connects all vertices together, without any cycles and with the minimum possible total edge weight. For a specified graph G_I with adjacency matrix $A_I(i, j) = a_{ij}$, its total edge weight of MST is denoted as $w_I = \sum a_{ij}$, where edge ij belongs to MST of G_I .

Problem formulation

Given a stellar image and a star catalog, our goal is to establish the correspondences between observed stars in the stellar image and catalog stars. Star identification is a feature search and matching process. Certain predefined features are extracted for the stellar image. These features could be geometry, like a triangle [24,30,35] or a pyramid [14,26], or a matrix, like the grid matrix [17,28,29], or a vector, like the singular values [12,13]. A database composed of some selected catalog stars together with their extracted features is generated (see “Database generation”). Star identification algorithms match features extracted from the stellar image to features of database stars and find the best matching result. Then some observed stars could find their correspondences to database stars through the best matching result.

To identify stars in the stellar image through graph matching, we predefine a feature called neighbor graph. As shown in Fig. 2, an observed main star I in the stellar image is selected first. Then its neighbor graph G_I is generated through viewing the observed stars as vertices, the angular separations of star-pairs as weighted edges (see “Database generation”). Every database star could be viewed as a database main star J , and its neighbor graph is denoted as G_J .

Graph matching aims to establish vertex correspondences between $G_I = (V_I, E_I)$ and $G_J = (V_J, E_J)$. For example,

Fig. 1 Celestial sphere reference frame

Fig. 2 Graph matching between neighbor graph G_I of stellar image and database neighbor graph G_J . The red arrows indicate vertex correspondences. Vertex k in G_I is caused by fake-star noise, so it does not have correspondence in G_J . Thus G_J corresponds to the subgraph G'_I (the solid portion) of G_I

in Fig. 2, the graph matching result is that the vertex $I, i, j \in V_I$ correspond to the vertex $J, i, j \in V_J$, respectively. Vertex $k \in V_I$ does not have correspondence in V_J since it is caused by fake-star noise. A matching error function is proposed to evaluate the graph matching result (see “Spectral graph matching”). Naturally, the database neighbor graph G_J that has the minimum matching error with G_I is the best matching result. Then some observed stars establish correspondences to database stars using the best matching result.

A rough search is applied for reducing the search scale because matching G_I to all database neighbor graphs is quite time-consuming. One unique and noise-tolerant indicator of the neighbor graph is required in the rough search. The total edge weight of MST is such kind of suitable indicator of the neighbor graph because it represents the scale of the vertex number, remaining relatively stable under different noise conditions. We calculate w_I (the total edge weight of MST of G_I), and denote the uncertainty of w_I as Δw . The rough search aims to find a candidate main star set S_I satisfying

$$S_I = \{J | w_I - \Delta w \leq w_J \leq w_I + \Delta w\}. \quad (1)$$

After the rough search, the scale of S_I is much smaller than the scale of database stars. Therefore, the search scale is reduced dramatically.

In summary, star identification is formulated to the rough search and graph matching. The identification result is the neighbor graph that has the minimum matching error. Since Δw varies in different cases in practice, the rough search and graph matching is operated under a dynamic search framework. The rough search, graph matching and the dynamic search framework are described in detail in “Algorithm description”.

Algorithm description

Database generation

The SAO star catalog [27] we use is an official version published by Astronomical Data Center, National Aeronautics and Space Administration (NASA), which could be downloaded via [2]. Since this official version is first established in 1979, it has been continuously used by global researchers, indicating its high reliability. In this paper, the highest visual magnitude available detected by the star tracker is set to 6.0. There are 4489 visible stars in the SAO star catalog. These visible stars serve as database stars because a database is generated using these stars through graph theory.

Each database star could be viewed as the main star, so a database containing 4489 main stars together with their features is generated. A database main star J has two features, w_J (the total edge weight of MST) for the rough search and

Fig. 3 The generation of the neighbor graph and MST. (1) Select the main star J and draw its neighbor circle with radius r . (2) Generate the neighbor graph G_J of J by viewing stars as vertices and viewing angular separation between star-pairs as weighted edges. G_J is a complete weighted graph (every star-pair is connected by an edge, and the edge weight is indicated by the length). (3) Generate the MST of G_J through Kruskal algorithm [3,19] for rough search

A_J (the adjacency matrix) for the graph matching. The generation of the neighbor graph and its MST is illustrated as Fig. 3. Each database main star J is projected into the image center first. Then centered on J , the circle with radius r is called neighbor circle, and the stars located in the neighbor circle except the main star are called neighbor stars. The main star and neighbor stars constitute a graph G_J by viewing stars as vertices and viewing angular separation between star-pairs as edges. We call G_J the neighbor graph of J . G_J is a complete graph, indicating the position information of observed stars is fully exploited. The angular separation $\theta_{i,j}$ between star i and star j could be calculated through

$$\theta_{i,j} = \arccos(v_i^l \cdot v_j^l),$$

where v_i^l is the unit vector of star i in an inertial based coordinate system [21], and it could be calculated using its right ascension α_i and declination β_i . Similar for v_j^l . G_J could be viewed as a weighted graph by treating $\theta_{i,j}$ as the weight of edge ij . Then the adjacency matrix A_J of G_J is generated through $A_J(i, j) = \theta_{i,j}$. A_J serves as a feature of the corresponding main star J for the graph matching. We adopt Kruskal algorithm [3,19] to search the MST of G_J and then calculate w_J . w_J also serves as a feature of the corresponding main star J for the rough search.

