

Article

A Fast Star-Detection Algorithm under Stray-Light Interference

Kaili Lu ^{1,2,*}, Huakang Li ^{1,2}, Ling Lin ^{1,2}, Renjie Zhao ^{1,2}, Enhai Liu ^{1,2} and Rujin Zhao ^{1,2}

¹ Key Laboratory of Science and Technology on Space Optoelectronic Precision Measurement, Chinese Academy of Sciences, Chengdu 610209, China; lihuakang@ioe.ac.cn (H.L.); hyj@ioe.ac.cn (L.L.); zhaorenjie@ioe.ac.cn (R.Z.); leh@ioe.ac.cn (E.L.); zhaorj@ioe.ac.cn (R.Z.)

² Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China

* Correspondence: lukaili@ioe.ac.cn

Abstract: The interference caused by stray light leads to the invalid attitude of star sensors in orbit, thus affecting the attitude control of satellites. In order to overcome this problem, this paper proposes a fast star-detection algorithm with strong stray-light suppression ability. The first step in the proposed method is stray-light suppression. The highlighted pixels are unified and then erosion and dilation operations based on a large template are performed. Using the background image only, which is filled with stray light, the cleaner star image is obtained by subtracting the background from the unified image. The second step in the proposed method is binarization. The binary star image is obtained by using a line-segment strategy combined with a local threshold. The third step in the proposed method is star labeling. It comprises connected-domain labeling based on the preordering of pixels and the calculation of centroid coordinates of stars in each connected domain. The experimental results show that the proposed algorithm extracts the stars stably under the interference of different stray lights. The proposed method consumes less resources, and the output delay is only 18.256 us. Moreover, the successful identification rate is 98% and the attitude accuracy of the X and Y axes is better than 5"(3 σ) when the star sensor works at the speed of zero.

Keywords: stray light; star detection; highlighted pixels; image subtraction; star labeling; preorder pixels



Citation: Lu, K.; Li, H.; Lin, L.; Zhao, R.; Liu, E.; Zhao, R. A Fast Star-Detection Algorithm under Stray-Light Interference. *Photonics* **2023**, *10*, 889. <https://doi.org/10.3390/photonics10080889>

Received: 4 July 2023

Revised: 26 July 2023

Accepted: 27 July 2023

Published: 1 August 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A star sensor [1] is a high-precision instrument for measuring the attitude of satellites. The star sensors are installed outside the satellites and usually operate in the presence of interference caused by stray light [2], such as sunlight, moonlight, earth-atmosphere light [3], etc. Due to the interference caused by stray light, the attitude data acquired using star sensors are easily corrupted. As a result, these data are unable to provide the attitude information of satellites accurately. Traditional star sensors rely on the lens hood [4] for performing the shading function. In order to achieve a suitable shading effect, a common technique is to increase the number of vanes inside the lens hood and the length of the lens hood. However, it is noteworthy that an increase in the volume and weight of the star sensor is an obvious disadvantage. Recently, with the advancements in China's satellite network mission, multi-satellite launching technology has become commonplace. Therefore, the modern development trends of star sensors mainly focus on miniaturization [5] and low power consumption. Please note that the smaller size of the star sensor indicates the smaller size of the lens hood, thus causing a significant reduction in the shading effect. Therefore, it is particularly important to propose a new method of star detection [6,7], which can perform stably under the interference of stray light in real-time.

The traditional star-detection algorithms include threshold segmentation [8], window filtering [9–11], etc. These methods have a better star-detection ability for star images with clean backgrounds. However, under the interference of stray light present in the field of view, the performance of traditional algorithms degrades significantly. As a result, the

research community has presented various methods for addressing this issue. Yu et al. [12] designed a new 7×7 filtering template, which improved the background estimation method and designed a new full-frame background filtering method. However, the limited template is unable to adapt to the different quantities of stray light. Wang et al. [13] studied the local adaptive threshold method, which had a high level of star detection under some complex backgrounds. However, due to high computational complexity, the local threshold method is often unable to perform in real-time. Inspired by the biological vision mechanism, Wei et al. [14] used multi-scale segmentation for realizing multiscale patch-based contrast measure (MPCM). This method adjusts the contrast between the target and the background and detects the bright and dark targets based on threshold segmentation. However, this method has poor suppression ability for a relatively bright thick clutter. Lu [15] proposed a first-order curvature-based method (MDWCM) for small-target detection in the presence of complex background interference. The performance of this algorithm was limited due to its computational complexity and long delay.

In order to deal with the stray-light interference in different scenarios, this paper proposes a new star-detection algorithm with strong stray-light suppression ability. The proposed method has good engineering applicability and real-time performance.

The main innovations of this work are summarized below.

