AN IMPROVED END-TO-END STAR MAP RECOGNITION METHOD FOR AEROSPACE VEHICLE

Yuan Zhang[1], Yuhe Yang[2], Qing Xiao[2], Rui Li[1]& Zhi Xu[1]

[1]School of Astronautics, Northwestern Polytechnical University, Xian 710072, China

[2]China Academy of Launch Vehicle Technology, Beijing 100076, China

Abstract

To address the star pattern recognition challenges arising from low signal-to-noise ratios of star points due to short exposure times of star sensors during the dynamic flight of aerospace vehicles, this paper proposes a convolutional neural network-based end-to-end star pattern recognition algorithm. According to the attitude information provided by the inertial navigation system, the sky region of the star to be identified was predicted, and the recognition pattern was input into the network model of the sky region. The star number to be identified is output by the network classifier to complete the star map recognition. This approach reduces model complexity and significantly enhances the accuracy, speed, and robustness of star pattern recognition.

Keywords: end-to-end recognition; motion blur; star map restoration; aerospace vehicle;

1. Introduction

The navigation system plays a crucial role in enabling aerospace vehicles to accomplish precise long-range flight missions. Inertial navigation, characterized by its high reliability, resistance to interference, strong autonomy, and high update frequency, serves as the core navigation equipment for aerial vehicles. However, the error accumulation in inertial navigation over time often fails to meet the requirements for precise navigation during long-duration flights. Additionally, satellite navigation systems are susceptible to electronic interference and may even be completely compromised during high-intensity information warfare. Therefore, starlight navigation, which offers high precision, long endurance, all-weather autonomy, and continuous positioning, serves as a "fallback" method to assist inertial navigation[1].

Starlight navigation requires observing the navigational stars recorded in the star catalog to determine the attitude of the vehicle. Therefore, the challenge of star pattern recognition lies in accurately and quickly identifying which star in the star catalog corresponds to each star point in the star image[2]. Currently, star pattern recognition algorithms can be broadly classified into three categories: subgraph isomorphism algorithms, pattern recognition algorithms, and artificial intelligence algorithms[3].

The subgraph isomorphism algorithm considers the star pattern recognition problem as a special case of the subgraph isomorphism problem. This type of star pattern recognition algorithm takes the star points within the field of view as the vertices of the graph, with the angular distances between the star points as the edges, constructing an undirected graph. By searching for a subgraph in the navigation database that is consistent with the observed undirected graph, star pattern recognition can be achieved. The most typical subgraph isomorphism algorithm is the triangle algorithm. The earliest version of the triangle algorithm was proposed by Liebe in 1996[4]. Based on this algorithm, numerous improved versions of the triangle algorithm have been developed subsequently[5].

Pattern recognition algorithms construct a unique pattern or signature for each navigational star, typically determined by the distribution of its neighboring star points. The navigational star in the database whose pattern most closely matches the observed pattern is identified as the corresponding star point. The grid algorithm, proposed by Padgett et al. in 1997, is one of the earliest pattern recognition matching algorithms[6]. The grid algorithm represents the relative positions of neighboring stars around the primary star using grid features for identification. The grid algorithm exhibits fast processing speed, high recognition rates, and strong robustness against positional noise, but it requires careful selection of calibration stars.

With the rapid advancement of deep learning technologies, artificial intelligence-based star pattern recognition algorithms have emerged as a focal point of contemporary research[8]. These algorithms primarily leverage neural networks and intelligent algorithms to identify geometric distribution features surrounding star points[8]. Neural networks exhibit exceptional generalization and clustering capabilities, enabling them to identify the closest prototype from incomplete or distorted observational patterns[9]. Furthermore, once trained, these networks demonstrate extremely fast online matching speeds[10].

The starlight navigation system, based on wide-field-of-view star sensors for aerial vehicles, faces a primary technical challenge during flight. Under dynamic conditions, the star sensor may experience issues such as starlight being overwhelmed by noise or trailing phenomena, which affects normal starlight observations and poses challenges to maintaining stable attitude determination for starlight navigation during dynamic flight. For this purpose, this paper addresses the issues of insufficient starlight navigation response capability and low attitude determination accuracy in integrated navigation systems during the dynamic flight of aerospace vehicles. We delve into the observational and computational mechanisms of wide-field star sensors to enhance the update rate of astrometric computation data. By developing intelligent star pattern processing algorithm strategies, we aim to meet the requirements for fully autonomous, high-precision, highly reliable, and real-time navigation of aerospace vehicles.

2. Measurement principle and measurement error of star sensor

2.1 Measurement principle of star sensor

To achieve starlight navigation for attitude determination, the primary task is to determine the pixel coordinates of stars on the imaging plane of the star sensor. Typically, each star occupies a 3×3 or 5×5 pixel window, necessitating the use of a centroid extraction algorithm to compute precise sub-pixel coordinates of the stars. The centroid method is commonly employed[11]. This algorithm calculates the precise sub-pixel coordinates of stars by weighted averaging the grayscale values of multiple pixels within the star's coverage area. It offers simplicity in computation while ensuring high positioning accuracy. Assuming the coordinates of stellar coverage in the star chart are $x1 \le u \le x2$ and $y1 \le v \le y2$, the formula for computing the centroid of a star using the centroid method is:

$$\begin{cases} u = \dfrac{\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} x_i f(x_i, y_i)}{\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} f(x_i, y_i)} \\ v = \dfrac{\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} y_i f(x_i, y_i)}{\sum_{x=x_1}^{x_2}\sum_{y=y_1}^{y_2} f(x_i, y_i)} \end{cases} \tag{1}$$

Using the centroid method to extract star points, the coordinates of the star points in the imaging coordinate system of the star sensor can be obtained as (u, v). Assuming that the optical axis of the star sensor is located at coordinates (u_0, v_0) in the plane of the sensor's coordinate system, and the

focal length of the star sensor is f, the star observation vector $\mathbf{W}_s$ in the star sensor coordinate system can be expressed as:

$$\mathbf{W}_s = \frac{1}{\sqrt{(u-u_0)^2+(v-v_0)^2+f^2}} \begin{bmatrix} (u-u_0) \\ (v-v_0) \\ -f \end{bmatrix} \tag{2}$$

