IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received April 8, 2019, accepted May 5, 2019, date of publication May 10, 2019, date of current version May 24, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915976

False Star Filtering for Star Sensor Based on Angular Distance Tracking

GANGYI WANG, WENCHAO LV, JIAN LI, AND XINGUO WEI
School of Instrument Science and Opto-Electronics Engineering, Beihang University, Beijing 100191, China

Corresponding author: Jian Li (lijian_0355@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61605007 and Grant 61705005.

ABSTRACT The reliability of star sensor in a harsh environment has recently become a research hotspot. In some harsh environment such as plume interference, a large number of false stars can be observed, leading to failure in star identification. In this paper, we propose a false star filtering algorithm, which can be used as a preprocessing algorithm for any existing star identification algorithm. By utilizing the difference between the motion of false stars and true stars, the algorithm performs angular distance tracking and star voting on multiple consecutive frames of star images and achieves false star filtering. The software simulation results show that for the star images containing more than 700 false stars, the algorithm is able to find out all true stars in less than 10 frames, and the success rate of the algorithm remains high when the star sensor rotates at up to 1°/s. The algorithm is also implemented on an existing star senor and evaluated with star images generated by a dynamic star simulator. The experimental results indicate that with the help of the proposed algorithm, the robustness of a normal star identification algorithm can be significantly improved.

INDEX TERMS Star sensor, harsh environment, false star filtering, star identification.

I. INTRODUCTION

Star sensor is a kind of attitude measurement device widely used in satellites, missiles and other spacecraft. Compared with other attitude measurement devices, star sensor is of the highest measurement precision, making it irreplaceable in many tasks which require high-precision attitude information.

The basic principle of star sensors is to find the correspondence between observed stars and cataloged stars, and then determine their attitude [1]. After decades of development, modern star sensors have been able to work steadily under ideal working conditions [2], [3]. However, when a star sensor is disturbed by strong radiation, plumes, stray lights, etc., a large number of false stars may appear in the FOV (field of view) of star sensor, which seriously interferes with star identification, leading to failure in attitude determination. Since these harsh conditions seriously affect the reliability of star sensor, many researches have focused on improving the robustness of star sensor to false stars. According to their sources and movement characteristics, false stars can be divided into three classes, as follows.

The first class is the transient false stars. These false stars are usually introduced by radiation impinging particles [4]. When a large number of electrons or protons hit the imaging device, false stars can be generated at the corresponding positions. This phenomenon often occurs when a star sensor is in strong radiation zones such as the South Atlantic Anomaly, the Earth radiation belts, interstellar space, etc. The number of transient false stars can be up to several hundreds or more, but they tend to appear at random positions in different frames [5]. Utilizing this characteristic, some algorithms compare the star positions of two consecutive frames and only keep the stars appear at similar positions in both frames [6], [7], which can effectively filter transient false stars.

The second class is the stationary false stars. These false stars are usually introduced by faraway space objects which reflect sunlight and image in the FOV of star sensor. Similar to true stars, the stationary false stars do not move over time, making them difficult to distinguish. When the number of stationary false stars is small (significantly fewer than the true stars), star identification algorithms commonly used in existing star sensors, such as the pyramid algorithm [8], [9] and the grid algorithm [10], [11] can still ensure successful identification. However, when the number of false stars increases further, the success rate of the two algorithms

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

decreases significantly. To solve this problem, some recent star identification algorithms optimized specifically for false stars have been proposed [12]–[14]. These algorithms are still effective when the number of false stars is several times the number of true stars.

The third class is the drifting false stars. Nearby space objects may also be imaged by star sensor, thereby forming false stars on the image plane. For instance, the gas or particles emitted by the jets carried by some spacecraft can reflect sunlight and form a large number of false stars in the FOV of star sensor. This phenomenon is called plume interference [15]. As these objects are not faraway from the star sensor, their movement can be sensed by the star sensor, thus the false stars generated by these objects drift randomly and slowly on the image plane. The drifting false stars cannot be filtered with the methods used to deal with transient false stars, because they usually appear at similar positions of two consecutive star images. On the other hand, the number of drifting stars can be up to dozens of times the number of true stars, which makes almost all star identification algorithms unusable or too slow in practice [13], [16].

In summary, there have been algorithms which can cope with transient false stars and stationary false stars, but drifting stars still cannot be effectively processed, especially when the number of drifting false stars is very large. To solve this problem, we propose a drifting false star filtering algorithm in this paper. The algorithm itself is not a star identification algorithm, but a preprocessing algorithm before star identification to filter out false stars, thus improving the robustness of any star identification algorithm to false stars. By utilizing the difference between the motion of drifting false stars and true stars, the algorithm performs angular distance tracking and star voting on multiple consecutive frames of star images, and then finds out the stars with stable angular distances between each other, thereby obtaining true stars from a large number of false stars.

The remainder of this paper is organized as follows: in Section 2, the elementary idea and the details of the proposed algorithm are described; in Section 3, experiments on both simulation star images and real scenes generated by a dynamic star simulator are used to verify the robustness of the proposed algorithm; in Section 4, conclusions of the proposed algorithm are drawn.

II. ALGORITHM DESCRIPTION

In this section, we first analyze the motion characteristics of drifting false stars and find out the difference between drifting false stars and true stars, then design a false star filtering algorithm based on tracking of angular distances within multiple consecutive star images. The computational complexity of the algorithm is given at the end of this section.

A. GENERAL IDEA OF THE PROPOSED ALGORITHM

Drifting false stars are usually generated by nearby space objects such as plume that reflect sunlight and image in the FOV. The spots of such false stars are similar to true stars and are difficult to be distinguished with a single frame of star image. A set of real on-orbit star images captured by a star sensor is shown in Figure 1, where true stars are painted in blue and false stars are painted in red. There are 40 frames of star images in total. All the stars in these frames are drawn in the same coordinate system, thereby showing the trajectory of each star clearly. As shown in Figure 1, all true stars move in the image plane with similar trajectories due to the rotation of the star sensor, while the trajectories of false stars are more various, usually different from those of true stars. The reason for this phenomenon is simple: As the vectors of true stars are invariant in the inertial system, all star spots of the true stars in the image plane translate in the same way if the star sensor rotates around the X and Y axes of the image plane, making the trajectories of all true stars similar. On the other hand, the vectors of drifting false stars usually drift slowly in the inertial system, which makes their trajectories different from each other.