First, the highlighted pixels are unified and then horizontal erosion and dilation operations using a large template are performed. Using this method, a background comprising all the stray light is obtained easily. Then, an enhanced and clean star image is obtained by image subtraction. Second, the star labeling implements the connected-domain labeling based on the preordered pixels. The labeling method is divided into different conditions if the distances between stars may be too small, or if the shape of stars may be heterotypic. The proposed algorithm is verified by performing experiments. The experimental results show that the proposed algorithm effectively detects accurate stars in the presence of interference caused by different stray lights. Moreover, the proposed algorithm has low computational complexity and real-time performance in different platforms. The field experiment shows that the proposed algorithm is beneficial for improving the successful identification rate and guaranteeing the attitude accuracy of the star sensor at different speeds.

The rest of this paper is organized as follows.

Section 2 presents the proposed algorithm. Section 3 presents the experimental results. Finally, Sections 4 and 5 conclude this paper and present directions for future work.

2. Materials and Methods

As presented in Figure 1, the proposed algorithm is divided into three steps: stray-light suppression, binarization and star labeling.

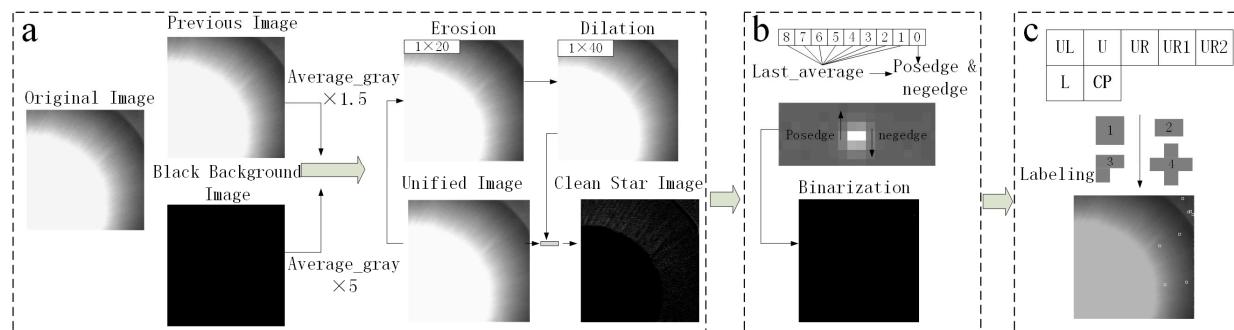


Figure 1. The flowchart of the proposed algorithm: (a) stray-light suppression; (b) binarization; (c) star labeling.

The star image f can be represented as an accumulation of background f_B , stray light f_C , and stars f_s as:

$$f = f_B + f_C + f_s, \quad (1)$$

where the background f_B is closely related to the detector parameter *adc_gain* and *digital_gain*, which are usually determined during the calibration process of the dark field. By adjusting the exposure time and gain parameters, the star target in the star image has a high signal-to-noise ratio, and the star is not saturated. It is noteworthy that the stray light f_C generally represents divergent characteristics in star images, including continuous light spots around the stars. The grayscale value of the light spots is high when the pixel is close to the stray-light center. As presented in Figure 2, under the interference of stray light, the contrast ratio between stars and the background decreases significantly, and low-magnitude stars are overwhelmed by the light spots. This results in a serious decline in star-detection results. The three-dimensional distribution of grayscale values presented in Figure 2 shows that the grayscale value of stray light is indefinite. Therefore, this work proposes a new star-detection algorithm with a strong ability to suppress stray light. The proposed algorithm has real-time performance and strong engineering applicability.

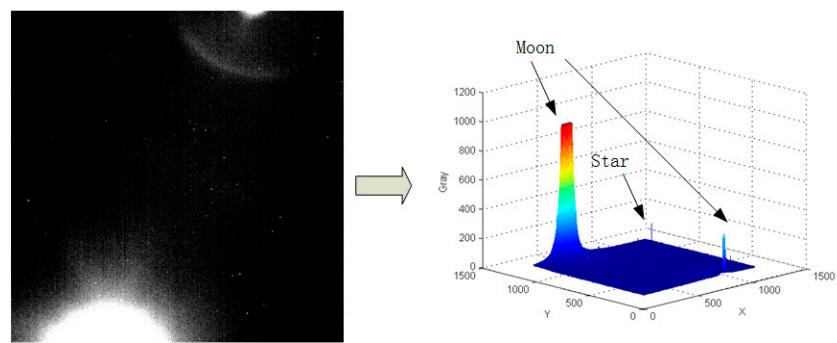


Figure 2. The stray light and the three-dimensional distribution of grayscale values.

2.1. Stray-Light Suppression

The purpose of this step is to eliminate the stray light f_C from the star image f and improve the signal-to-noise ratio of stars. We use the methods of background prediction and background subtraction for eliminating the stray light. The suitable filter template W_B is designed to realize the background prediction.