Through star pattern recognition technology, it is possible to determine the specific correspondence between the detected stars in the star chart and the stars listed in the star catalog. According to the star catalog, the reference vector of a star in the celestial coordinate system can be denoted as Vc. Based on the aforementioned coordinate system definition, the following relationship can be derived:

$$\mathbf{W}_s = \mathbf{C}_i^s \mathbf{V}_c = \mathbf{C}_b^s \mathbf{C}_n^b \mathbf{C}_i^n \mathbf{V}_c \tag{3}$$

Where $\mathbf{C}_i^s$ represents the rotation matrix from the inertial coordinate system to the star sensor coordinate system. Using equation(4), we can solve for the spacecraft attitude matrix $\mathbf{C}_n^b$ based on the star sensor installation matrix.

$$\mathbf{C}_n^b = (\mathbf{C}_b^s)^{\mathrm{T}} \mathbf{C}_i^s (\mathbf{C}_i^n)^{\mathrm{T}} \tag{4}$$

2.2 Measurement error of star sensor

During the imaging process of a star sensor, the three types of noise that have the greatest impact are shot noise, dark current noise, and readout noise. All three types of noise are classified as random noise. Since random noise follows a probabilistic distribution, the variance of the noise is typically used to measure its absolute magnitude. These three types of noise are independent and uncorrelated, meaning their noise power can be directly added together, hence they are also referred to as additive noise. The signal-to-noise ratio (SNR) is generally used to measure the relative magnitude of various noises. For dynamic star maps, when the light signal is weak, readout noise and dark current noise cannot be neglected. Photon shot noise, however, cannot be reduced or compensated for, which primarily limits the SNR. The SNR of a star map can be approximately expressed as:

$$SNR = \frac{S}{\sigma_n} = \frac{S}{\sqrt{\sigma_{shot}^2 + \sigma_{dark}^2 + \sigma_{read}^2}} \tag{5}$$

Where S represents the number of signal electrons from the star point; σ_{shot}^2 denotes the variance of shot noise, which correlates with the number of incident photons; σ_{dark}^2 represents the variance of dark current noise, which is dependent on the exposure time; and σ_{read}^2 signifies the readout noise, which is related to the intrinsic properties of the star sensor's fabrication process.

3. End-to-end recognition method of fuzzy star map based on grid algorithm

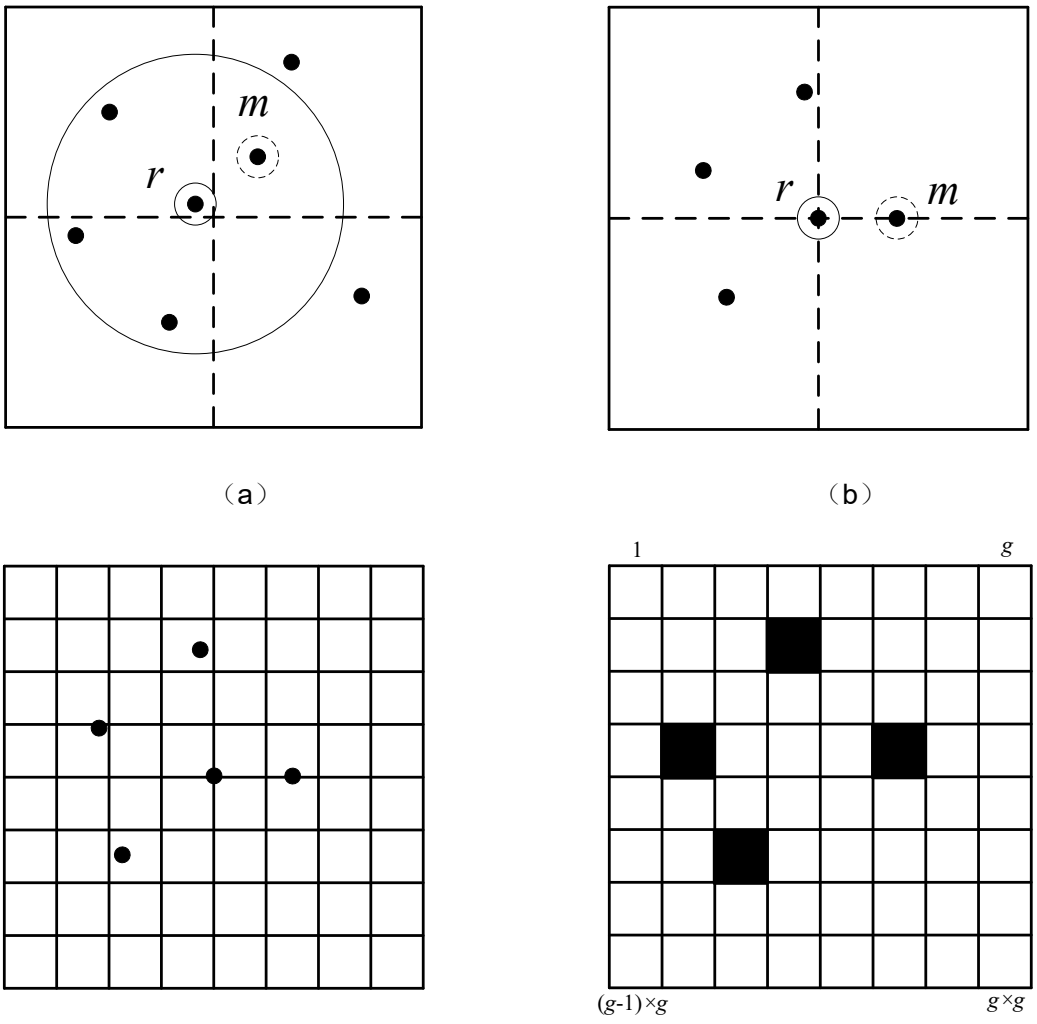
The grid algorithm, a pattern recognition-based star image recognition method, demonstrates good anti-interference capabilities. In this section, we build upon the grid star image recognition algorithm and incorporate convolutional neural networks (CNNs) to further enhance the robustness of the algorithm.