Rough search

After a stellar image is obtained through the star tracker, a main star I is selected and its two features, w_I (the total edge weight of MST) and A_I (the adjacency matrix), need to be generated. First, the centroid pixel coordinate of each observed star is derived through star centroiding [21]. To fully exploit the position information of observed stars, the neighbor circle should fall into the image frame as much as possible. Thus, the star nearest the image center is selected as the main star I . Then the stars falling into the neighbor circle are selected as neighbor stars. The angular separation $\theta_{i,j}$ between star i and star j could be calculated using

$$\theta_{i,j} = \arccos(v_i^s \cdot v_j^s),$$

where v_i^s is the unit vector of star i in a star tracker-based coordinate system [21], and similar for v_j^s . Finally, w_I and A_I are generated using the method described before.

As shown in Fig. 4, the rough search is carried out in the database. Assume that the uncertainty of w_I is Δw . The rough search aims to find a candidate main star set S_I containing all database main stars J whose w_J satisfies $w_I - \Delta w \leq w_J \leq w_I + \Delta w$. The k-vector range search [25] is a quite fast range search algorithm. It is only need several calculations to find the elements in the target range no matter how large the scale of the searched database is. The rough search utilizes the k-vector range search to obtain the candidate main star set S_I . Therefore, w_I has the possibility to correspond to w_J of main stars J in S_I . Also, the main star I and its neighbor stars have the possibility to correspond to main stars J in S_I and their neighbor stars, respectively.

Fig. 4 The framework of rough search. The MST generation is the same as Fig. 3. The edges in MST are weighted edges, whose weights are indicated by their length. The k-vector range search [25] takes the MST generated through the stellar image and database main star set as inputs and outputs the candidate main star set S_I

Spectral graph matching

The graph matching is used to establish vertex correspondences between a neighbor graph $G_I = (V_I, E_I)$ obtained from the stellar image and a database neighbor graph $G_J = (V_J, E_J)$. We adopt the spectral graph matching algorithm [18,40] because it is a quite concise and efficient graph matching algorithm with only a little sacrifice of accuracy compared with the double-stochastic approximation methods. The spectral graph matching framework is shown in Fig. 5. First, an affinity matrix $K_{I,J}$ is constructed to represent vertex similarity and edge similarity between G_I and G_J . Then the unit principal eigenvector $\gamma_{I,J}$ of $K_{I,J}$ is viewed as the confidence of the possible vertex correspondences. Finally, the permutation matrix $P_{I,J}$ representing vertex correspondences are derived through a binarization and reshape technique and the matching error $\delta_{I,J}$ is calculated.

Denote the vertex number of G_I, G_J as n_I, n_J , respectively, then the $n_I n_J \times n_I n_J$ affinity matrix $K_{I,J}$ between G_I and G_J is constructed as follows. First, the representation of vertex and edge similarity in $K_{I,J}$ is described. The diagonal entries of $K_{I,J}$ represent the vertex similarity. For example,

Fig. 5 The framework of spectral graph matching. The neighbor graph generation is the same as Fig. 3. The spectral graph matching algorithm [18,40] takes the neighbor graph G_I of the stellar image and database neighbor graph G_J as inputs and outputs the vertex correspondences, which are indicated by the red arrows. Thus G_J corresponds to the subgraph G'_I (the solid portion) of G_I

$K_{I,J}(ij, ij)$ is the similarity between vertex $i \in V_I$ and vertex $j \in V_J$. The non-diagonal entries of $K_{I,J}$ represent the edge similarity. For example, $K_{I,J}(ij, km)$ is the similarity between edge $ik \in E_I$ and edge $jm \in E_J$. Because the neighbor graph is an undirected graph and the vertex or edge similarity is non-negative, $K_{I,J}$ is a symmetric matrix with non-negative entries. The affinity matrix $K_{I,J}$ is also “global” because it contains all possible vertex similarity and edge similarity, which is the key to global matching. Second, vertex and edge similarity are specified. Since edges represent all position information of observed stars and using only edges is enough for the graph matching, the vertex similarity $K_{I,J}(ij, ij)$ is defined as 0. The edge similarity $K_{I,J}(ij, km)$ is defined with the Gaussian kernel.

$$K_{I,J}(ij, km) = e^{-c|A_I(i,k) - A_J(j,m)|}, \quad (2)$$

where A_I, A_J are adjacency matrices of G_I, G_J , c is a positive number to adjust the magnitude of the edge similarity. Apparently, $K_{I,J}(ij, km) \in (0, 1]$. If and only if $A_I(i, k)$ equals to $A_J(j, m)$, $K_{I,J}(ij, km)$ equals to 1, which is the maximum similarity between edge $ik \in E_I$ and edge $jm \in E_J$. The larger the difference between $A_I(i, k)$ and $A_J(j, m)$ is, the closer to 0 the similarity between edge $ik \in E_I$ and edge $jm \in E_J$ is.