Unlike normal backgrounds, we consider the impact of stray light on the background prediction. It is noteworthy that due to the inconsistency in the size and distribution of stray light, the direct template filter is useless. Therefore, this work proposes an innovative technique to address this issue, i.e., combining the highlighted pixels and unifying with the horizontal erosion and dilation operations.

In this step, first, the highlighted pixels are unified. The pixels with gray values greater than the highlighted threshold M are set to M uniformly. The aim is to fuse the inhomogeneous boundary and inner of the stray light together for eliminating the local gradient characteristics of stray light, thus improving the background segmentation.

$$f'_i = \begin{cases} M & f_i > M \\ f_i & \text{else} \end{cases}, \quad (2)$$

where M denotes the highlighted threshold, f_i denotes the original image, and f'_i denotes the unified image. Based on prior knowledge, the average gray value $\text{Ave}(f_{i-1})$ of the previous star image f_{i-1} and the average gray value $\text{Ave}(f_B)$ of the black background image f_B from the dark field are considered comprehensively. The relative minimum value is selected as the highlighted threshold M .

$$M = \min(1.5 * \text{Ave}(f_{i-1}), 5 * \text{Ave}(f_B)). \quad (3)$$

As presented in Figure 3, after the unification of the highlighted pixels, the grayscale consistency between the boundary and inner region of stray light becomes better.

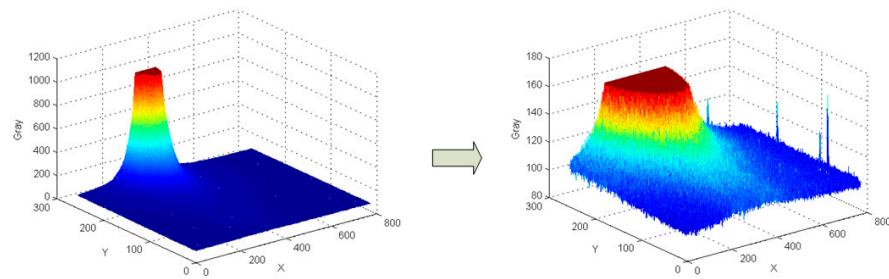


Figure 3. The three-dimensional distribution of stray light before and after the unification of the highlighted pixels.

Second, the filter template W_B is used for background segmentation. Considering that the window sizes of stars range from 3×3 to 9×9 [16], the small template $SE1$ for image convolution is used for the erosion operation. It can eliminate the star object and discrete single-pixel noise. We select the horizontal 1×20 template as it reduces the data cache during the implementation. Afterwards, the dilation operation is performed by using the large template $SE2$. This recovers the scale information of stray light in the star image as much as possible. In order to guarantee the sufficiency of background suppression, the size of the dilation template can be expanded appropriately. The horizontal 1×40 template is selected as $SE2$. The background f_B is obtained after erosion and dilation operations.

$$f_B = (f'_i \ominus SE1) \oplus SE2. \quad (4)$$

Third, we obtain the clean star image f_c by subtracting the background f_B from the unified image f'_i . This is mathematically expressed as follows:

$$f_c = f'_i - f_B. \quad (5)$$

As presented in Figure 4, the stray-light suppression is realized through background subtraction. The resulting star points have a higher signal-to-noise ratio in the star image.

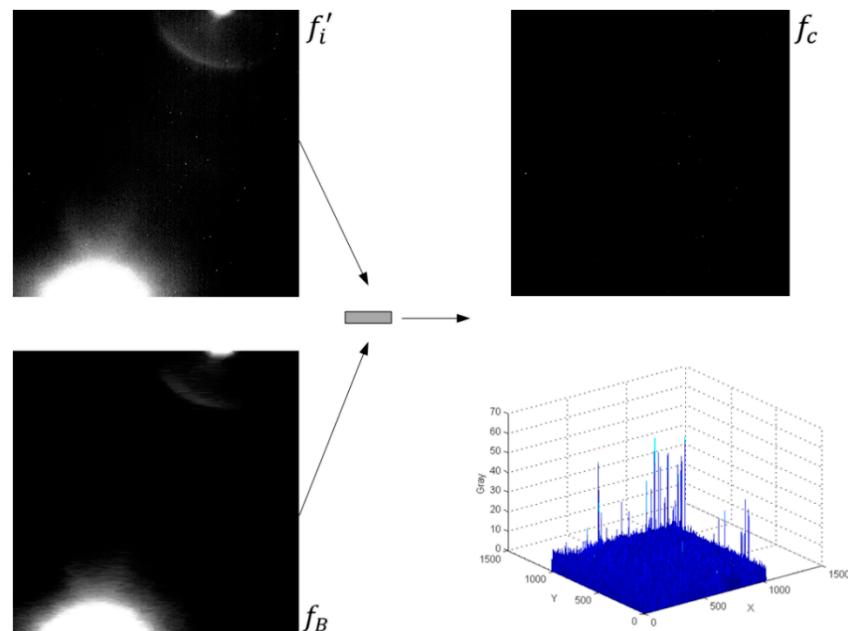


Figure 4. The subtraction of the unified image f'_i and the background f_B , and the three-dimensional distribution of the clean star image f_c .