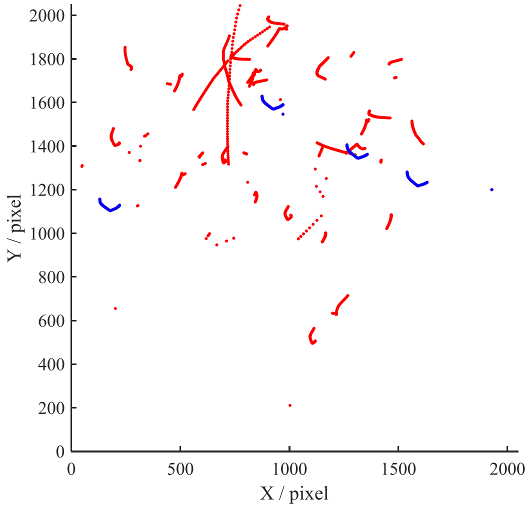


FIGURE 1. The trajectories of a set of real on-orbit stars, where true stars are painted in blue and false stars are painted in red.

Although the trajectories of true stars can be different when the star sensor rotates round the Z-axis of the image plane, the angular distance between each pair of true stars is always invariant. Based on this characteristic, it is feasible to filter false stars by using multiple frames of star images. Let N consecutive star images be numbered as frame $1, 2, \ldots, N$, and M stars observed in each star image be numbered as star $1, 2, \ldots, M$. Calculate the angular distance $d_{ij}^{(k)}$ of each pair of stars, where i, j denote star i and star j, k denotes frame k. If star i and star j are both true stars, the following formula is satisfied:

$$d_{ij}^{(k)} = d_{ij}^{(1)}, \quad k \in [1, N] \tag{1}$$

i.e., the angular distance between any pair of true stars keep unchanged in all frames. Conversely, if there is a frame k such that $d_{ij}^{(k)} \neq d_{ij}^{(1)}$, then at least one star in star i and star j is a false star. To further determine which star

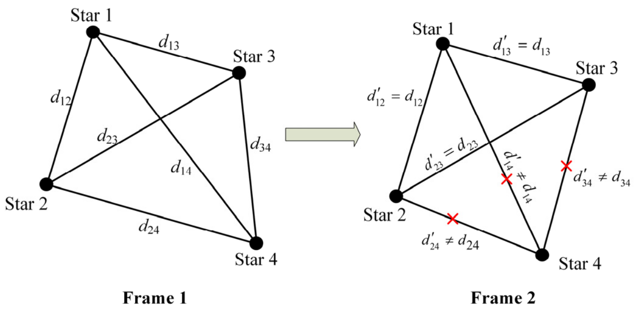


FIGURE 2. Illustration of false star filtering, where d_{ij} and d'_{ij} denotes the angular distance between star i and star j in two different frames respectively.

is a false star, it is also necessary to analyze the angular distance between the two stars and other stars. If any star does not satisfy Eq. (1) with most other stars, then the star is probably a false star. Figure 2 illustrates the principle of this idea with 4 stars, where d_{ij} and d'_{ij} denotes the angular distance between star i and star j in two different frames respectively. In frame 2, the angular distances between star 4 and the other three stars have changed, thus star 4 is determined to be a false star and filtered out. On the other hand, the angular distances between each pair of stars in star 1, 2, and 3 have not changed, thus the three stars are not filtered out.

According to this idea, a false star filtering algorithm based on angular distance tracking within multiple consecutive frames of star images is proposed in this paper. The basic flow of the algorithm is as follows:

(1) Take the first frame as the initial frame and record all the stars in the frame as the initial stars, and then initialize the angular distance filtering matrix, which records whether the angular distance between each pair of stars satisfies Eq. (1) in all frames.

(2) Whenever a new frame arrives, track the stars in the new frame and establish the correspondence between the initial stars and the stars in the new frame.

(3) Update the angular distance filtering matrix according to whether each angular distance in the new frame has changed compared with its correspondence in the original frame, which is named as "angular distance tracking".

(4) Find out false stars by voting to each star according to the angular distance filtering matrix.

(5) Output stars not determined as false stars and go back to step (2) to process the next frame.

The block diagram of the proposed algorithm is shown in Figure 3. Except for the first frame, each newly arrived frame is sequentially processed by three steps: star tracking, angular distance tracking and star voting. If a star is determined to be a false star in one frame, then the star is filtered out and not processed in any of the following frames. Therefore, the number of false stars decreases gradually over time until all the remaining stars are true stars. Details of each step are described in the following subsections.

B. INITIALIZATION AND STAR TRACKING

The core of the proposed algorithm is tracking the variances of angular distances within multiple consecutive frames to distinguish true stars and false stars. In order to track the angular distance of the same pair of stars in multiple frames, it is necessary to track stars first, i.e., to find the correspondence of the same star in different frames. We first present a method to track stars in two consecutive frames, and then extend the method to the multi-frame case.

As the angular velocity of a star sensor is usually limited, the angular distance between the star vectors corresponding to the same star in two consecutive frames is usually small. Based on this characteristic, the following process is designed to achieve star tracking for two consecutive frames:

(1) Let $\{\vec{x}_i^{(1)}\}$ and $\{\vec{x}_j^{(2)}\}$ be the two sets of stars in the old frame and new frame, respectively. Create a set $\{(i, j)\}$ of star pairs and initialize it to an empty set. The member (i, j) in the set denotes that star $\vec{x}_i^{(1)}$ in the old frame and star $\vec{x}_j^{(2)}$ in the new frame is the same star.

(2) Calculate the angular distance d_{ij} between each star in $\{\vec{x}_i^{(1)}\}$ and each star in $\{\vec{x}_j^{(2)}\}$, and then compare it with threshold δ_1, where δ_1 is named as "star tracking radius". If $d_{ij} < \delta_1$, it indicates that star $\vec{x}_i^{(1)}$ in the old frame is close enough to star $\vec{x}_j^{(2)}$ in the new frame, thus star pair (i, j) is added to set $\{(i, j)\}$. The angular distance d_{ij} can be calculated as follows:

$$d_{ij} = \frac{\arccos\left(\left\langle \vec{x}_i^{(1)}, \vec{x}_j^{(2)} \right\rangle\right)}{\left\| \vec{x}_i^{(1)} \right\| \left\| \vec{x}_j^{(2)} \right\|} \tag{2}$$

where $\langle\rangle$ represents inner product of two vectors, and $\|\|$ represents the norm of a vector.

(3) Check all the star pairs in set $\{(i, j)\}$. If any star i in the old frame appears more than once in the star pairs, it indicates that more than one star in the new frame is close enough to star i, thus we are unable to set up a unique correspondence for star i. In this case, all the star pairs including star i are removed from set $\{(i, j)\}$. For the same reason, if any star j in the new frame appears more than once in the star pairs, all the star pairs including star j are also removed from set $\{(i, j)\}$.

The star tracking process is illustrated in Figure 4, where the stars in the old frame are marked with "★" and indexed with 1, 2, 3 . . .; stars in the new frame are marked with "☆" and indexed with a, b, c . . . Only star a in the new frame is in the neighborhood of star 1, thus star pair $(1, a)$ is added to set $\{(i, j)\}$. There are 0 stars and 2 stars in the neighborhoods of star 2 and star 3 respectively, and there is a common star in the neighborhoods of star 4 and star 5, thus stars 2, 3, 4, 5 cannot find their one-to-one correspondences, i.e., tracking of the four stars is failed.