The $n_I \times n_J$ permutation matrix $P_{I,J}$ is used to represent vertex correspondences between G_I and G_J .

$$P_{I,J}(i, j) = \begin{cases} 1, & \text{If } i \in V_I \text{ corresponds to } j \in V_J \\ 0, & \text{Otherwise,} \end{cases} \quad (3)$$

where $i = 1, 2, \dots, n_I$, $j = 1, 2, \dots, n_J$. Then graph matching problem could be formulated as a quadratic assignment problem as follows (see [18,40] for details)

$$\begin{aligned} P_{I,J}^* &= \operatorname{argmax} \left(\operatorname{vec}(P_{I,J})^T K_{I,J} \operatorname{vec}(P_{I,J}) \right) \\ \text{s.t. } &P_{I,J}(i, j) \in \{0, 1\}, \quad i = 1, 2, \dots, n_I, \quad j = 1, 2, \dots, n_J \\ &P_{I,J}^T \mathbf{1}_{n_I} \leq \mathbf{1}_{n_I}, \quad P_{I,J} \mathbf{1}_{n_J} \leq \mathbf{1}_{n_J}, \end{aligned} \quad (4)$$

where $\operatorname{vec}(\cdot)$ stacks all columns of a matrix to form a vector, $\mathbf{1}_{n_J}$ is an n_J length all-1 vector, the same for $\mathbf{1}_{n_I}$. Problem (4) is an NP-hard integer optimization problem.

The spectral graph matching algorithm [18,40] is adopted to find local optima of Problem (4) and establish vertex correspondences between G_I and G_J . The unit principal eigenvector (the unit eigenvector corresponding to the maximum eigenvalue) $\gamma_{I,J}$ of $K_{I,J}$ is calculated first. According to the Perron–Frobenius theorem, the entries of $\gamma_{I,J}$ will be in the interval $[0, 1]$. It has been proved that $\gamma_{I,J}$ is the optimal solution of a simplified problem (relaxing the integer constraint to the unit constraint) of problem (4). Then $\gamma_{I,J}(ij)$

could be seen as the confidence of correspondence between vertex $i \in V_I$ and vertex $j \in V_J$. A binarization technique is utilized to transform $\gamma_{I,J}$ to $\operatorname{vec}(P_{I,J})$ (see [18,40] for details). Then we reshape $\operatorname{vec}(P_{I,J})$ to the permutation matrix $P_{I,J}$ that represents vertex correspondences between G_I and G_J .

We propose a new matching error function to evaluate the graph matching results. Take the same example in Figs. 2 and 5 to illustrate the process. The graph matching result between $G_I = (V_I, E_I)$ and $G_J = (V_J, E_J)$ is that vertex $I, i, j \in V_I$ correspond to vertex $J, i, j \in V_J$. So G_J corresponds to the subgraph G'_I (the solid portion) of G_I . Note that vertex $k \in V_I$ does not have correspondence in V_J because it is caused by fake-star noise. We denote the adjacency matrix of G_I, G'_I, G_J as A_I, A'_I, A_J , respectively. A'_I is a trimmed matrix of A_I because G'_I is the subgraph of G_I . Then the matching error $\delta_{I,J}$ is given as follows.

$$\delta_{I,J} = \frac{\|A'_I - A_J\|_F}{\min(n_I, n_J)}, \quad (5)$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix (the square root of the sum of the squares of all entries of a matrix), $\min(n_I, n_J)$ is the number of matched vertices. It could be seen that $\delta_{I,J}$ is only related to the corresponding parts of the two graphs and has no relationship with vertex k in G_I . Usually, the noise vertex would not belong to the corresponding parts of the two graphs, so the noise vertex would not increase the matching error. This characteristic indicates noise-tolerance ability.

The rough search and spectral graph matching are incorporated to complete star identification. For the main star I in a stellar image, w_I and A_I are generated. w_I is used for the rough search to obtain the candidate main star set S_I . For each candidate main star J in S_I , its neighbor graph G_J is used to do graph matching with G_I . The permutation matrix $P_{I,J}$ and the matching error $\delta_{I,J}$ is calculated. Apparently, the candidate main star that has the minimum matching error and its neighbor stars is considered as the best identification result of I and neighbor stars of I , respectively.

Dynamic search framework

In practice, setting a fixed Δw (the uncertainty of w_I) is not appropriate for all cases because Δw is varied in different cases. For example, Δw is small under the position noise, while it is relatively large when lost-stars or fake-stars occur at the boundary region of the neighbor circle. Therefore, if Δw is set too large, more time is wasted unnecessarily in some cases. If Δw is set too small, the failure rate would increase because the true corresponding main star in more cases would not in S_I .

Algorithm 1 The dynamic search framework

Given: the initial uncertainty of w_I : Δw , the step size of Δw : $h^{\Delta w}$, the maximum of Δw : $\Delta w^{\max}$, the threshold of matching error: ϵ^δ

Define RoughSearch($w_I, \Delta w$)

Generate $S_I = \{J | w_I - \Delta w \leq w_J \leq w_I + \Delta w\}$ through k-vector range search

return S_I

Define SpectralGraphMatching(A_I, A_J)

Construct affinity matrix $K_{I,J}$ between G_I and G_J

Calculate principle eigenvector $\gamma_{I,J}$ of $K_{I,J}$

Binarize and reshape $\gamma_{I,J}$ to permutation matrix $P_{I,J}$

Calculate matching error $\delta_{I,J}$ using $P_{I,J}$

return $P_{I,J}, \delta_{I,J}$

Input: a stellar image

1: Select main star I through the stellar image

2: Construct G_I , Calculate A_I and w_I

3: **repeat**

4: $S_I = \text{RoughSearch}(w_I, \Delta w)$

5: **for** J in S_I **do**

6: $P_{I,J}, \delta_{I,J} = \text{SpectralGraphMatching}(A_I, A_J)$

7: **end for**

8: $J^* = \arg \min_{J \in S_I} \delta_{I,J}$

9: $\Delta w = \Delta w + h^{\Delta w}$

10: **until** $\delta_{I,J^*} \leq \epsilon^\delta$ or $\Delta w > \Delta w^{\max}$