3.1 Limitations of traditional grid recognition algorithms

The grid algorithm is a typical pattern recognition-based star image recognition method, first proposed by Padgett and Kreutz-Delgado in 1997[6]. The fundamental principle of this algorithm involves defining a grid pattern for each star. The specific construction process includes the following steps: First, select the star r to be identified within the star image and reposition it along with other stars within a certain angular distance to the center of the star image. Next, identify the nearest neighboring star m within this range, which is farther from r than a predetermined distance p, as the reference star. Rotate the star image about r so that m points towards the east direction of the star map. Then,

the star map is divided into $g*g$ grids. All stars in the sky area, except for the star to be identified r, are projected onto each grid. If a grid contains a star, it is marked as 1; otherwise, it is marked as 0. Finally, flatten the 0, 1 matrix of the grid into a vector v and store it in the navigation star catalog.

The pattern establishment for the star r to be identified is completed. By repeating the above steps, a feature library for star map matching is established, transforming the star image recognition problem into searching for grid patterns similar to the star to be identified within the feature library. The process of constructing grid patterns in the grid algorithm is illustrated in Figure 1. Experiments have shown that the grid algorithm has good robustness and a high recognition success rate. To some extent, it does not require parameter adjustments to adapt to the noise interference in star sensors.



（a） （b）

（c） （d）

Figure 1 – Grid algorithm schematic diagram

Despite the advantages of the grid algorithm[11], it still exhibits limitations in feature construction and recognition, which are primarily manifested in the following two aspects:

(1) The accuracy of selecting the correct guide star is relatively low. Even in the absence of positional noise in the star points, the probability of correctly identifying neighboring stars is only about 50%. This probability further decreases as positional noise increases. The main reasons for this outcome include: firstly, when the star to be identified is at the edge of the field of view, there is an increased likelihood that the guide star falls outside the field of view; secondly, under the presence of magnitude noise, it is impossible to ensure that the extracted brightness information of the star points has sufficiently high precision. When an error occurs in determining the guide star, an incorrect feature

pattern will be constructed, leading to a failure in accurate identification. The grid algorithm mitigates the impact of erroneous neighboring star determination by increasing the number of stars to be identified, thereby maintaining a high recognition rate.

(2) The feature pattern cannot accurately reflect the intrinsic similarity of the star map. For example, consider a grid with g=8 in Figure 2. Suppose its feature pattern vector is v. According to the construction process of the grid pattern feature vector, the elements (14, 19, 39, 45, 51) of the feature vector v are 1, while the others are 0. Under the influence of the positional noise of star points, a star located at the grid edge may move from position A to position B, resulting in an extracted feature pattern vector v'. Evidently, the elements (19, 22, 39, 45, 51) are 1, while the others are 0. It is apparent that there is a significant discrepancy between the feature vectors extracted using this method, despite similar distribution characteristics, failing to accurately reflect their similarity in the feature space. Consequently, this algorithm cannot meet the robustness requirements under noisy conditions.

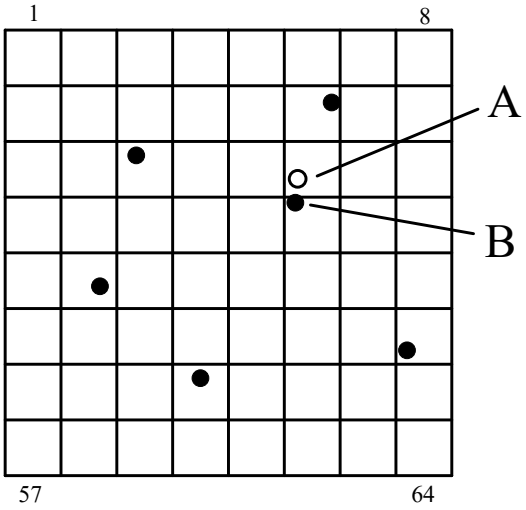


Figure 2 – Grid star map with g=8

For some neural network star map recognition algorithms, the distance vectors or angular distances between reference stars and their neighboring stars are used to identify the reference stars. This means that patterns must be created in advance. Generally, the more patterns there are, the higher the accuracy. Unfortunately, too many patterns can lead to excessive computational complexity during network training. Therefore, these methods may not be suitable for practical tasks. Additionally, these methods make it difficult to achieve good recognition when there are false stars because false stars can alter the distance vectors or angular distances with reference stars.

In order to address the shortcomings of grid algorithms and existing neural network star map recognition algorithms, an improvement approach is proposed. The idea is to combine the simplicity and ease of implementation of grid algorithms with the robustness of convolutional neural networks in handling small deformations, translations, and rotations of images. The model database is transformed into a network structure, reducing memory usage and shortening recognition time. The basic idea of the improved grid recognition algorithm based on convolutional neural networks is as follows: first, preprocess the star map by extracting star points from the star map and constructing recognition patterns for the stars to be identified. Then, divide the entire celestial region into partitions and train separate recognition networks for each partition. Next, predict the celestial region of the star to be identified based on attitude information provided by the inertial navigation system, and input the recognition pattern into the network model of that partition. Finally, the network classifier outputs the star ID of the star to be identified, completing the star map recognition.

3.2 Improved end-to-end recognition algorithm based on convolutional neural network

Compared to other traditional machine learning algorithms, CNN employ a multi-layer structure to enhance the network's generalization and abstraction capabilities[12]. The primary feature of CNNs lies in their ability to autonomously learn complex models by extracting abstract features from the input layer through a series of operations such as convolution filtering, local normalization, nonlinear functions, and downsampling. The network parameters are optimized through a training process involving backpropagation and optimization algorithms like gradient descent. The initial layers of the network typically represent low-level features, while the final layers are capable of integrating these low-level features to recognize and classify more complex target concepts. The overall structure of the Convolutional Neural Network is shown in Figure 3.