As described in Section 2.1, since the angular distances between stars in each frame should be compared with those in the original frame, we need to set up the correspondence of stars between each newly arrived frame and the original frame. However, the tracking method for two consecutive

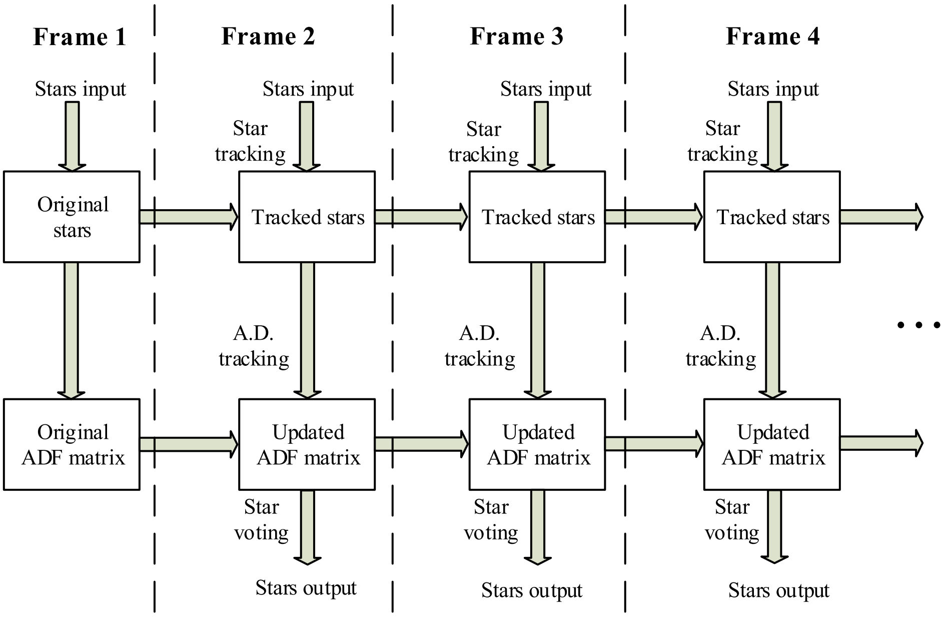


FIGURE 3. Block diagram of the proposed algorithm, where A.D. is short for angular distance and ADF is short for angular distance filtering.

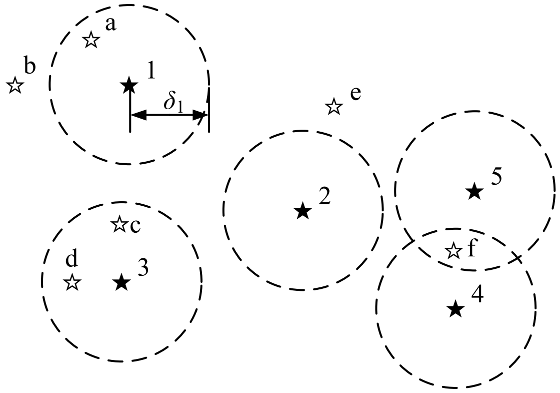


FIGURE 4. Illustration of the star tracking process in two consecutive frames.

frames is not suitable for this case because a star vector usually continuously moves away from its original position with the rotation of the star sensor. To solve this problem, we set up the correspondence indirectly as follows:

(1) Use the first frame as the original frame and create the original star vector set $\{\vec{x}_i^{(1)}\}$ with all the stars in the original frame. In addition, create the false star set $\{b_i\}$, which records all the stars in the original frame determined as false stars. $\{b_i\}$ is initialized to be an empty set.

(2) When the second frame arrives, set up the correspondence of the stars between the second frame and the original frame with the method above. The correspondence is represented as $f^{(2)}: \{\vec{x}_i^{(2)}\} \rightarrow \{\vec{x}_i^{(1)}\}$.

(3) For each newly arrived frame k (where $k \geq 3$), the correspondence between frame k and frame $k-1$ is first set up, which is represented as $g : \{\vec{x}_i^{(k)}\} \rightarrow \{\vec{x}_i^{(k-1)}\}$. After that, it is combined with the correspondence $f^{(k-1)}: \{\vec{x}_i^{(k-1)}\} \rightarrow \{\vec{x}_i^{(1)}\}$ that has already set up for frame $k-1$, thus the correspondence $f^{(k)}$ can be obtained with $f^{(k)} = g \circ f^{(k-1)}$, i.e., the correspondence of the stars between frame k and the original frame is set up through the process $\{\vec{x}_i^{(k)}\} \rightarrow \{\vec{x}_i^{(k-1)}\} \rightarrow \{\vec{x}_i^{(1)}\}$.

After the three steps above, there may be still some stars in the original frame that cannot find their correspondence in frame k, i.e., these stars failed to be tracked in frame k. These stars are added to the false star set $\{b_i\}$ and not used in the following process.

C. ANGULAR DISTANCE TRACKING AND STAR VOTING

As described previously, the star tracking step can distinguish false stars in some extent. The transient false stars and some quick drifting false stars may be filtered out as they cannot be successfully tracked within multiple frames. However, there can be still many false stars that drift slowly enough to pass the star tracking step. To filter these false stars, tracking of angular distances within multiple frames is needed.

For reliable star identification, it is usually required that a minimum number N of true stars should be observed by the star sensor. The exact value of N is related to the star identification algorithm, generally not less than 4. Therefore, a star sensor is usually designed to be able to observe at least N stars with its boresight pointed to any direction. Based on this prerequisite, for any true star, there should be at least $N-1$ other stars which keep the angular distance with the star invariant in all frames, otherwise, the star must be a false star. Based on this principle, we design the following angular distance tracking and star voting method.

TABLE 1. An example of angular distance tracking and star voting, where A.D. is short for angular distance, shaded cells represent that the stars in the corresponding lines has been filtered out.

	Frame 2		Frame 3		Frame 4	
Star index	Tracked A.D.	Votes	Tracked A.D.	Votes	Tracked A.D.	Votes
1 (true)	2, 3, 4, 5, 6, 7	6	2, 3, 4, 6, 7	5	2, 3, 4, 6	4
2 (true)	1, 3, 4, 5, 6, 7	6	1, 3, 4, 6, 7	5	1, 3, 4, 6	4
3 (true)	1, 2, 4, 6	4	1, 2, 4, 6	4	1, 2, 4	3
4 (true)	1, 2, 3, 6	4	1, 2, 3, 6	4	1, 2, 3	3
5 (false)	1, 2	2	-	-	-	-
6 (false)	1, 2, 3, 4, 7	5	1, 2, 3, 4	4	1, 2	2
7 (false)	1, 2, 6	3	1, 2	2	-	-

(1) When the first frame arrives, the following initialization steps are performed:

(i) Create the angular distance filtering matrix $\mathbf{F} = [f_{ij}]$, where each member f_{ij} records whether the angular distance between stars $\vec{x}_i^{(1)}$ and $\vec{x}_j^{(1)}$ is invariant in all arrived frames. Each f_{ij} is initialized to be 1.