2.2. Binarization

After the suppression of stray light, the star image with a clean background is obtained. In order to segment the stars from the background, threshold segmentation is required. The traditional global threshold cannot adapt to the stars of different magnitudes. Additionally, the computational complexity of the traditional block threshold is high as well. Therefore, this work proposes the idea of a horizontal local threshold. Please note that the local threshold is calculated while the star image is sliding. The horizontal local threshold is also conducive to reducing the storage consumption in engineering applications.

As shown in the Figure 5, the pixels are scanned from left to right. We compare the current pixel I_c with the mean value I_a of the left eight pixels. If $I_c > I_a + G_{th}$ is satisfied, then the current pixel I_c is considered as the grayscale step point, which initially meets the standard of the star edge. The output binary value I_b at the corresponding position is set to 1, and the length L_i of star line is also set to 1. Then, the gray threshold T_h of the subsequent pixel is locked as $I_a + G_{th}$, and the comparison with the next pixel is performed. If the subsequent pixel I_c satisfies $I_c > T_h$, the output binary value I_b at the corresponding position is set to 1, and the length of the star line segment is incremented by 1. If the subsequent pixel I_c satisfies $I_c \leq T_h$, the output binary value I_b at the corresponding position is set to 0. At the same time, it is necessary to estimate whether the length L_i is greater than 1. If this is not satisfied, it is considered that the previous grayscale step point is only a single-pixel noise. Please note that it is necessary to re-assign the binary value I_{b-1} of the previous pixel to 0. When the search for an old star ends and the search for a new star begins, the current pixel I_c should be compared with the mean value I_a of the left eight pixels. The gray threshold T_h cannot be locked until the edge of the star is obtained.

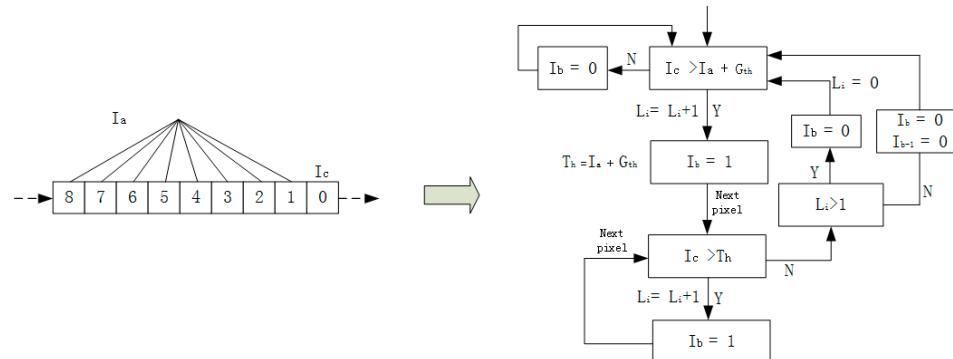


Figure 5. The process of star-image binarization.

Therefore, the lines with lengths greater than 1 are set to 1, while other pixels are set to 0.

2.3. Labeling

After binarization, it is necessary to mark the connected domain of the star points. The traditional methods of connected-domain labeling [17] include eight-connected-domain labeling or four-connected-domain labeling. These methods usually require at least two traversals for realizing connected-domain labeling. Moreover, these methods consume more memory and cause greater output delay. As shown in Figure 6a, the shapes of normal stars tend to be Gaussian-like distributions. However, shapes similar to those shown in Figure 6b or Figure 6c also appear. In order to enable the adoption of different shapes of stars and achieve higher real-time performance, this work proposes a new connected-domain labeling method based on preordered pixels. As presented in Figure 7, we design a two-line filter template and a judgment strategy for the connected domain. CP represents the current pixel, and L, UL, U, UR, UR1, UR2 are preorder pixels.

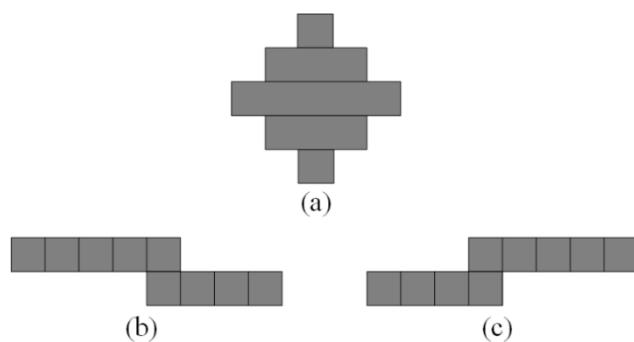


Figure 6. Different shapes of stars: (a) the Gaussian-like distributions; (b,c) the heterotypic distributions.