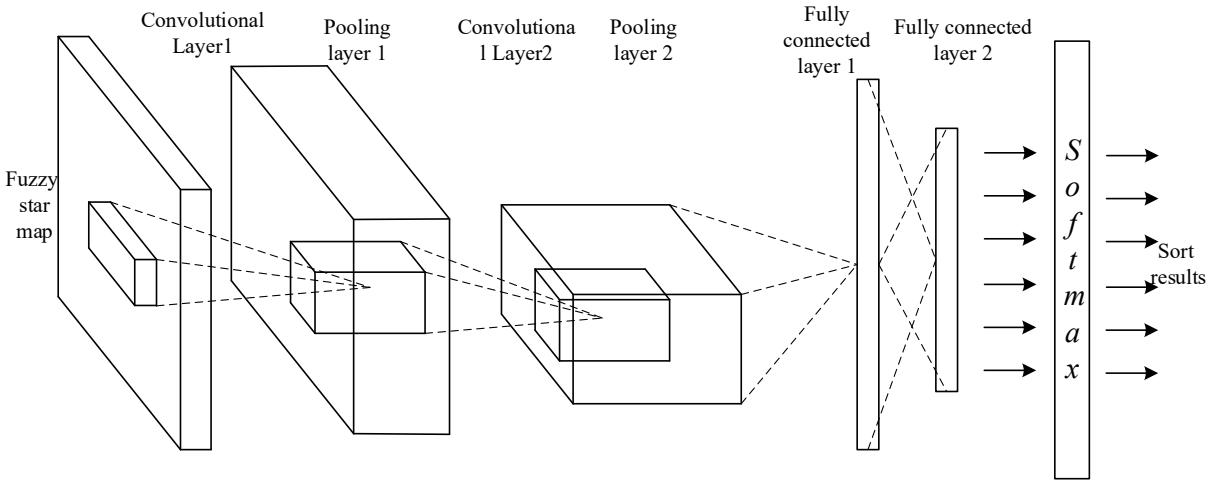


Figure 3 – Convolutional neural network structure diagram

3.2.1 Convolution layer

The convolution operation employs mechanisms of local connections and weight sharing, simulating the behavior of cell units with local receptive fields to extract primary signal features. Local connections refer to each neuron in the convolutional layer being connected to neurons within a certain region of the previous layer's features. Weight sharing means that neurons in the same feature use the same connection weights to establish relationships with local connections in the previous layer, thereby effectively reducing the number of network training parameters. Each neuron in the convolutional layer receives input from a specific region of neurons in the previous layer's feature map, with the size of this region determined by the size of the convolution kernel. The convolutional layer convolves the input data with m learnable convolution kernels, adds biases, and then processes it through an activation function to obtain m feature maps. The next layer of the convolutional structure consists of n feature maps, which are obtained by convolving each of the m feature maps from the previous layer with $n*m$ convolutional kernels. After combining the results of every m convolutions, adding biases, and processing through an activation function, the output is achieved. The mathematical expression for the convolutional layer is as follows.

$$\mathbf{y}_{m,n} = f\left(\sum_{j=0}^{J-1}\sum_{i=0}^{I-1}\mathbf{x}_{m+i,n+j}\mathbf{w}_{ij} + \mathbf{b}\right) \tag{6}$$

Where x is the input two-dimensional data; y is the output of size $M*N$; where $0 \le m < M$, $0 \le m < M$; w represents a convolution kernel of size $J*I$; b stands for bias; f stands for activation function.

3.2.2 Downsampling processing

The dimensionality of the feature vectors generated after convolution is quite large. If these features are directly used for classification and recognition, it will result in massive computational load and high complexity. Therefore, before training a classifier on the extracted features, dimensionality

reduction is necessary. To simulate the dimensionality reduction process of complex cellular units, downsampling operations are employed to filter and combine primary features into higher-level and more abstract features. In the network, after sampling through the downsampling layer, the number of output feature maps remains unchanged, but the computational load of the network is significantly reduced. Additionally, the network's invariance to object translation and scaling is enhanced, improving the network's robustness. The mathematical expression for the downsampling operation is shown in Equation(7), where common downsampling operations include max pooling and average pooling.

$$\mathbf{y}_{m,n} = \frac{1}{S_1 S_2} \sum_{j=0}^{S_2-1} \sum_{i=0}^{S_1-1} \mathbf{x}_{m\times S_1+i, n\times S_2+j} \tag{7}$$

Where x is the input vector after convolution processing, y is the output after downsampling, S_1 and S_2 are the downsampling scales.

3.2.3 Fully connected layer

To enhance the nonlinear mapping capability of the network while limiting its size, a fully connected layer is introduced after the feature extraction layer. Unlike the convolutional layer, the fully connected layer can perceive global information, integrating the local features extracted by the convolutional layer to form global features for specific target tasks. In this layer, each neuron is connected to all neurons in the previous layer, while neurons within the same layer are not connected. The mathematical expression for the fully connected layer is as follows:

$$\mathbf{o}_j^{(l)} = f\left(\sum_{i=1}^{n} \mathbf{x}_i^{(l-1)} \times \mathbf{w}_{ji}^{(l)} + \mathbf{b}^{(l)} \right) \tag{8}$$

Where n is the number of neurons in the previous layer, l is the number of current layers, w_{ji} is the connection weight of neuron j in this layer and neuron i in the previous layer, $b^{(l)}$ is the bias of neuron j in this layer, and f represents the activation function.

3.2.4 Activation function

When using the gradient descent method to update the weights of nodes in a deep neural network, the gradient change value for each layer needs to be multiplied by $\partial y_p^{l+1} / \partial y_p^l$, where y_p^l represents the value of the p-th node in the l-th layer. If the values of $\partial y_p^{l+1} / \partial y_p^l$ are all less than 1, the gradient will gradually decrease as the recursion continues, leading to the vanishing gradient problem. This results in slower updates of the weights W and biases b in the hidden layers, making it difficult to achieve convergence.

The ReLU activation function is commonly used in convolutional neural networks. The expression for the ReLU function is:

$$f(x) = \begin{cases} x & x \geq 0 \\ 0 & x < 0 \end{cases} \tag{9}$$

From Equation(9), we know that the ReLU function remains unchanged when $x \geq 0$; it is zero, when $x < 0$. The derivative of this function is simple: when $x \geq 0$, the derivative is 1, and when $x < 0$, the derivative is 0. Therefore, it does not suffer from gradient vanishing issues during backpropagation, helping to mitigate the problem of vanishing gradients.

3.3 Network parameter setting

Firstly, the selection of the pattern radius R for the raster image is determined to statistically analyze the number of neighboring stars within different angular distances for stars with a magnitude less than 6.0 in the entire sky. The statistical results are shown in Figure 4. From the statistical results, it

can be seen that when $R \geq 8^{\circ}$, all stars in the sky have 8 or more neighboring stars, whereas when R=6、7°, some stars have only 1-2 neighboring stars. Since this algorithm belongs to the category of pattern recognition algorithms, it requires a sufficient number of neighboring stars as recognition information. At the same time, if the pattern radius is too large, it results in an insufficient number of navigational stars being recognizable within a single star map. Therefore, comprehensively considering these factors, a pattern radius of R=8° is chosen.