11: **if** $\delta_{I,J^*} \leq \epsilon^\delta$ **then**

12: **return** J^* and P_{I,J^*} as star identification result

13: **else**

14: **return** identification failure

15: **end if**

We proposed a dynamic search framework (Algorithm 1) based on increasing Δw to solve the above problems. For the main star I in a stellar image, w_I is calculated. Δw is set small in the beginning. The rough search uses w_I and Δw to obtain the candidate main star set S_I . For each candidate main star J in S_I , its neighbor graph G_J is used to match with G_I . The permutation matrix $P_{I,J}$ and the matching error $\delta_{I,J}$ is calculated. We denote the candidate main star with the minimum matching error as J^* .

$$J^* = \arg \min_{J \in S_I} \delta_{I,J}. \quad (6)$$

Also, the minimum matching error is denoted as δ_{I,J^*} . We preset a small enough matching error threshold ϵ^δ . If $\delta_{I,J^*} \leq \epsilon^\delta$, then J^* is considered as the identification result of I . Similarly, neighbor stars of J^* are considered as the identification result of neighbor stars of I . If $\delta_{I,J^*} > \epsilon^\delta$, we enlarge Δw for a more throughout rough search and graph matching. The process is repeated until δ_{I,J^*} is smaller than ϵ^δ or Δw is large enough. Note that repeated searches are excluded when Δw is enlarged. Star identification fails if Δw is large enough. Note that the failure happens rarely, and the failure is more desirable than the mis-identification.

Essentially, the dynamic search framework is a bi-level optimization framework. The lower level is the accuracy

because the accuracy is always the first consideration. The time consumption is minimized as much as possible under the premise of high enough accuracy. So the upper level of the bi-level optimization is the minimization of time consumption.

Simulation and experiment

In this section, the simulation setting is described first. Then three simulation experiments are conducted to test the proposed algorithm, followed by three parameter experiments to select hyper parameters. Finally, the proposed algorithm is tested on real stellar images obtained by a star tracker on an orbiting spacecraft.

Simulation setting

Simulated stellar images are generated for simulation experiments. The resolution of the simulated stellar image is 1024×1024 px with a $20^\circ \times 20^\circ$ field of view (FOV). To project a database star i in the FOV into the image plane, the unit vector of star i needs to be transformed from an inertial based coordinate system into a star tracker based coordinate system as follows.

$$v_i^s = \begin{bmatrix} v_i^{sx} \\ v_i^{sy} \\ v_i^{sz} \end{bmatrix} = R_x \left(\beta_0 - \frac{\pi}{2} \right) R_z \left(\alpha_0 - \frac{\pi}{2} \right) v_i^l, \quad (7)$$

where R_x is the rotation matrix whose rotation axis is the x axis and the same for R_z , (α_0, β_0) is the right ascension and declination of the optical axis direction. Note that the v_i^s in Eq. 7 is derived under the assumption that the roll angle of CCD is 0° . Then the centroid pixel coordinate (X_i, Y_i) of star i is calculated.

$$X_i = -\frac{v_i^{sx} N^x}{2v_i^{sz} \tan \left(\frac{FOV^x}{2} \right)}, \quad (8)$$

$$Y_i = -\frac{v_i^{sy} N^y}{2v_i^{sz} \tan \left(\frac{FOV^y}{2} \right)},$$

where (N^x, N^y) and (FOV^x, FOV^y) are the resolution and FOV of CCD, respectively. Note that (X_i, Y_i) is the centroid pixel coordinate with the center of the image as the origin. The simulated stellar image is obtained after projecting all database stars in the FOV into the image plane.

The performance of star identification algorithms are evaluated by the accuracy, which is calculated through

$$Accuracy = \frac{NS}{TN - NF}, \quad (9)$$

Table 1 The results of parameter selection

Parameter	Meaning	Value
r	Radius of the neighbor circle	6°
Δw	Initial value of the uncertainty of w_I	0.1°
$h^{\Delta w}$	Step size of the uncertainty of w_I	0.1°
$\Delta w^{\max}$	Maximum of the uncertainty of w_I	6°
c	Edge similarity adjustor	1.74
ϵ^δ	Threshold of the matching error	0.4°

where TN is the total number of stellar images, NF is the number of identification failure, NS is the number of successfully identified stellar images. A stellar image is considered to be successfully identified if all observed stars in the neighbor circle are identified. Note that $TN = NF + NS + NM$, where NM is the number of mis-identified stellar images. $TN - NF$ is the number of identification with results returned.

Simulation experiment

To carry out simulation experiments, parameter experiments are designed and implemented to select the following hyper parameters (see “Parameter experiment”).

Simulation experiments are conducted to test the proposed algorithm. The proposed algorithm is compared with pyramid [26], MGrid [17] and ODVP algorithm [22]. Usually, there are three types of noise in the stellar image, position noise, lost-star noise and fake-star noise. Therefore, three experiments are conducted to test three algorithms. The first experiment tests the algorithms under position noise. The second experiment is implemented under lost-star noise mixed with position noise. The third one is about the fake-star noise mixed with position noise. The performance of the four algorithms is evaluated under the same simulation conditions.