(ii) Create the voting vector $\mathbf{H} = [h_i]$, where each member h_i records the votes that star $\vec{x}_i^{(k)}$ gets in frame k. All members in $\mathbf{H}$ are initialized to be 0.

(2) When frame k arrives, perform angular distance tracking for each pair of tracked stars $\vec{x}_i^{(k)}$ and $\vec{x}_j^{(k)}$ with the following two rules:

(i) Get stars $\vec{x}_{i'}^{(1)}$ and $\vec{x}_{j'}^{(1)}$ in the initial frame which respectively correspond to $\vec{x}_i^{(k)}$ and $\vec{x}_j^{(k)}$ according to correspondence $f^{(k)}$. The member $f_{i'j'}$ in the angular distance filtering matrix F should be 1, i.e., the angular distance between this pair of stars keeps invariant in all frames before frame k.

(ii) Calculate the angular distance $d_{ij}^{(k)}$ between $\vec{x}_i^{(k)}$ and $\vec{x}_j^{(k)}$, the angular distance $d_{i'j'}^{(1)}$ between $\vec{x}_{i'}^{(1)}$ and $\vec{x}_{j'}^{(1)}$. The two angular distances should satisfy $\left|d_{ij}^{(k)} - d_{i'j'}^{(1)}\right| < \delta_2$, i.e., the angular distance between this pair of stars is close enough to that in the initial frame. Otherwise, the member $f_{i'j'}$ in F is updated to be 0.

(3) For a pair of stars $\vec{x}_i^{(k)}$ and $\vec{x}_j^{(k)}$ satisfying the two rules above, each one of the two stars get one vote, i.e., h_i and h_j are increased by 1, respectively.

(4) After step (2) and (3) are performed for all pairs of stars, examine each member h_i in $\mathbf{H}$. If $h_i < N - 1$, then star $\vec{x}_i^{(k)}$ is determined as a false star, and its correspondence $\vec{x}_i^{(1)}$ in the initial frame is added to the false star set $\{b_i\}$.

After the four steps above, all the stars not included in the false star set $\{b_i\}$ are output for star identification. With more and more frames arrived, the false star set $\{b_i\}$ and angular distance tracking matrix $\mathbf{F}$ are updated continuously, making the ratio of true stars increases monotonously until all the stars output are true stars.

An example of angular distance tracking and star voting is given in Table 1. There are seven stars passed the star tracking step in four consecutive frames of star images. The first frame is used as the initial frame, and the following three frames are used to perform angular distance tracking and star voting sequentially. The stars are indexed as 1$\sim$7, in which stars 1$\sim$4 are true stars and stars 5$\sim$7 are false stars. In frame 2, there are two numbers in the "Tracked A.D." column in the row of star 5, which represents that the angular distances between star 5 and star 1, star 5 and star 2 are successfully tracked. As there are only two tracked angular distances, star 5 gets only 2 votes. In this example, we set $N = 4$, i.e., a star should get at least 3 votes to pass the star voting step, thus star 5 is filtered out in frame 2 while the other six stars are not, because they get at least 3 votes. The same operations are also performed in frame 3 and frame 4, where star 6 and star 7 are filtered out sequentially since they cannot get enough votes. In this way, all the three false stars are filtered out in 4 frames by performing angular distance tracking and star voting.

D. COMPUTATIONAL COMPLEXITY ANALYSIS

The proposed algorithm filters false stars gradually in multiple frames. Except for the initial frame, all the frames are processed with three steps, including star tracking, angular distance tracking and star voting. Suppose the number of tracked stars in a frame is N, then both star tracking and angular distance tracking need to calculate angular distance according to Eq. (2) for $O(N^2)$ times. In Eq. (2), there are one dot product, one division, two normalizations, one inverse cosine operation in total. As the star vectors can be normalized before Eq. (2), the total number of normalization can be reduced to N times, which is neglectable compared with other operations which are $O(N^2)$ times. One dot product consists of three multiplications, two additions, while one inverse cosine operation is equal to six multiplications [17]. Suppose the computational complexity of multiplication, addition and division are all 1, then the complexity of star tracking and angular distance tracking are both $O(N^2)$. As there are $O(N^2)$ additions in star voting, its complexity is $O(N^2)$. In summary, the computational complexity of the proposed algorithm for each frame is $O(N^2)$. Since the number of false stars which are filtered out increases with the increasing of processed frames, the number of tracked stars N decreases gradually

TABLE 2. Parameters of the optical system for simulation.

Parameters	Value
FOV	ϕ14.5°
Focal length	44 mm
Resolution	2048×2048 pixels
Max. magnitude	6.0

TABLE 3. Parameters of the proposed algorithm.

Parameters	Description	Value
δ_1	Star tracking radius	0.125°
δ_2	Angular distance tracking radius	20″
N	Min. number of true stars	4

until all the false stars are filtered out. Therefore, the computational complexity of the proposed algorithm decreases gradually from $O(N^2)$ to $O(M^2)$, where M is the number of true stars.

III. EXPERIMENTAL RESULTS

To evaluate the performance of the proposed algorithm, we tested the proposed algorithm on simulation star images with different number and drifting velocity of false stars, and different rotation velocity of star sensor. Moreover, we also implemented the algorithm on an existing star sensor and tested it with real scenes generated with a dynamic star simulator.

A. PARAMETER SELECTION

The parameters of the optical system used to generate simulation star images are listed in Table 2.

Except for the parameters of the optical system, it is also necessary to determine the parameters of the proposed algorithm. All the parameters are listed in Table 3, which are determined in the following way:

(1) δ_1 is the maximum allowed angular distance that two stars in two consecutive frames can be determined as the same star. The larger the parameter is, the larger the rotation angle for a star between two consecutive frames is allowed, i.e., star tracking is robust to larger rotation velocity of the star sensor. However, larger δ_1 could also increase the probability that more than one star falls into the neighborhood of one star, making it difficult to find out one-to-one correspondence, just like the stars 3, 4, and 5 in Figure 4. Therefore, δ_1 should be determined according to the maximum rotation velocity and the frame rate. In the simulation experiments, the maximum rotation velocity of the star sensor is limited below 1°/s and the frame rate is fixed at 8Hz, which ensures the rotation angle between two consecutive frames smaller than 0.125°. Therefore, the angular distance between the same star in two consecutive frames must be no larger than 0.125°, thus we set $\delta_1 = 0.125°$.