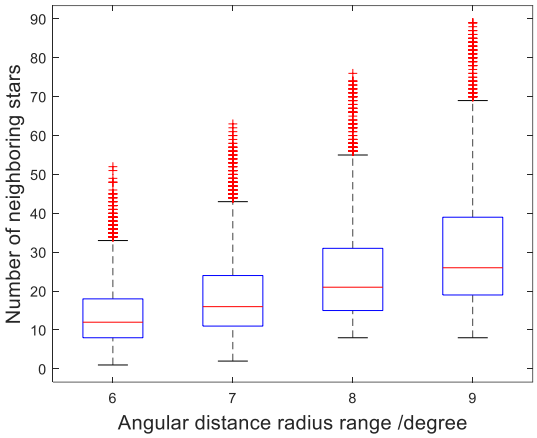


Figure 4 – Distribution of neighboring stars within different angular distance ranges

Secondly, based on the determined pattern radius, the size of the star map to construct the pattern and the size of the maximum pooling layer window L in the preprocessing are calculated. Calculations show that under the star sensor, an angular distance of 8° can span up to 667 pixel blocks. To avoid the impacts of estimation errors, a star map size of 1430×1430 pixels is redundantly selected for rasterization processing, and a window size of 22 with a stride of 22 is chosen for the maximum pooling layer to downsample the star map within the pixel. Finally, a 65×65 raster image size is obtained as the network output. Each raster spans approximately 0.25° of angular distance, allowing for the distinction of each neighboring star.

To prevent overfitting, the network selected in this paper consists of only two convolutional layers and two fully connected layers. The two convolutional layers contain 32 and 16 square convolution kernels respectively. A max-pooling layer with a window size of 2×2 and a stride of 2 is connected after each convolutional layer, followed by batch normalization and the selection of the ReLU function as the non-linear activation. The final two layers are fully connected layers, and the number of neurons in each layer is adjusted according to the number of navigational stars in the region. The number of neurons in the first fully connected layer is twice the number of navigational stars, while the second fully connected layer has the same number of neurons as the number of navigational stars. Dropout is added in the first fully connected layer to reduce overfitting, and the final layer performs softmax classification. The training set and test set are divided from the training samples at an 8:2 ratio. Other training parameters are set as shown in Table 1.

Table 1: Training Parameter Settings

Parameter	Value
Num epochs	250
Learning rate	0.001
L2 weight decay	0.99
Learn rate drop factor	0.4
Drop step size	50
Batchsize	256

4. Simulation verification

4.1 Simulation of the nominal trajectory of aerospace vehicle

This study focuses on the hypersonic glide vehicle provided by NASA, known as the "High Lift-to-Drag Ratio Hypersonic Glide Vehicle" (abbreviated as CAV-H). The CAV-H hypersonic glide vehicle serves as the prototype for the HTV-2 hypersonic glide vehicle[13], jointly developed by the United States Air Force and Lockheed Martin. The aerodynamic shape and specific data of this vehicle are shown in Figure 5.

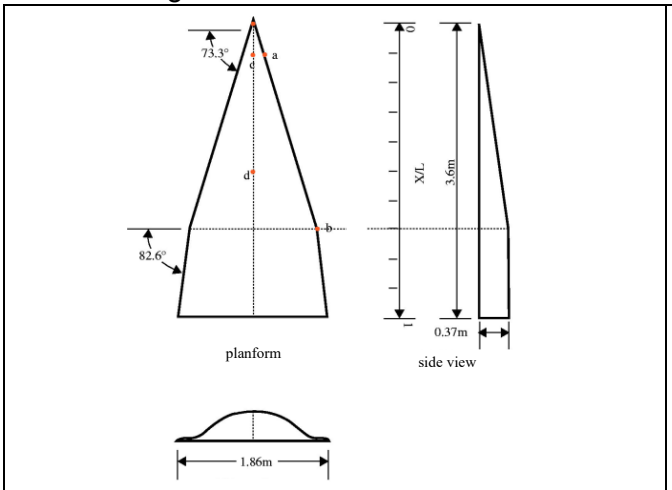


Figure 5 – CAV-H aerodynamic configuration and data

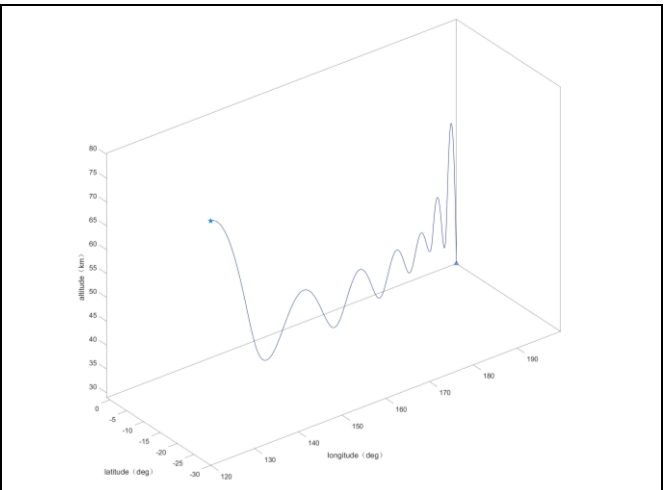


Figure 6 – Nominal trajectory simulation

The general parameters of the aerospace vehicle are shown in the table below:

Table 2 – CAV-H General parameters

General parameters	Value
Mass（kg）	907.0
Reference area（m2）	0.4836
Reference length（m）	3.6

Table 3 –Initial parameters

Initial parameters	Value
Longitude of the launching point（deg）	115.0
Latitude of transmitting point（deg）	-30
Launch height（km）	1.5
Point of launch (deg)	90

4.2 Training set sample generation

4.2.1 Input identification mode construction

Due to the rotational invariance property of convolutional neural networks in extracting image features, it is unnecessary to determine the rotation direction in advance. Instead, the star map to be recognized is directly downsampled using a circular range with the star as the center and a radius R. As shown in Figure 6, the process involves zero-padding the pixels outside this range, followed by max-pooling with a window size of $L*L$ and a stride of L. Subsequently, the compressed image undergoes binarization: grid points with a grayscale value greater than the threshold T are assigned as 1, while those less than T are assigned as 0. Finally, the resulting binary image serves as the

recognition pattern for network input.