X	X	X	X	X	X	X	X
X	Preorder pixels		X	X	X	X	X
X	X	UL	U	UR	UR1	UR2	X
X	X	L	CP	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X

Figure 7. Two-line filter template based on the preordered pixels.

As presented in Figure 8, there may exist some scenarios when the distance between two stars is only 1 pixel. In addition, considering that the shape of stars may be heterotypic, such as in Figure 6, the process of marking connected domains must be divided into different cases.

- ① When $CP = 1$ and there already exists a label number of connected domains for the current pixel (the label numbers of all pixels are 0 initially), then we skip the current pixel and analyze the next pixel.
- ② When $CP = 1$ and the current label number $L_N = 0$.

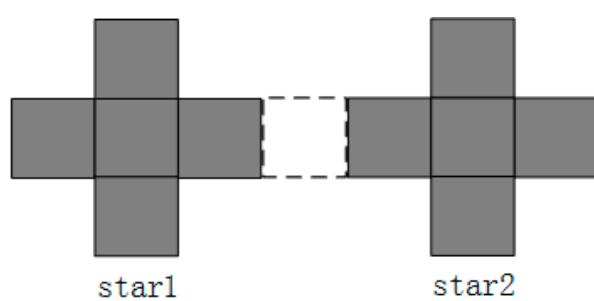


Figure 8. A situation where the distance between two stars is only 1 pixel.

As presented in Figure 9, if $L = 0$, $UL = 0$, $U = 0$, $UR = 0$, $UR1 = 0$, and $UR2 = 0$, the current pixel CP is marked as the new connected domain, and the label number L_N is set to m ($m \geq 0$).

0	0	0	0	0
0	1			

Figure 9. The beginning of a new connected domain.

If at least one of L, UL, U, UR, UR1, UR2 equals 1, we analyze UR1 or UR2 in terms of which entity becomes equal to 1 first. As shown in Figure 10, if one of them is equal to 1, we search the pixels towards left in the first row until we find 0. Then, the previous column number is set to j. At the same time, we search the pixel from CP towards the right until we find 0, and the previous column number is set to i. Finally, we analyze the two column numbers.

X	X	X	UR1	UR2
X	1	R	R1	

Figure 10. The process of searching for a pixel in two rows.

As shown in Figure 11, if $i < (j - 1)$ is satisfied, CP is not related to UR1 and UR2. Here, we compare the preordered pixels L, UL, U, and UR. (1) If all the preordered pixels are equal to 0, the label number L_N of CP is set as m_n . (2) If one of the preordered pixels is equal to 1 and their label numbers are equal to m'_n , then the label number L_N of CP is set as m'_n . On the contrary, if one of the preordered pixels is equal to 1 and their label numbers are different, the label number L_N of CP is set as the maximum value m_b among them. At the same time, we need to build the mapping table between the minimum value m_s and corresponding maximum value m_b .

UL	U	UR	0	1
L	1	1	0	

Figure 11. CP is not related to UR1 and UR2 when $i < (j - 1)$.

As presented in Figure 12, if $i \geq (j - 1)$ is satisfied, CP is related to UR1 and UR2. Then, we compare the preordered pixels L, UL, U, and UR. (1) If all the preordered pixels are equal to 0, the label numbers of the pixels from CP to the rightmost growing point (column number equals i) are set as the label number m_c of UR1 and UR2. (2) If one of the preordered pixels is equal 1, the label numbers of the pixels from CP to the rightmost growing point (column number equals i) are set as the maximum value m_b of L, UL, U, UR, UR1, and UR2. At the same time, we need to build the mapping table between the minimum value m_s and corresponding maximum value m_b . When the entire star image is traversed, all the minimum values of the label numbers should be replaced with the maximum value.

As presented in Figure 13, after the image is completely traversed, we obtain the label number of the star effectively.

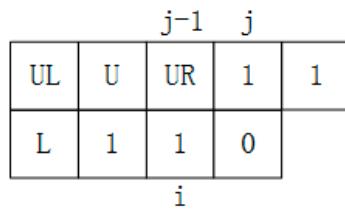


Figure 12. CP is not related to UR1 and UR2 when $i \geq (j - 1)$.