(2) δ_2 is the maximum allowed gap between the angular distances of the same pair of stars in different frames. This parameter is relative to the accuracy of the star sensor. If the standard deviation for the star sensor to measure a star vector is σ, then the standard deviation for the angular distance between two stars is about $\sqrt{2}\sigma$. Therefore, if the variation of the angular distance between a pair of stars is larger than multiple times of $\sqrt{2}\sigma$, then at least one of the two stars is a false star. In the experiments, we suppose $\sigma = 2''$, and set $\delta_2 = 20''$, which is more than 7 times of $\sqrt{2}\sigma$, making it almost impossible that the angular distance between a pair of true stars changes larger than δ_2.

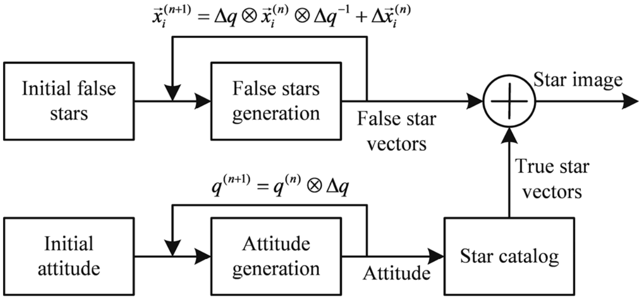


FIGURE 5. Simulation star image generation process.

(3) N is the minimum number of true stars in the FOV. Considering that at least 4 true stars are generally required for reliable star identification [8], [9], we choose $N = 4$. As this parameter determines the minimum votes for a true star in the star voting step, larger N makes it more difficult for a false star to get enough votes, thus the false stars are easier to be filtered out. Therefore, if a star sensor ensures that the number of observable true stars is larger than 4 for any boresight direction, N can be larger accordingly to improve the ability to filter out false stars.

B. EXPERIMENTS ON SIMULATION STAR IMAGES

Three simulation experiments are designed in this section, which evaluate the performance of the proposed algorithm for different number of false stars, different drifting velocity of false stars and different rotation velocity of the star sensor, respectively. In the following sub-subsections, generation of simulation star images and the evaluation criteria of performance are first introduced, and then the experimental results and discussion of the three experiments are given respectively.

1) GENERATION OF SIMULATION STAR IMAGES

The simulation star image generator outputs multiple frames of star images which include a number of drifting false stars. For any two consecutive frames, the attitude distance should be no larger than the star tracking radius δ_1. Each false star persists and drifts slowly in multiple frames. The block diagram of the simulation star image generator is shown in Figure 5, which consists of two main threads: the true star generation thread generates a random attitude for the initial frame first, according to which generates the star vectors of the true stars, and then rotates the attitude of each subsequent

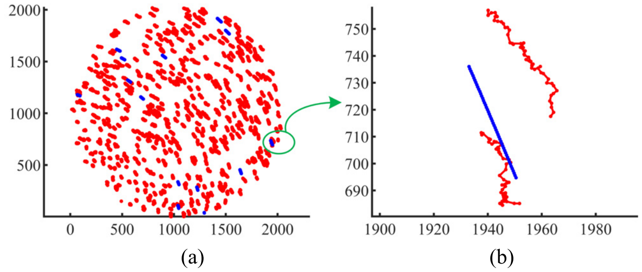


FIGURE 6. Illustration of a set of simulation star images. (a) The trajectories of all stars in all frames, where the true stars are painted in blue and the false stars are painted in red. (b) is a partial enlargement of (a). At the fixed rotation velocity, the trajectory of each true star is a smooth curve, while the trajectory of each false star superimposes small drifts on the smooth curve.

frame by a fixed Δq (where Δq is a quaternion) based on the previous frame; the false star generation thread randomly generates vectors of all false stars for the initial frame, and in each subsequent frame the vectors are first rotated by Δq, then added with an small extra displacement to make them drift slowly. Finally, the true stars and false stars are mixed together to generate the simulation star images. Figure 6 illustrates the trajectories of all stars in a set of simulation star images, where the star sensor is rotated at 0.16°/s, the data frame is 8Hz, and the number of frames is 50. The false stars are about 35 times more than the true stars.

2) PERFORMANCE CRITERIA

The proposed algorithm can be regarded as an iteration process on multiple consecutive star images. The iteration stops when all the false stars are filtered out or the number of stars is less than the minimum allowed number of true stars N. We evaluate the performance of the algorithm with the following two criteria:

(i) The probability of successful convergence p_c. If all the false stars are filtered out after iteration, and the number of true stars is not less N, then the algorithm converges successfully.

(ii) The number of frames f_c required for the algorithm to converge. Smaller f_c indicates that the algorithm converges faster, i.e., the star sensor can determine its attitude in less time.

3) ROBUSTNESS TO THE NUMBER OF FALSE STARS

The ability to filter out false stars is the core goal of the proposed algorithm. Since the number of drifting false stars can be up to several hundreds in practice, we tested the algorithm with the number of false stars ranging from 10 to 1000. The rotation velocity of the star sensor is fixed at 0.16°/s, and the drifting velocity of the each false star conforms to the Gaussian distribution with a mean of 0 and a standard deviation of 25″/frame. For each number of false stars, 1000 sets of consecutive images are generated, with which p_c and f_c are counted. The results are shown in Figure 7 and Figure 8.

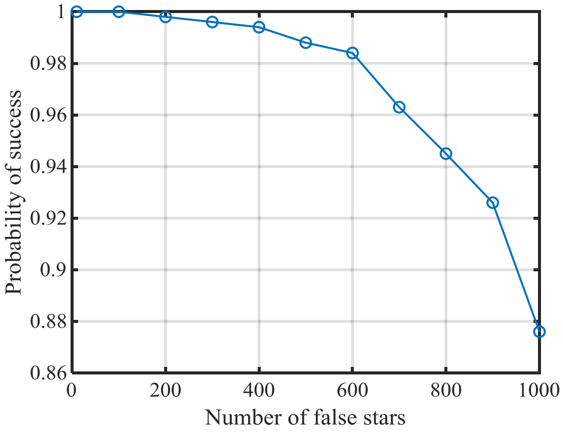


FIGURE 7. Probability of successful convergence for different number of false stars.