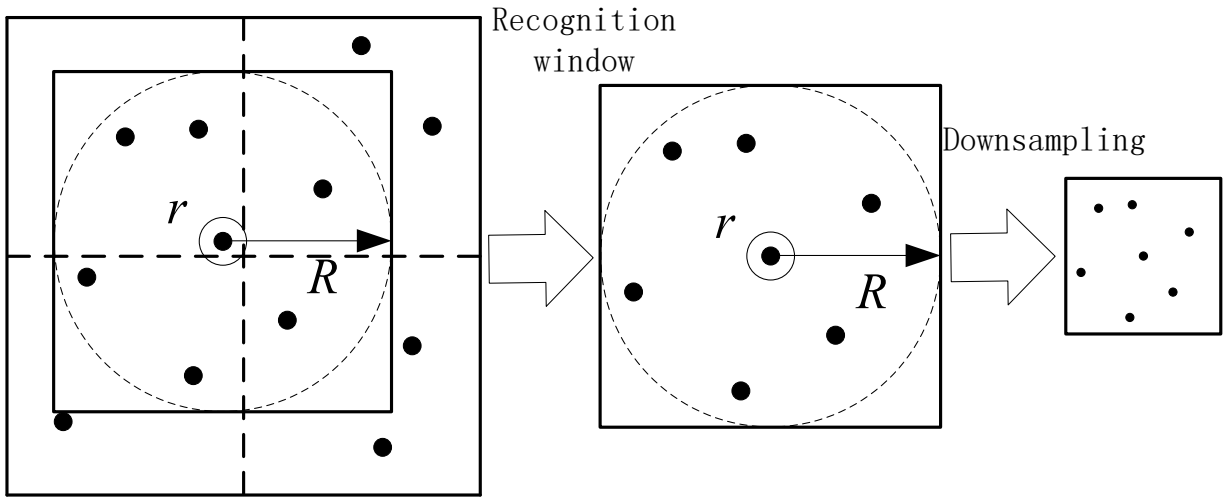


Figure 7 – Improved raster algorithm for pattern recognition

To improve the robustness of the recognition network in different noise conditions, it is necessary to train with an ample amount of samples. The samples are divided into two categories: noise-free and noisy. The noise-free samples are constructed based on simulated star maps under ideal conditions, following the recognition patterns mentioned earlier. On the other hand, noisy samples require the introduction of varying degrees of noise into the ideal star map. The type and quantity of original noisy samples directly affect the network's resistance to interference. The more noisy samples in the training set, the higher the accuracy of recognition. However, due to hardware constraints and efficiency, there are limitations on the types and quantities of noise. The types and methods of adding noise are shown in Figure 7.

(1) Position noise

The position noise results in the displacement of the stellar coordinates in the ideal star chart. Gaussian noise with mean 0 and standard deviations σ_x and σ_y, is added to the x and y coordinates of all stars in the ideal star chart except for the primary star to be identified. In order to adhere to the principle of symmetry, the setting σ_x is equal to σ_y. The standard deviations range from 0 pixels to 2 pixels in increments of 0.4 pixels, yielding six different categories of positional noise of varying magnitudes.

(2) Magnitude noise

For stars with magnitudes near the detection limit of the sensor, magnitude noise may cause them to dim and disappear. Conversely, for stars previously undetectable, magnitude noise may brighten them into visibility. Thus, a detectable magnitude threshold of 6.0Mv is set, and the ideal star chart is simulated according to this threshold. Magnitude noise is then added to each star in the ideal star chart except for the primary star to be identified, and stars dimmer than 6.0Mv are filtered out. Finally, noise samples can be constructed based on the previously generated noisy images. The magnitude noise follows a Gaussian distribution with the parameter a ranging from 0Mv to 1.0Mv in increments of 0.2Mv, resulting in six different categories of magnitude noise.