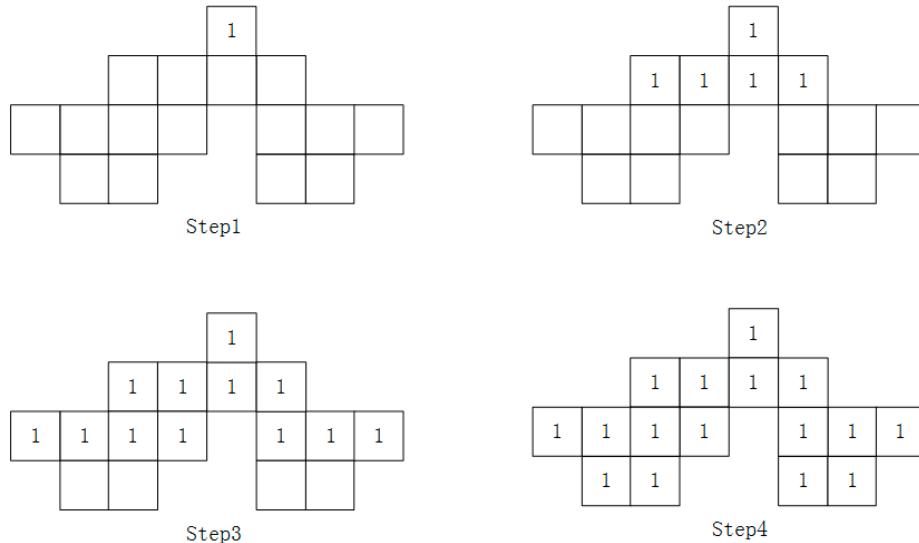


Figure 13. The labeling process of the heterotypic star.

During the process of connected-domain marking, we calculate the accumulation $G(m)$ of gray pixels $I(x, y)$, the product accumulation $X(m)$ of coordinate x and gray pixels $I(x, y)$, and the product accumulation $Y(m)$ of coordinate y and gray pixels $I(x, y)$ in the same label m .

$$\begin{aligned} G(m) &= G(m) + I(x, y), \\ X(m) &= X(m) + x * I(x, y), \\ Y(m) &= Y(m) + y * I(x, y). \end{aligned} \quad (6)$$

As presented in Figure 14, based on the mapping relationship $\{m_s : m_b\}$ between the minimum values m_s and maximum values m_b during the marking of the connected domain, we merge $G(m_s)$ into $G(m_b)$, merge $X(m_s)$ into $X(m_b)$, and merge $Y(m_s)$ into $Y(m_b)$ at the same time.

Figure 14. The merging operation of the minimum label values and maximum label values.

When the merging operation is completed, the coarse centroid coordinates (x_q, y_q) of each connected domain are calculated as follows:

$$\begin{aligned} x_q &= \frac{X(m)}{G(m)}, \\ y_q &= \frac{Y(m)}{G(m)}. \end{aligned} \quad (7)$$

When the coarse centroid coordinates (x_q, y_q) are obtained, the precise centroid coordinates (x_c, y_c) of the stars are calculated by using the method of centroid extraction with the threshold [18]. As presented in Figure 15, considering the coarse centroid coordinates (x_q, y_q) as the center, the local 15×15 image is obtained. Then, the average gray value of 56 pixels in the surrounding area N is calculated and is set as the background value B . The difference between each pixel value $I(x, y)$ and the background value B in the central area M (14×14) is used as the actual gray value of the star after the stray-light suppression. The precise coordinates (x_c, y_c) are obtained as follows:

$$\begin{aligned} x_c &= \frac{\sum_{x=1}^{14} \sum_{y=1}^{14} I'(x, y) * x}{\sum_{x=1}^{14} \sum_{y=1}^{14} I'(x, y)}, \\ y_c &= \frac{\sum_{x=1}^{14} \sum_{y=1}^{14} I'(x, y) * y}{\sum_{x=1}^{14} \sum_{y=1}^{14} I'(x, y)}, \\ I'(x, y) &= \begin{cases} I(x, y) - B & I(x, y) > B \\ 0 & \text{else} \end{cases}. \end{aligned} \quad (8)$$

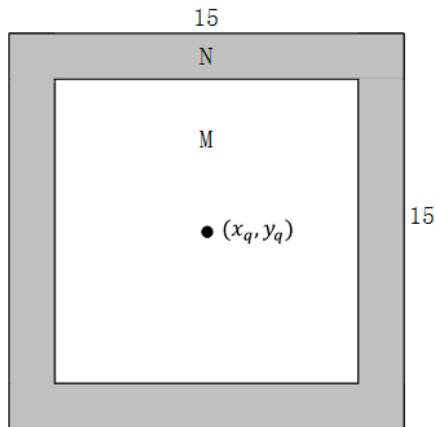


Figure 15. The method of centroid extraction with the threshold.

2.4. The Hardware Implementation of the Proposed Algorithm

As shown in Figure 16, the hardware implementation scheme comprises three pipeline modules, including stray-light suppression, binarization, and star labeling.

Among them, the key steps of hardware implementation [19] are stray-light suppression and star labeling.