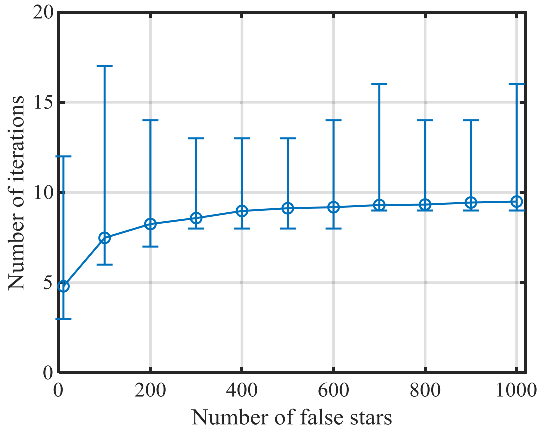


FIGURE 8. Number of frames required to converge for different number of false stars, where the circle, the upper end and the lower end of each case correspond to the mean, maximum and minimum number of frames, respectively.

As shown in Figure 7, the proposed algorithm is quite robust to false stars: when the number of false stars is not more than 100, the success probability p_c is 100%, which is only slightly decreased to 98.3% when the number reaches 600. Even when the number of false stars is as high as 1000, p_c is still more than 87%. The main reason for the failed cases is that with the number of false stars getting larger, the stars in FOV become more and more dense, increasing the probability that true stars failed to be tracked, which leads to the decrease of p_c.

As shown in Figure 8, the required number of frames f_c for convergence increases slowly with the increase of the number of false stars, but the maximum of f_c keeps lower than 20, which indicates that the convergence speed of the algorithm is quite stable. The reason is that the algorithm distinguish false stars mainly based on angular distance drifting of false stars. With the increase of newly arrived frames, the variation of the angular distance of false stars gradually increases in a random walk process, which makes the false stars filtered out at a nearly constant ratio in each new frame, thus the curve of f_c is similar to a logarithm function and increases slowly.

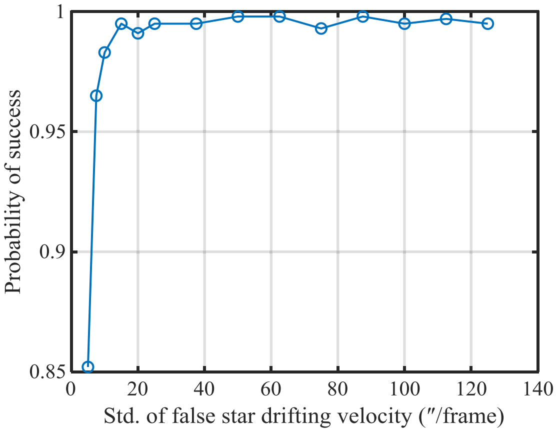


FIGURE 9. Probability of successful convergence for different drifting velocity of false stars.

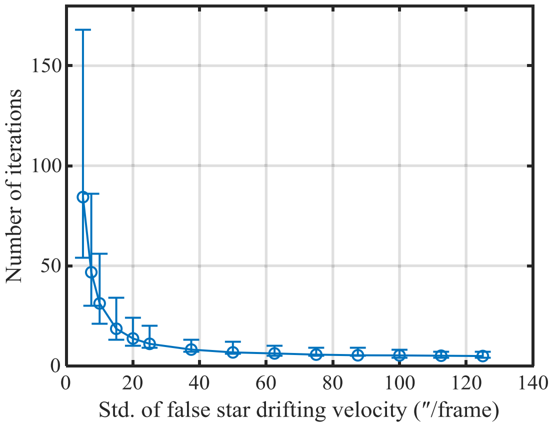


FIGURE 10. Number of frames required to converge for different drifting velocity of false stars, where the circle, the upper end and the lower end of each case correspond to the mean, maximum and minimum number of frames, respectively.

4) ROBUSTNESS TO THE DRIFTING VELOCITY OF FALSE STARS

The drifting velocity of false stars also affects the performance of the algorithm. In this experiment, the drifting angle for each star conforms to a Gaussian distribution, and the algorithm is tested in the cases that the standard deviation of the drifting velocity ranges from 5″/frame to 125″/frame. The number of false stars is fixed at 500 and the rotation velocity of the star senor is fixed at 0.16°/s. For each drifting velocity, 1000 sets of simulation star images are generated, with which p_c and f_c are counted. Figure 9 and Figure 10 show the experimental results.

When the false stars drift slowly, the angular distance of false stars also changes slowly, making it requires more frames to accumulate enough variance of angular distance. Therefore, the number of frames f_c required to converge is larger when the false stars drift slower, as shown in Figure 10.

In the case that the standard deviation of drifting velocity is only 5″/frame, as shown in Figure 9 and Figure 10, the success probability p_c is about 85%, while f_c reaches 168 frames in the worst case. The decline of p_c is directly caused by the increase of f_c: the probability that a true star keeps being tracked decreases with the increase of frames, making the algorithm easier to fail due to the number of true stars less than the minimum required number of true stars N. With the drifting velocity increases, f_c decreases significantly, thus p_c also increases accordingly. When the drifting velocity is not less than 15″/frame, f_c drops to less than 20, and p_c remains higher than 99% in these cases.

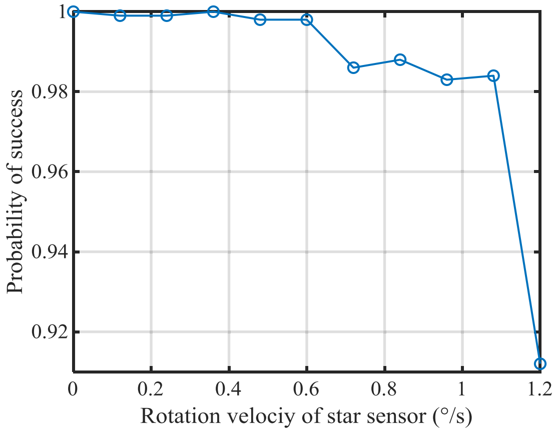


FIGURE 11. Probability of successful convergence for different rotation velocity of the star sensor.

To make the algorithm effective at lower drifting velocity of false stars, a feasible method is to use smaller δ_2. A smaller δ_2 makes the algorithm more sensitive to the variance of angular distance, which makes a false star distinguishable by the algorithm even if it only drifts a very small angle. Therefore, if the accuracy of the star sensor is high enough that ensures the error of measured angular distance between true stars always smaller than δ_2, then the success probability p_c can be improved by using smaller δ_2.

5) ROBUSTNESS TO ROTATION VELOCITY OF THE STAR SENSOR

The rotation velocity can also affect the performance of the algorithm. In this experiment, we test the algorithm with the rotation velocity of the star sensor ranging from 0°/s to 1.2° /s. The number of false stars is fixed at 500, and the drifting velocity of the false stars conforms to the Gaussian distribution with a mean of 0 and a standard deviation of 25″/frame. For each rotation velocity, 1000 sets of consecutive star images are generated, with which p_c and f_c are counted. The experimental results are shown in Figure 11 and Figure 12.