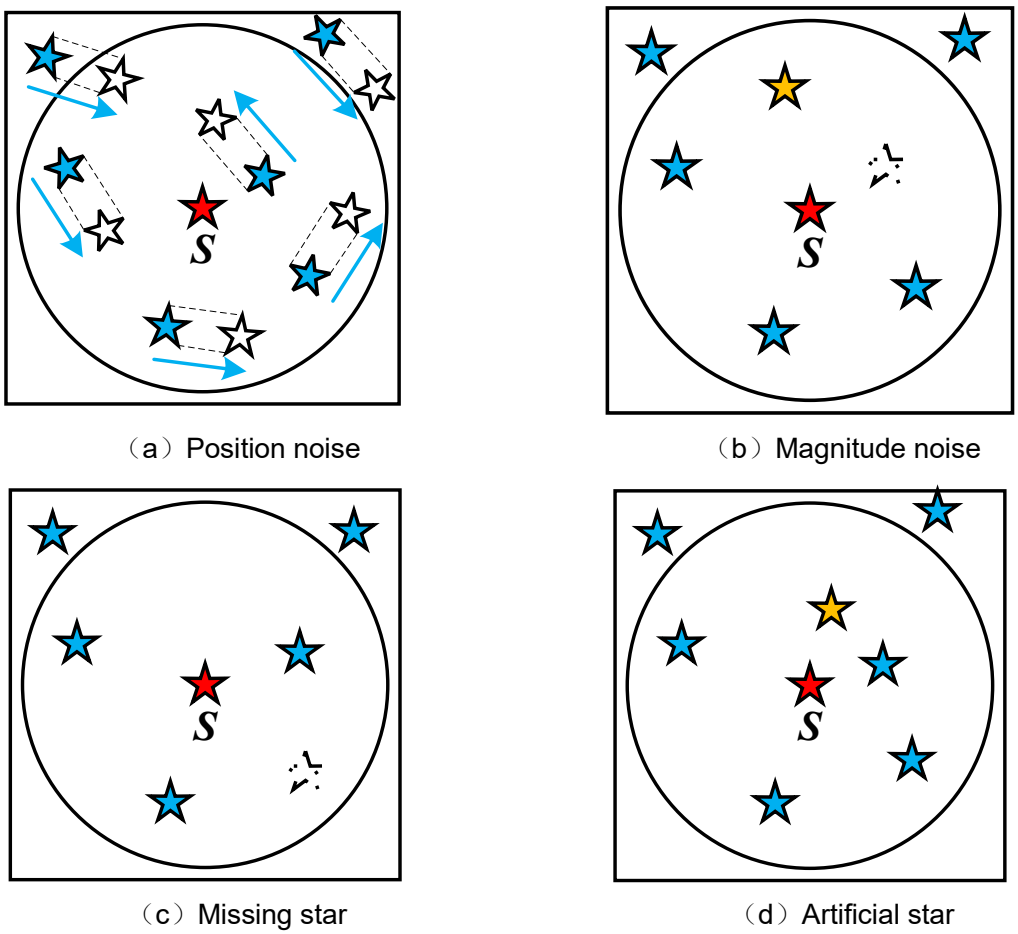
(3) Missing star

Some stars that should appear may randomly disappear due to circuit noise or imaging noise in the optical system. For the ideal star chart, excluding the primary star to be identified, 0 to 5 stars are randomly discarded to generate samples with missing stars. This results in six different categories of star-missing samples with varying degrees of star loss.

(4) Artificial star

Artificial stars are common in real star charts; planets, cosmic dust, background white noise, and even flames from an airplane's tail can be mistaken for false stars. The appearance of false stars is random and therefore considered to be uniformly distributed across the image plane. In a circular area with a radius R around the star to be identified, randomly add 0 to 5 false stars with magnitudes ranging from 0Mv to 6Mv. This results in six different categories of false star samples with varying degrees of false star occurrence.

Based on the analysis above, there are a total of 1,296 categories of noise samples. For each category, 5 star maps are generated, resulting in 6,480 training samples for each star recognition pattern. Additionally, during the training process, each sample is randomly rotated around its center at various angles.



（a）Position noise （b）Magnitude noise

（c）Missing star （d）Artificial star

Figure 8 – Noise sample diagram

4.3 End-to-end identification algorithm performance comparison

To evaluate the performance of the improved grid star map recognition algorithm based on convolutional neural networks presented in this section, we tested the robustness of this algorithm under four different noise conditions. Additionally, given that this algorithm falls under the category of pattern recognition algorithms, we used two other typical pattern recognition star identification algorithms as references: the first is the traditional grid algorithm; and the second is an improved version of the traditional grid algorithm based on radial-angular features[14]. Star maps were generated for recognition testing by randomly selecting the star sensor's optical axis direction and the rotation angle around the optical axis, with the default assumption that the celestial region pointed to by the optical axis is known.

First, the robustness of the star map recognition algorithm under centroid position error conditions is tested. Eight levels of positional noise are set, with a mean of 0 and standard deviations ranging from

0.2 to 1.6 pixels. For each noise level, 10,000 simulated star maps are randomly generated as test data. The impact of positional noise on the recognition rates of different algorithms is shown in Figure 9. The results show that the recognition rate of this algorithm consistently remains above 96%, and it stays above 99% without any degradation when the positional noise is less than 1 pixel. This indicates that the algorithm presented in this paper possesses considerable robustness against slight deformations caused by positional noise. In comparison with the other two algorithms, as positional noise increases, the recognition rate of the proposed algorithm always surpasses that of the others. The recognition rate of the traditional grid algorithm experiences a steep decline. When the positional noise increases to 1.6 pixels, the recognition rate of the traditional grid algorithm drops to 85.64%, while the recognition rate of the algorithm based on radial-angular features decreases to 94.22%. In contrast, the proposed algorithm maintains a recognition rate of 96.40%, showing only a slight decline.