The first key step is the suppression of stray light. It includes four steps: highlighted pixels' unification, horizontal erosion and dilation operations, and background subtraction. If a traditional $s \times t$ rectangular window is used for erosion and dilation operations, multiple lines of data have to be stored, which requires multiple FIFOs for storing the row pixels. In order to save the resources and reduce the processing time, $1 \times s$ size window is used for erosion and dilation operations. Therefore, only one FIFO is required to realize the cache of image line, and the search of the maximum and minimum values in the neighborhood pixels can be completed easily by comparing the adjacent cache data. After erosion and dilation operations, the background subtraction operation is used to subtract the dilated image from the original image after pixel alignment. Therefore, the original image is required to complete the delay calibration based on two-stage FIFO.

The second key step is star labeling. The purpose of star labeling is to find the connected domains and calculate the centroid coordinates of each connected domain. As shown in Figure 17, the algorithm proposed in this work is different from the traditional four connected domains or eight connected domains. The preordered binary pixels of

two adjacent rows are used to perform a logical comparison. Therefore, only one FIFO is needed to store the binary image. At the same time, in order to calculate the coarse centroid coordinates, it is necessary to calculate the accumulation $G(m)$ of gray $I(x, y)$, the product accumulation $X(m)$ of coordinate x and gray $I(x, y)$, and the product accumulation $Y(m)$ of coordinate y and gray $I(x, y)$. Therefore, it is necessary to instantiate three internal dual-port RAM. The label number of the connected domain is set as the address of RAM, and $G(m)$, $X(m)$, and $Y(m)$ are set as the data for reading and writing.

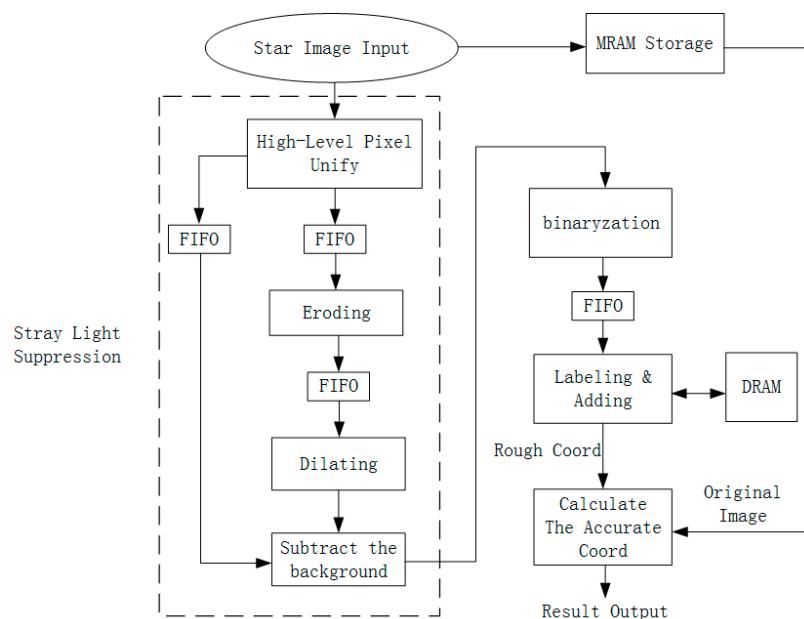


Figure 16. The hardware implementation scheme of the proposed algorithm.

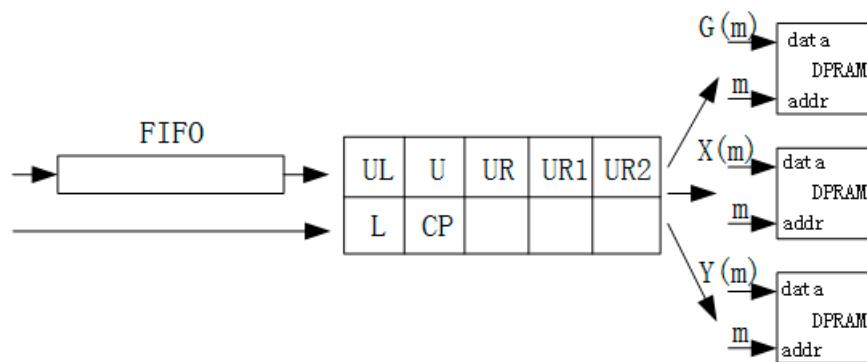


Figure 17. The implementing method of accumulation.

3. Results

3.1. Experimental Conditions

In this section, the algorithm proposed in this work is verified by performing experiments. Based on the empirical results, we set $G_{th} = 20$, $SE1 = 1 \times 20$, $SE2 = 1 \times 40$.

In Section 3.2.1, the simulations are performed to verify the ability of the proposed algorithm in different working conditions. There are eight real star-image sequences (S1–S8) used for performing the experiments. There is moonlight interference, sunlight interference, earth-atmosphere light interference, or daylight interference in the star images. The simulation platform comprises a computer with 2.5 GHz Intel I7 CPU and 16 GB of memory. The simulation software is MATLAB R2012b, and the operating system is Windows 7.

In Section 3.2.2, we compare the resource consumption on three different field programmable gate array (FPGA) platforms and calculate the delay between the last line of the image and accumulations. The simulation software is Modelsim SE 6.4e.