As shown in Figure 11, when the rotation velocity of the star sensor is not higher than 0.6°/s, the success probability p_c keeps higher than 99.5%. When the rotation speed increases to 1.1°/s, p_c decreases slightly to 98%. The reason for the decline of p_c is that some true stars gradually moves out of FOV when the rotation velocity is high, which can lead to the number of true stars smaller than N in some cases.

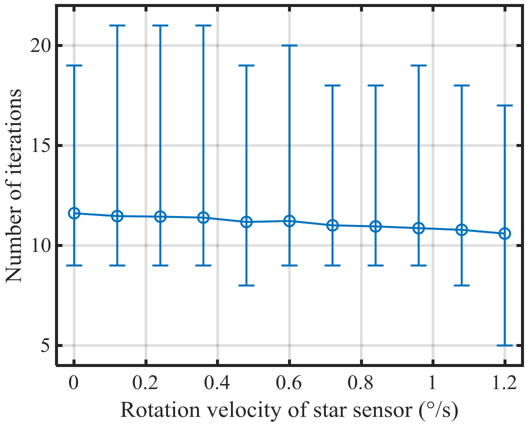


FIGURE 12. Number of frames required to converge for different rotation velocity of the star sensor, where the circle, the upper end and the lower end of each case correspond to the mean, maximum and minimum number of frames, respectively.

When the rotation velocity increases to 1.2° /s, p_c decreases significantly from higher than 98% to 91%. The reason is that the rotation angle of the same star between two consecutive frames is larger than the star tracking radius δ_1 when the rotation velocity is too high, making the true stars hard to be tracked. It can be predicted that tracking of true stars will be harder with further increase of the rotation velocity, making p_c decrease further. According to Fig 12, the number of frames f_c required to converge does not changes obviously with the rotation velocity of the star sensor.

The star tracking ability can be improved by using larger δ_1, thereby improving the robustness to the rotation velocity of the star sensor. However, the robustness to the number of false stars will decline when δ_1 is too large. The reason is that larger δ_1 will increase the probability that a false star falls within the tracking radius of a true star when the number of false stars is large, leading to tracking of the true star failed. Therefore, the robustness to rotation velocity and the robustness to the number of false stars is mutually exclusive to some extent. In practice, the best working parameters should be selected according to the specific working conditions.

C. EVALUATION ON AN EXISTING STAR SENSOR

We have also implemented the proposed the algorithm on an existing star sensor to evaluate the proposed algorithm further. The complete test bench consists of a star sensor, a dynamic star simulator and a PC, as shown in Figure 13. The dynamic star simulator generates star scenes including both true stars and false stars; the star sensor captures the star images and perform false star filtering with the proposed algorithm and star identification; the PC receives the attitude measured by the star sensor with a RS-422 interface.

Different from the software simulation experiments, the performance on the existing star sensor is evaluated in a more straightforward way. Each scene generated by the dynamic star simulator is processed in two cases. In the first case, stars captured by the star sensor are first filtered by the proposed algorithm and then identified with a star identification algorithm, while in the second case, star identification is directly performed without star filtering by the proposed algorithm. The successful identification rates of the two cases are counted and compared, where we say star identification is successful when the attitude output by the star sensor is consistent with the attitude given to the dynamic star simulator.

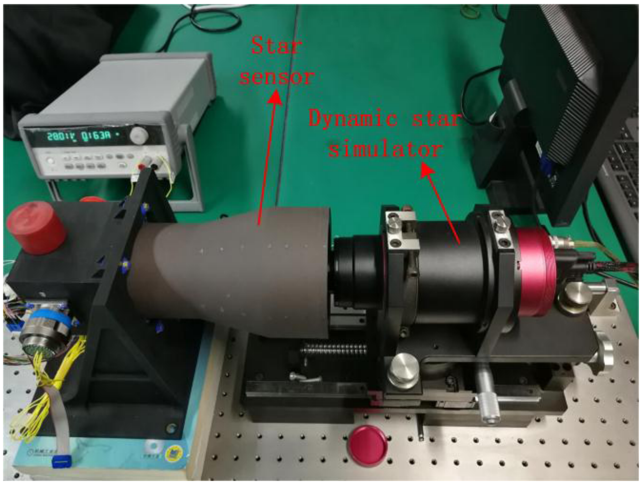


FIGURE 13. The complete test bench for the experiment on an existing star sensor.

The successful identification rates in the scenes with different number false stars are counted in the experiment. Limited by the computational ability (100 MHz ARM processor) and the amount of memory (128 KB) of the star sensor, the maximum number of false stars in one scene is set at 200. For each case, 1000 scenes with different initial attitude are randomly selected. Each scene rotates slowly and is continuously displayed for 10 seconds, which can be imaged by the star sensor for about 80 times. The rotation velocity of each scene is randomly selected, which conforms to a uniform distribution between 0° /s and 1° /s. False stars in the scenes drift randomly at the velocity conforming to a uniform distribution between 0.02° /s and 0.3° /s. A sample image captured by the star sensor in one of the scenes is shown in Figure 14, which includes 9 true stars and 155 false stars.

The star identification algorithm adopted in the experiment is the pyramid algorithm [8]. The algorithm is of high robustness to false stars and is widely used in star sensor. The identification results by the algorithm with and without false star filtering by the proposed algorithm are shown in Figure 15. It can be seen that without the help of the proposed algorithm, the identification rate of the pyramid algorithm decreases dramatically with the increasing of the number of false stars. When the number is as high as 200, the pyramid algorithm is almost unusable. By contrast, with the help of the proposed algorithm, nearly all the false stars are filtered out before they are sent to the pyramid algorithm, making the identification rate remains nearly 100% for the images including up to 200 false stars.

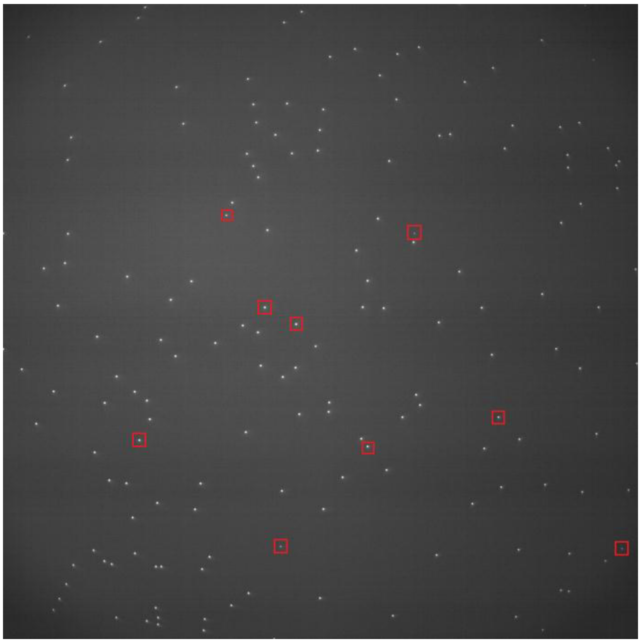


FIGURE 14. A sample image captured by the star sensor in one of the scenes. True stars are circled with red squares, and all the other stars are false stars.