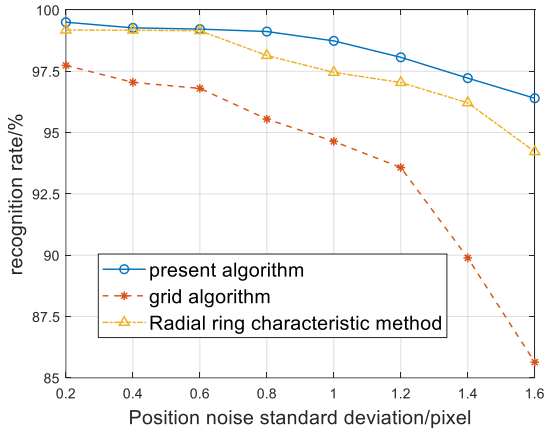


Figure 9 – The effect of position noise on recognition rate

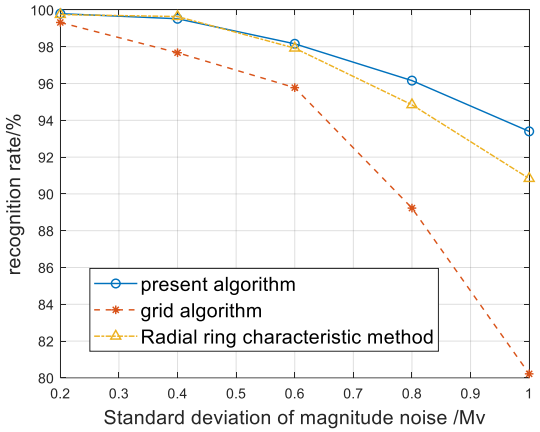


Figure 10 – The effect of magnitude noise on recognition rate

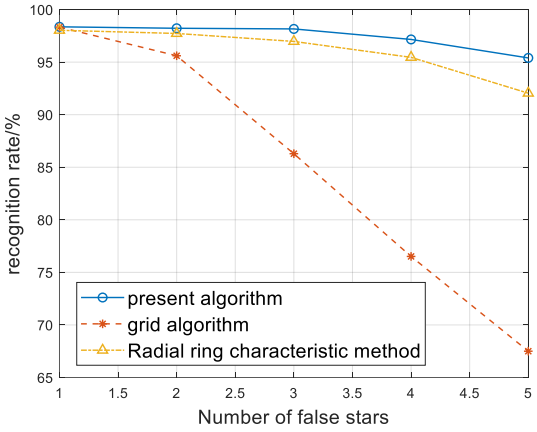


Figure 11 – The effect of the number of false stars on the recognition rate

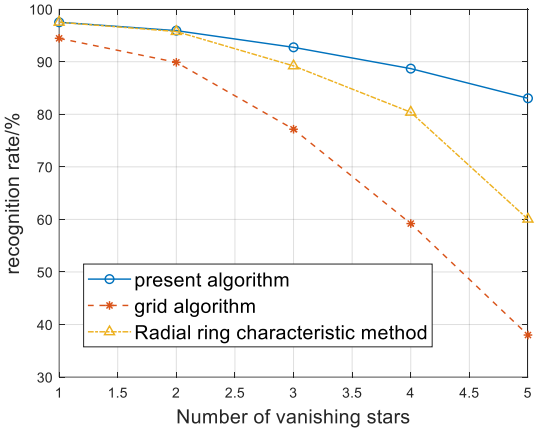


Figure 12 – The effect of the number of missing stars on the recognition rate

Then, the robustness of the star map recognition algorithm under magnitude noise is tested. Five levels of magnitude noise are set with a mean of 0 and standard deviations ranging from 0.2 to 1.0 magnitudes. For each level, 10,000 simulated star maps are randomly generated as test data. The impact of magnitude noise on the recognition rates of different algorithms is shown in Figure 10. The results show that all three algorithms can maintain a recognition rate above 97% when the magnitude noise is below 0.4 magnitudes. This is because grid images are constructed independently of brightness information. However, under high-magnitude noise, the recognition rate of the traditional grid algorithm drops to 80%, and the recognition rate of the algorithm based on radial-angular features also falls to 90%. This decline is due to some low-brightness stars within the field of view disappearing or false stars appearing under magnitude noise, which affects the establishment of

recognition patterns. Although the proposed algorithm is slightly affected overall, its recognition rate remains higher than the other two algorithms.

Next, the robustness of the star map recognition algorithm under the influence of different numbers of false stars is tested. The number of false stars increases from 0 to 5, and their positions and magnitudes are randomly added to the simulation. Simultaneously, 10,000 simulated star maps are randomly generated as test data. The impact of the number of false stars on the recognition rates of different algorithms is shown in Figure 11. The results indicate that the algorithm presented in this paper demonstrates strong robustness under the influence of false stars, maintaining an accuracy rate above 95%. When the number of false stars increases from 1 to 5, the recognition accuracy of the radial-angular-based algorithm slightly decreases from 98.03% to 92.24%, whereas the recognition accuracy of the traditional grid algorithm drops from 98.37% to 67.59%. As mentioned previously, the key to generating grid images is identifying the correct reference stars, and the presence of false stars leads to incorrect selections.

Lastly, we tested the robustness of the star map recognition algorithm under the influence of different numbers of missing stars. Within the radius range of the recognition pattern, we randomly reduced the number of stars by 1 to 5, while simultaneously generating 10,000 simulated star maps as test data. The impact of the number of missing stars on the recognition rates of different algorithms is shown in Figure 12. The results indicate that the algorithm presented in this paper performs better under the influence of missing stars compared to traditional algorithms. When the number of missing stars increases from 1 to 5, the recognition accuracy of the radial-angular-based algorithm decreases from 97.541% to 60.04%, while the recognition accuracy of the traditional grid algorithm drops from 94.45% to 38.23%. In contrast, the accuracy rate of the proposed algorithm remains above 80%, significantly higher than the other two algorithms.

5. Conclusion

The paper proposes an end-to-end method for recognizing blurred star patterns. The algorithm simplifies the traditional recognition process of first determining the centroid position and then matching the star pattern. In real flight environments, noise in star points can seriously reduce the signal-to-noise ratio of star patterns. In this paper, the CNN algorithm is used to quickly detect star patterns. By using the above solution, the attitude accuracy of the starlight navigation system can be improved. The simulation example given in the paper demonstrates the effectiveness of this method.

6. Contact Author Email Address

Corresponding author: xuzhi@nwpu.edu.cn

7. Copyright Statement

References

[1] Abosekeen A, Iqbal U, Noureldin A. A novel multi-level integrated navigation system for challenging GNSS environments. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 22, No.8, pp 4838-4852, 2020.

[2] Na M, Jia P. A survey of all-sky autonomous star identification algorithms. *1st International Symposium on Systems and Control in Aerospace and Astronautics*. China, Vol. 6, pp 901,2006.

[3] Rijlaarsdam D, Yous H, Byrne J. A survey of lost-in-space star identification algorithms since 2009. *Sensors*, Vol. 20, No. 9, pp 2579,2020.

[4] Liebe C C. Pattern Recognition of Star Constellations for Spacecraft Applications. *IEEE Aerospace and Electronic Systems Magazine*, Vol. 8, No. 1, pp 31-39, 1993.

[5] Quine B M, Durrant-Whyte H F. A fast autonomous star-acquisition algorithm for spacecraft. *Control Engineering Practice*, Vol. 4, No. 12, pp 1735-1740, 1996.
[6] Padgett C, Kreutz-Delgado K. A grid algorithm for autonomous star identification. *IEEE Transactions on Aerospace & Electronic Systems*, Vol. 33, No. 1, pp 202-213, 1997.
[7] Du J, Wei X, Li J. Star identification based on radial triangle mapping Matrix. *IEEE Sensors Journal*, Vol. 22, No. 9, pp 8795-8807, 2022.
[8] Rijlaarsdam D, Yous H, Byrne J. Efficient star identification using a neural network. *Sensors*, Vol. 20, No. 13, pp 3684, 2020
[9] Jiang J, Liu L, Zhang G. Star identification based on spider-web image and hierarchical CNN. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 4, pp 3055-3062, 2019.
[10] Yang S, Liu L, Zhou J. Robust and Efficient Star Identification Algorithm based on 1-D Convolutional Neural Network. *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 58, No. 5, pp 4156-4167, 2022.
[11] Shortis M R. Comparison of some techniques for the subpixel location of discrete target images. *Proceedings of the Spie Videometrics III* Boston Usa, Vol. 1, pp 1-9, 1994.
[12] Gu J, Wang Z, Kuen J. Recent advances in convolutional neural networks. *Pattern recognition*, Vol. 77, pp 354-377, 2018.
[13] SPEYER J L, DANNEMILLER D, WALKER D. Periodic optical cruise of an atmospheric vehicle. *Journal of Guidance, Control, and Dynamics*, Vol. 8, No. 1, pp 31-38, 1985.
[14] Zhang G, Wei X, Jiang J. Full-sky autonomous star identification based on radial and cyclic features of star pattern. *Image and Vision Computing*, Vol. 26, No. 7, pp 891-897, 2008.