Position noise The position noise follows a Gaussian distribution with mean 0 px, and the std is set from 0 px to 3 px with the step of 0.5 px. The performance of the four algorithms is shown in Fig. 6. The mean accuracy (MA), standard deviation (stdA), range (RA) of the accuracy list are calcu-

Fig. 6 Accuracy under position noise

lated. They together with accuracy under the largest noise (ALN) are shown in Table 2. The average performance of the proposed algorithm is superior to the pyramid, MGrid and ODVP algorithm. The proposed algorithm is also more robust and stable than the three algorithms when position noise increases. When the std of position noise is the largest (3 px), the performance of the proposed algorithm is much better than the three algorithms. We attribute the performance improvement under position noise to the global matching property of graph matching. First, the displacement brought by position noise is much smaller than the radius of the neighbor circle. Besides, all stars in the neighbor circle are viewed as a whole during graph matching. Therefore, the global geometric information of stars in the neighbor circle is still kept well under position noise and the accuracy also keeps stable.

Lost-star mixed with position noise The number of lost-star is set as 1 and the setting of position noise is the same as the first experiment. Figure 7 shows the statistical results of the accuracy of the four algorithms. When the std of position noise is small (0.0 and 0.5 specifically), the pyramid algorithm (100%, 99.80%) performs slightly better than the proposed algorithm (99.60%, 99.60%). However, with the increase of position noise, the performance of the pyramid algorithm decreases dramatically (94.9% when std of position noise is 3.0). While the proposed algorithm still keeps a quite high accuracy (mostly above 99%). Also, the MA, stdA,

Table 2 Performance comparison under different noise conditions

	Position noise				Lost-star mixed with position noise				Fake-star mixed with position noise			
	Pyramid	MGrid	ODVP	Ours	Pyramid	MGrid	ODVP	Ours	Pyramid	MGrid	ODVP	Ours
MA (%)	97.63	98.30	94.96	99.71	97.73	96.90	95.57	99.26	97.67	97.03	93.14	99.42
stdA (%)	1.86	1.09	3.93	0.22	1.86	1.26	3.74	0.38	1.84	1.30	5.05	0.28
RA (%)	5.20	3.60	9.67	0.60	5.10	4.43	11.64	0.97	4.90	3.80	14.70	0.85
ALN (%)	94.80	96.40	90.25	99.69	94.90	94.26	88.15	99.05	95.10	94.99	84.38	99.16

The bold terms mean better performance

Fig. 7 Accuracy under 1 lost-star mixed with position noise

Fig. 8 Accuracy under 1 fake-star mixed with position noise

RA and ALN of three algorithms are calculated and shown in Table 2. The proposed algorithm not only far surpasses the three algorithms averagely, but is much more robust to noise. The proposed algorithm greatly outperforms the three algorithms under the largest position noise (std 3 px). The performance improvement under lost-star noise is attributed to the properties of graph matching and the design of the matching error function. First, the graph matching could match two graphs with different vertex numbers. Second, the designed matching error is only related to the corresponding parts of the two graphs. The neighbor graph of the stellar image has fewer vertices than the corresponding database neighbor graph under lost-star noise, so the corresponding parts of the two graphs become smaller. The matching error would not increase in this case mostly and the accuracy is also not affected largely.

Fake-star mixed with position noise The number of fake-star is set as 1 and the position noise setting is the same as the above two experiments. The experiment results are shown in Fig. 8. The pyramid algorithm (100%, 99.70%) performs a little better than the proposed algorithm (99.70%,

99.39%) when the std of position noise is small (0.0 and 0.5 specifically). However, the proposed algorithm keeps a quite high accuracy (always above 99%) with the increase of position noise, while the performance of the pyramid algorithm decreases dramatically (95.1% with std 3.0). The MA, stdA, RA and ALN in Table 2 are used to compare the performance of three algorithms. Obviously, the average performance of the proposed algorithm is much better than the pyramid, MGrid and ODVP algorithm. The robustness of the proposed algorithm also outperforms the three algorithms greatly. When the position noise is largest (std 3 px), the identification of the proposed algorithm is much more accurate than the three algorithms. The reason for performance improvement under fake-star noise is similar to the lost-star case. The fake-star vertices do not belong to the corresponding parts of the two graphs, so the fake-star noise would not increase the matching error and the accuracy still keeps at a quite high level.

As can be seen through the above description, the proposed algorithm outperforms the pyramid, MGrid and ODVP algorithm under the position noise, lost-star noise and fake-star noise. More importantly, the accuracy of the proposed algorithm does not decrease with the increase of the position noise in the above three experiments, but keeps relatively stable at a quite high level. This indicates the good noise-tolerance ability of the proposed algorithm.

Parameter experiment

Parameter experiments are designed and implemented to select the hyper parameters in Table 1.

The radius of the neighbor circle r is set as 6° according to the following experiment results. r is a critical parameter of the proposed algorithm. If r is too small, there are fewer neighbor stars, so the neighbor graph is relatively simple. Naturally, the uniqueness of the neighbor graph decreases, thus the possibility of mismatching increases. If r is too large, the neighbor circle may beyond the image frame, which causes some information loss and increases the mismatching possibility. Therefore, r should keep the uniqueness of the neighbor graph and the neighbor circle should fall into the image frame as much as possible. The discussion above provides a method to select the best r . We randomly select the right ascension and declination of the optical axis direction to form a stellar image. The observed star closest to the image center is selected as the main star I . We draw the smallest circle centered on I and tangent to the image frame. The angular separation between I and the tangent point could be derived. We denote it as $r^{\max}$ because it is the maximum possible value of r to make the neighbor circle fall into the image frame (Fig. 9). We repeat the process 2000 times. The proportion of simulations that satisfying $r^x < r^{\max}$ is calculated with different r^x in region $[4^\circ, 10^\circ]$ (Fig. 10). Obviously, the

Fig. 9 The calculation of $r^{\max}$. The triangle is the image center, and I is the main star. The radius of the smallest circle centered on I and tangent to the image frame is denoted as $r^{\max}$

Fig. 10 The proportion of simulations satisfying $r^x < r^{\max}$ with various r^x

larger r is, the more unique the neighbor graph is. Besides, if r is less than $r^{\max}$, the neighbor circle fully falls into the stellar image. Therefore, the largest r^x satisfying $r^x < r^{\max}$ in the proportion of near 100% is the best r . Thus we set r as 6° .