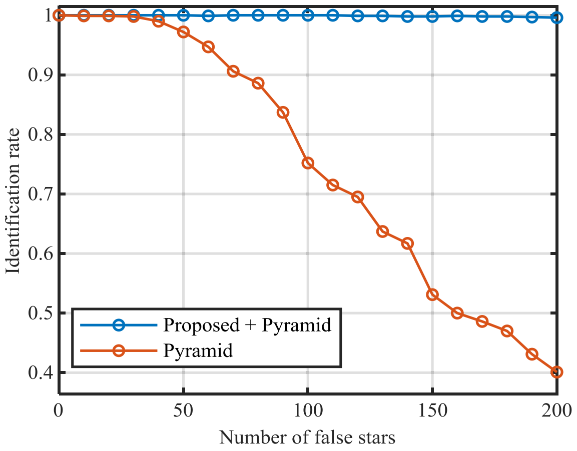


FIGURE 15. Identification results by the pyramid algorithm with and without false star filtering by the proposed algorithm.

IV. CONCLUSIONS

A false star filtering algorithm is proposed in this paper to make star sensor robust to drifting false stars. By utilizing the difference between the motion of drifting false stars and true stars, the algorithm achieves distinguishing true stars from hundreds of false stars, which is of great help for star identification. Experiments on simulation star images indicate that the algorithm is quite robust to the number of false stars. On average, the algorithm is able to find out all true stars from star images containing up to 700 false stars in no more than 10 consecutive frames. Moreover, the success probability of the algorithm is still more than 98% when the star sensor rotates at 1° /s. Another experiment on an existing star sensor indicates that with the proposed algorithm to filter out false stars, the robustness of a normal star identification algorithm can be significantly improved. In the future, we plan to further evaluate the algorithm with on-orbit experiments.

REFERENCES

[1] C. C. Liebe, "Accuracy performance of star trackers—A tutorial," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 38, no. 2, pp. 587–599, Apr. 2002.

[2] A. Greenbaum, T. Brady, and C. J. Dennehy, "Finding the gaps in space GNC hardware," in *Proc. IEEE Aerosp. Conf.*, Mar. 2014, pp. 1–15.

[3] C. C. Liebe *et al.*, "Micro APS based star tracker," in *Proc. IEEE Aerosp. Conf.*, Mar. 2002, p. 5.

[4] J. L. Jørgensen, G. G. Thuesen, M. Betto, and T. Riis, "Radiation impacts on star-tracker performance and vision systems in space," *Acta Astronautica*, vol. 46, nos. 2–6, pp. 415–422, 2000.

[5] G. Berrighi and D. Procopio, "Star sensors and harsh environment," in *Proc. Int. ESA Conf. Spacecraft Guid., Navigat. Control Syst.*, 2003, pp. 19–25.

[6] F. Boldrini, D. Procopio, S. P. Airy, and L. Giulicchi, "Miniaturised star tracker (AA-STR) ready to fly," in *Proc. 4S Symp. Small Satell., Syst. Services*, vol. 571, Nov. 2004, p. 571.

[7] M. Lauer, L. Jauregui, and S. Kielbassa, "Operational experience with autonomous star trackers on ESA interplanetary spacecraft," in *Proc. 20th Int. Symp. Space Flight Dyn.*, Sep. 2007, pp. 1–6.

[8] D. Mortari, M. A. Samaan, C. Bruccoleri, and J. L. Junkins, "The pyramid star identification technique," *Navigation*, vol. 51, no. 3, pp. 171–183, 2004.

[9] B. B. Spratling and D. Mortari, "A survey on star identification algorithms," *Algorithms*, vol. 2, no. 1, pp. 93–107, Jan. 2009.

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

[11] M. Na, D. Zheng, and P. Jia, "Modified grid algorithm for noisy all-sky autonomous star identification," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 45, no. 2, pp. 516–522, Apr. 2009.

[12] T. Delabie, T. Durt, and J. Vandersteen, "Highly robust lost-in-space algorithm based on the shortest distance transform," *J. Guid., Control, Dyn.*, vol. 36, no. 2, pp. 476–484, Mar/Apr. 2013.

[13] V. Schiattarella, D. Spiller, and F. Curti, "A novel star identification technique robust to high presence of false objects: The multi-poles algorithm," *Adv. Space Res.*, vol. 59, no. 8, pp. 2133–2147, 2017.

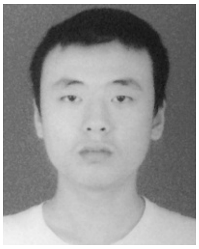
[14] G. Wang, J. Li, and X. Wei, "Star identification based on hash map," *IEEE Sensors J.*, vol. 18, no. 4, pp. 1591–1599, Nov. 2018.

[15] M. M. Scase, C. P. Caulfield, S. B. Dalziel, and J. C. R. Hunt, "Time-dependent plumes and jets with decreasing source strengths," *J. Fluid Mech.*, vol. 563, no. 563, pp. 443–461, 2006.

[16] C. C. Liebe *et al.*, "Star tracker design considerations for the Europa Orbiter mission," in *Proc. IEEE Aerosp. Conf.*, vol. 2, Mar. 1999, pp. 67–81.

[17] R. P. Brent, "Multiple-precision zero-finding methods and the complexity of elementary function evaluation," in *Analytic Computational Complexity*. New York, NY, USA: Academic, 1976, pp. 151–176.

GANGYI WANG received the bachelor's degree in electronics and information engineering, the master's degree, and the Ph.D. degree in information and communication engineering from the Harbin Institute of Technology, in 2006, 2008, and 2013, respectively. He is currently a Lecturer with the School of Instrumentation Science and Opto-Electronics Engineering, Beihang University, China. His research interests include star tracker, computer vision, and embedded systems.



WENCHAO LV received the bachelor's degree in opto-electronics and information engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2017. He is currently pursuing the master's degree with the School of Instrumentation Science and Opto-Electronics Engineering, Beihang University, China. His research interests include star tracker and precision measurement.

JIAN LI received the bachelor's and Ph.D. degrees in precision instrument from Beihang University, Beijing, China, where he is currently pursuing the master's degree with the School of Automation Science and Electrical Engineering. His research interests include the attitude determination sensing systems and image processing.

XINGUO WEI was born in 1977. He received the bachelor's, master's, and Ph.D. degrees from the School of Automation Science and Electrical Engineering, Beihang University, China, where he is currently a Professor with the School of Instrumentation Science and Opto-Electronics Engineering. He has authored over 40 articles, and over ten inventions. His research interests include precision measurement, machine vision, and image processing.