For the w_I uncertainty, we set the initial value Δw as 0.1° , the step size $h^{\Delta w}$ as 0.1° , the maximum value $\Delta w^{\max}$ as 6° based on following experiment results. First, the w_I of a randomly simulated stellar image is calculated. Then we randomly add one of the three types of noise: position noise with mean 0 px and std 1 px, 1 lost-star noise, 1 fake-star noise. We calculate w_I^n (w_I under noise) of the stellar image under noise. The difference value Δ is derived through

$$\Delta = |w_I - w_I^n|.$$

The process is repeated 12,000 times. The proportion of simulations that satisfying $\Delta < \Delta^x$ is calculated with different Δ^x in region $[0^\circ, 6^\circ]$ (Fig. 11). It could be seen that the maximum possible value for Δ is 6° in this experiment, so the maximum value $\Delta w^{\max}$ is set as 6° to get high enough accu-

Fig. 11 The proportion of simulations satisfying $\Delta < \Delta^x$ with various Δ^x

racy. Usually, the smaller the initial value Δw and the step size $h^{\Delta w}$ are, the faster the proposed algorithm could find the matching result with small enough matching error to complete star identification. Also, it could be seen in Fig. 11 that Δ is quite small in most cases (less than 0.1 at 70%, less than 0.2 at 80%), which means a relatively small Δw is enough to complete star identification in most cases. Therefore, we set the initial value Δw as 0.1° and the step size $h^{\Delta w}$ as 0.1° .

The experiment results show that the edge similarity adjustor c is not critical to identification results. We set c as 1.74 empirically.

We set the threshold of the matching error ϵ^δ as 0.4° according to the accuracy and failure rate with different ϵ^δ . The failure rate is calculated through

$$\text{Failure rate} = \frac{\text{NF}}{\text{TN}},$$

where TN is the total identification number, NF is the number of identification failure. ϵ^δ is influential to identification results. If ϵ^δ is too small, the accuracy increases, while the failure rate goes up. If ϵ^δ is too large, the failure rate decreases, but the accuracy goes down. Note that the failure is more desirable than the mis-identification, so we pay more attention to the accuracy rather than the failure rate. Therefore, the best ϵ^δ results in quite high accuracy and relatively low failure rate. To select ϵ^δ that has good error tolerance ability, the large position noise condition is taken into consideration. The position noise follows a Gaussian distribution with mean 0 px and standard deviation (std) 3 px. The results are shown in Figs. 12 and 13. It could be seen that the accuracy and failure rate both decrease with the increase of ϵ^δ roughly. When ϵ^δ is smaller than or equal to 0.4° , the accuracy is quite high (near 100%). When ϵ^δ is larger than or equal to 0.4° , the failure rate is stable at around 2%, which is a relatively low failure rate under such a large position noise condition. Thus, we set ϵ^δ as 0.4° .

Fig. 12 Accuracy with different ϵ^δ

Fig. 13 Failure rate with different ϵ^δ

Orbiting experiment on real data

To compare the proposed algorithm with the pyramid and modified grid algorithm on real data, we collect some stellar images obtained by a star tracker on an orbiting spacecraft. The FOV and resolution of the star tracker are $19.7^\circ \times 11.2^\circ$ and 1920×1080 px, respectively. Some parameters are re-selected to adapt to the characteristics of the star tracker. The visual magnitude threshold is set as 5.0. The radius of the neighbor circle r is set as 5° . The threshold of the matching error ϵ^δ is set as 0.2° .

The comparison results are shown as follows. For a specific identification case, the proposed algorithm achieves star identification successfully, while the pyramid algorithm reports identification failure because of large noise, and the MGrid and ODVP algorithm return worse mis-identification. The stellar image is shown in Fig. 14. We invert its gray values for better visualization (Fig. 15a). The star nearest the image center is selected as the main star I . Centering on I , there are five neighbor stars falling into the neighbor circle. Through our proposed algorithm, I is identified as the

Fig. 14 The stellar image obtained from a star tracker on an orbiting spacecraft

(a) The inverted stellar image of Fig. 14.

(b) The stars in the neighbor graph of database star 123005.

Fig. 15 The star identification results of the stellar image obtained from a star tracker on an orbiting spacecraft. The main star I in the orbiting stellar image corresponds to database star 123035 (the main star J). The correspondences between neighbor stars of I and J are illustrated using circles with the same color. The database star 123035 (circled with the red dotted circle) indicates lost-star noise

database star 123005 (the SAO catalog star whose catalog number is 123005). We denote the database star 123005 as J . The stars in the neighbor graph of J are shown in Fig. 15b. The correspondences between neighbor stars of I and J are illustrated using circles with the same color. Therefore, the five neighbor stars of I are identified as the database stars 123013, 123107, 123140, 122671, 122754, respectively. It could also be seen that the database star 123035 (circled with the red dotted circle) could not find its corresponding star in Fig. 15a because of lost-star noise. Thus, the effectiveness of the proposed algorithm on real data is verified.

Fig. 16 The proportion of time consumption t satisfying $t < t_x$ with various t_x

Time consumption analysis

Our approach is applied in lost-in-space case, where no prior attitude information is available. In practice, after the attitude is calculated in lost-in-space case, approaches for recursive case where some prior attitude information is available could be applied [33]. Naturally, the time consumption for lost-in-space case is relatively larger than that for recursive case, so the demand of time consumption for lost-in-space case is relatively lower. Our algorithm is implemented in Python and evaluated on an Intel i7-8700 CPU. The time consumption in our simulation experiments is collected and analyzed. The results are shown in Fig. 16. It could be seen that 60% of the stellar images could be identified in 1 second, which is acceptable considering the high accuracy.

Discussion

In this paper, we provide a fresh graph-matching-based perspective for star identification. The practical significance and the future research are described as follows.

First, our algorithm could achieve global matching for all stars in the neighbor circle, which is more reliable in practice. In our approach, all star vertices (except fake-star vertices) could find their correspondences among database stars. These increase the uniqueness of the neighbor graph of each star, thus improving the accuracy. Directly matching the neighbor graph of the stellar image with a global graph constructed by all database stars is an ideal solution. However, due to the constraints of calculation ability, the direct matching is not achievable so far. Our method is a step closer to the direct matching, which belongs to our future work.

Second, the lost-star noise or fake-star noise would not increase the matching error, bringing good noise-tolerance ability. It is described before that the matching error is only related to the corresponding parts of the two graphs. When the fake-star noise happens, the fake-star vertices would not belong to the corresponding parts of the two graphs, so the fake-star noise would not increase the matching error. When the lost-star noise happens, some star vertices in the corresponding database neighbor graph would not find their correspondence in the neighbor graph of the stellar image. Thus, the corresponding parts of the two graphs become smaller. The matching error would not increase in this case mostly. These discussions support that the proposed algorithm has good noise-tolerance ability. It also provides a novel thought for the design of better error functions.

Third, our method can be extended for incomplete global matching, which belong to our future research. When lost-star noise and fake star noise happen simultaneously, the true matching result is an incomplete global matching. That is to say, some vertices in G_I can not find correspondences in G_J because of the lost-star noise. Also, some vertices in G_J can not find correspondences in G_I due to the fake-star noise. In such cases, a small change is needed to adapt to the incomplete global matching. As mentioned before, $\gamma_{I,J}(ij)$ could be seen as the confidence of correspondence between vertex i in G_I and vertex j in G_J . Therefore, we can set a minimum confidence threshold, then exclude all the corresponding relationships with confidence less than the threshold. The incomplete global matching could be achieved. By the way, the same technique could be used for time consumption reduction. We can set another maximum confidence threshold, only when the maximum corresponding confidence is greater than the threshold, can we carry out the following steps, including binarization, reshape and matching error calculation. Thus the time consumption could be reduced.

Conclusion

A novel star identification approach based on graph matching has been proposed for the attitude determination of spacecraft. The proposed approach contains rough search, graph matching, and a dynamic search framework. The rough search utilizes the k-vector range search to reduce the search scale. The graph matching is adopted to finely identify stars. The dynamic search framework is actually a bi-level optimization framework with the accuracy optimization as the lower level and the minimization of time consumption as the upper level. Our proposed approach could achieve global matching for all stars in the neighbor circle. Simulation experiments under the position noise, lost-star noise, and fake-star noise conditions all outperform higher accuracy

compared with the pyramid, MGrid and ODVP algorithm in most cases. Directly matching the neighbor graph of the stellar image with a global graph constructed by all database stars is a better solution, which is what our further study would focus on.

Funding Funding was provided by Oceanic Interdisciplinary Program of Shanghai Jiao Tong University (Grant No. SL2020MS015) and National Natural Science Foundation of China (Grant No. 62176152).

Availability of data and material The SAO star catalog is available in [2,27]. The collected real stellar images will be opened after acceptance.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code availability We will open the source code after acceptance.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Abdel-Basset M, Manogaran G, Rashad H, Zaid ANH (2018) A comprehensive review of quadratic assignment problem: variants, hybrids and applications. *J Ambient Intell Humaniz Comput* 1–24
- Browse Software Development Team (2004) Sao database. <https://heasarc.gsfc.nasa.gov/W3Browse/star-catalog/sao.html>
- Chakraborty M, Chowdhury S, Chakraborty J, Mehera R, Pal RK (2019) Algorithms for generating all possible spanning trees of a simple undirected connected graph: an extensive review. *Complex Intell Syst* 5(3):265–281
- Chen J, Wang L, Wang S, Wang X, Ren H (2021) An effective matching algorithm with adaptive tie-breaking strategy for online food delivery problem. *Complex Intell Syst* 1–22
- Cole CL, Crassidis JL (2006) Fast star-pattern recognition using planar triangles. *J Guid Control Dyn* 29(1):64–71
- Cour T, Srinivasan P, Shi J (2007) Balanced graph matching. In: *Advances in neural information processing systems*, pp 313–320
- Gold S, Rangarajan A (1996) A graduated assignment algorithm for graph matching. *IEEE Trans Pattern Anal Mach Intell* 18(4):377–388
- Hongchi S, Rongjun M, Huajun D (2018) A star identification method based on mixed characteristics and lvq neural network. In: *2018 10th international conference on modelling, identification and control (ICMIC)*. IEEE, pp 1–6
- Jiang D, Zhang K, Debeir O (2017) A graph based algorithm for star recognition. In: *IEEE information technology, networking, electronic and automation control conference*
- Jiang D, Zhang K, Debeir O (2018) Star recognition based on path optimization in star sensor with multiple fields of view. *Adv Astron*
- Jiang J, Liu L, Zhang G (2019) Star identification based on spider-web image and hierarchical cnn. *IEEE Trans Aerosp Electron Syst* 56(4):3055–3062
- Juang JN, Wang YC (2012) Further studies on singular value method for star pattern recognition and attitude determination. *J Astronaut Sci* 59(1–2):370–380
- Kim H, Junkins J, Juang J (2003) An efficient and robust singular value method for star pattern recognition and attitude determination. NASA, Tech Rep TM-2003-212142
- Kolomenkin M, Pollak S, Shimshoni I, Lindenbaum M (2008) Geometric voting algorithm for star trackers. *IEEE Trans Aerosp Electron Syst* 44(2):441–456
- Le X, Yan Z, Xi J (2017) A collective neurodynamic system for distributed optimization with applications in model predictive control. *IEEE Trans Emerg Top Comput Intell* 1(4):305–314
- Le X, Chen S, Yan Z, Xi J (2018) A neurodynamic approach to distributed optimization with globally coupled constraints. *IEEE Trans Cybern* 48(11):3149–3158
- Lee H, Bang H (2007) Star pattern identification technique by modified grid algorithm. *IEEE Trans Aerosp Electron Syst* 43(3):1112–1116
- Leordeanu M, Hebert M (2005) A spectral technique for correspondence problems using pairwise constraints. In: *10th IEEE international conference on computer vision (ICCV'05)*, IEEE, vol 2, pp 1482–1489
- Li H, Xia Q, Wang Y et al (2017) Research and improvement of kruskal algorithm. *J Comput Commun* 5(12):63
- Liang X, Ma W, Zhou J, Kong S (2020) Star identification algorithm based on image normalization and zernike moments. *IEEE Access* 8:29228–29237
- Liebe CC (2002) Accuracy performance of star trackers—a tutorial. *IEEE Trans Aerosp Electron Syst* 38(2):587–599
- Luo L, Xu L, Zhang H (2015) An autonomous star identification algorithm based on one-dimensional vector pattern for star sensors. *Sensors* 15(7):16412–16429
- Ma Y, Jiang J, Zhang G (2019) Stellar instrument magnitude estimation in infinite-dimensional space. *IEEE Sens J* 20(3):1422–1432
- Mortari D (1997) Search-less algorithm for star pattern recognition. *J Astronaut Sci* 45(2):179–194
- Mortari D, Neta B (2000) K-vector range searching techniques. *Spacefl Mech* 2000:449–463
- Mortari D, Samaan MA, Bruccoleri C, Junkins JL (2004) The pyramid star identification technique. *Navigation* 51(3):171–183
- Myers J, Sande C, Miller A, Warren Jr W, Tracewell D (1997) Sky2000-master star catalog-star catalog database. *Bull Am Astron Soc* 191(128.12)
- Na M, Zheng D, Jia P (2009) Modified grid algorithm for noisy all-sky autonomous star identification. *IEEE Trans Aerosp Electron Syst* 45(2):516–522
- Padgett C, Kreutz-Delgado K (1997) A grid algorithm for autonomous star identification. *IEEE Trans Aerosp Electron Syst* 33(1):202–213
- Quine B, Durrant-Whyte HF (1996) A fast autonomous star-acquisition algorithm for spacecraft. *Control Eng Pract* 4(12):1735–1740
- Rijlaarsdam D, Yous H, Byrne J, Oddenino D, Furano G, Moloney D (2020) Efficient star identification using a neural network. *Sensors* 20(13):3684

32. Samirbhai MD, Chen S, Low KS (2018) A hamming distance and spearman correlation based star identification algorithm. *IEEE Trans Aerosp Electron Syst* 55(1):17–30
33. Spratling BB, Mortari D (2009) A survey on star identification algorithms. *Algorithms* 2(1):93–107
34. Truesdale N, Skeen M, Diller J, Dinkel K, Dischner Z, Holt A, Murphy T, Schuette S, Zizzi A (2013) Daystar: modeling the daytime performance of a star tracker for high altitude balloons. In: 51st AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, p 139
35. Wang G, Li J, Wei X (2017) Star identification based on hash map. *IEEE Sens J* 18(4):1591–1599
36. Wang H, Wang Z, Wang B, Yu Z, Jin Z, Crassidis JL (2020) An artificial intelligence enhanced star identification algorithm. *Front Inf Technol Electron Eng* 21(11):1661–1670
37. Wang R, Yan J, Yang X (2019) Learning combinatorial embedding networks for deep graph matching. In: *Proceedings of the IEEE/CVF international conference on computer vision*, pp 3056–3065
38. Xie L, Hu Z, Cai X, Zhang W, Chen J (2021) Explainable recommendation based on knowledge graph and multi-objective optimization. *Complex Intell Syst* 1–12
39. Yan J, Cho M, Zha H, Yang X, Chu SM (2015) Multi-graph matching via affinity optimization with graduated consistency regularization. *IEEE Trans Pattern Anal Mach Intell* 38(6):1228–1242
40. Zafar A, Sminchisescu C (2018) Deep learning of graph matching. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp 2684–2693
41. Zhao B, Le X, Xi J (2019) A novel SDASS descriptor for fully encoding the information of a 3d local surface. *Inf Sci* 483:363–382
42. Zhao Y, Wei X, Li J, Wang G (2016) Star identification algorithm based on k-l transformation and star walk formation. *IEEE Sens J* 16(13):5202–5210

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.