In Section 3.2.3, the field experiment is performed. The experiment platform is a self-developed miniaturized star sensor. The star sensor is installed on the two-dimensional rotation table, and it faces the moon directly. The image resolution is 2048×2048 pixels, and the frame frequency is 4 Hz. The integration time of the star sensor is 100 ms. The real-time attitude quaternion [20,21] and exposure time are output and used to analyze the attitude accuracy and successful identification rate of the star sensor.

The Monte Carlo analysis is used to calculate the successful identification rate [22] for successive frames of a video sequence. The attitude accuracy is analyzed when the two-axis rotation table is operated at the speed of 0 (the actual star is still moving slowly due to the rotation of the Earth) and $1^\circ/\text{s}$, respectively. In order to obtain the attitude accuracy, we calculate the fitting attitude quaternion $[Q_0, Q_1, Q_2, Q_3]$ from the actual attitude quaternion $[q_0, q_1, q_2, q_3]$. Finally, we obtain the standard deviation $[\text{STD}_x, \text{STD}_y, \text{STD}_z]$ from the error value $[\text{Err}_x, \text{Err}_y, \text{Err}_z]$ as follows:

$$\begin{aligned} X_{q0} &= q_0 * Q_0 - q_1 * Q_1 - q_2 * Q_2 - q_3 * Q_3, \\ X_{q1} &= q_0 * Q_1 + q_1 * Q_0 + q_2 * Q_3 - q_3 * Q_2, \\ X_{q2} &= q_0 * Q_2 + q_2 * Q_0 + q_3 * Q_1 - q_1 * Q_3, \\ X_{q3} &= q_0 * Q_3 + q_3 * Q_0 + q_1 * Q_2 - q_2 * Q_1, \end{aligned}$$

$$M = \begin{bmatrix} X_{q1}X_{q1} - X_{q2}X_{q2} - X_{q3}X_{q3} + X_{q0}X_{q0} & 2(X_{q1}X_{q2} + X_{q3}X_{q0}) & 2(X_{q1}X_{q3} - X_{q2}X_{q0}) \\ 2(X_{q1}X_{q2} - X_{q3}X_{q0}) & -X_{q1}X_{q1} + X_{q2}X_{q2} - X_{q3}X_{q3} + X_{q0}X_{q0} & 2(X_{q2}X_{q3} + X_{q1}X_{q0}) \\ 2(X_{q1}X_{q3} + X_{q2}X_{q0}) & 2(X_{q2}X_{q3} - X_{q1}X_{q0}) & X_{q1}X_{q1} - X_{q2}X_{q2} - X_{q3}X_{q3} + X_{q0}X_{q0} \end{bmatrix}, \quad (9)$$

$$\begin{aligned} \text{Err}_x &= \frac{\text{atan}\left(\frac{M[2,3]}{M[3,3]}\right) * 180 * 3600}{\pi}, \quad \text{Err}_y = \frac{-\text{asin}(M[1,3]) * 180 * 3600}{\pi}, \\ \text{Err}_z &= \frac{\text{atan}\left(\frac{M[1,2]}{M[1,1]}\right) * 180 * 3600}{\pi}. \end{aligned}$$

The units of error value $[\text{Err}_x, \text{Err}_y, \text{Err}_z]$ are arc-seconds.

3.2. Experimental Results

3.2.1. The Analysis of Star Detection in Real Image Sequences

In this experiment, the proposed algorithm is applied to analyze the real image sequences. There are eight real star-image sequences (S1–S8) used to perform the experiments, and the eight sequences are acquired under the interference of stray light. Sequence S1 denotes the condition when the earth-atmosphere light entered the field of view; S2 to S7 denote the conditions with moonlight or sunlight interference. S8 denotes the condition when the star image is acquired during daytime. These sequences represent different conditions under stray-light interference. Therefore, we verify the effect of the proposed algorithm in different working conditions.

As presented in Figure 18, the first and third rows represent the original star images with stray-light interference. The second and fourth rows represent the images filled with the detected star. The stars are marked with 10×10 white boxes. In S1, due to the influence of the earth-atmosphere light in the field of view, the average gray value of the whole image is much higher as compared to the average gray value of the black background. Therefore, the proposed algorithm selects five times the gray value of the black background as the highlight threshold by default. It can be seen that no false stars are extracted in

the earth-atmosphere light, and several stars with low signal-to-noise ratios in the black background are extracted successfully.

In S2 to S8, although the scenes are different, there is a common feature. The proportion of stray light on the target surface is small, and the energy is not strong. Therefore, 1.5 times the average gray value of the whole image is set as the highlight threshold by default. Among S2, S3, S5, S6, and S7, a large number of star points are extracted in the field of view. In S4 and S8, the brightest stars in the field of view are also extracted correctly.

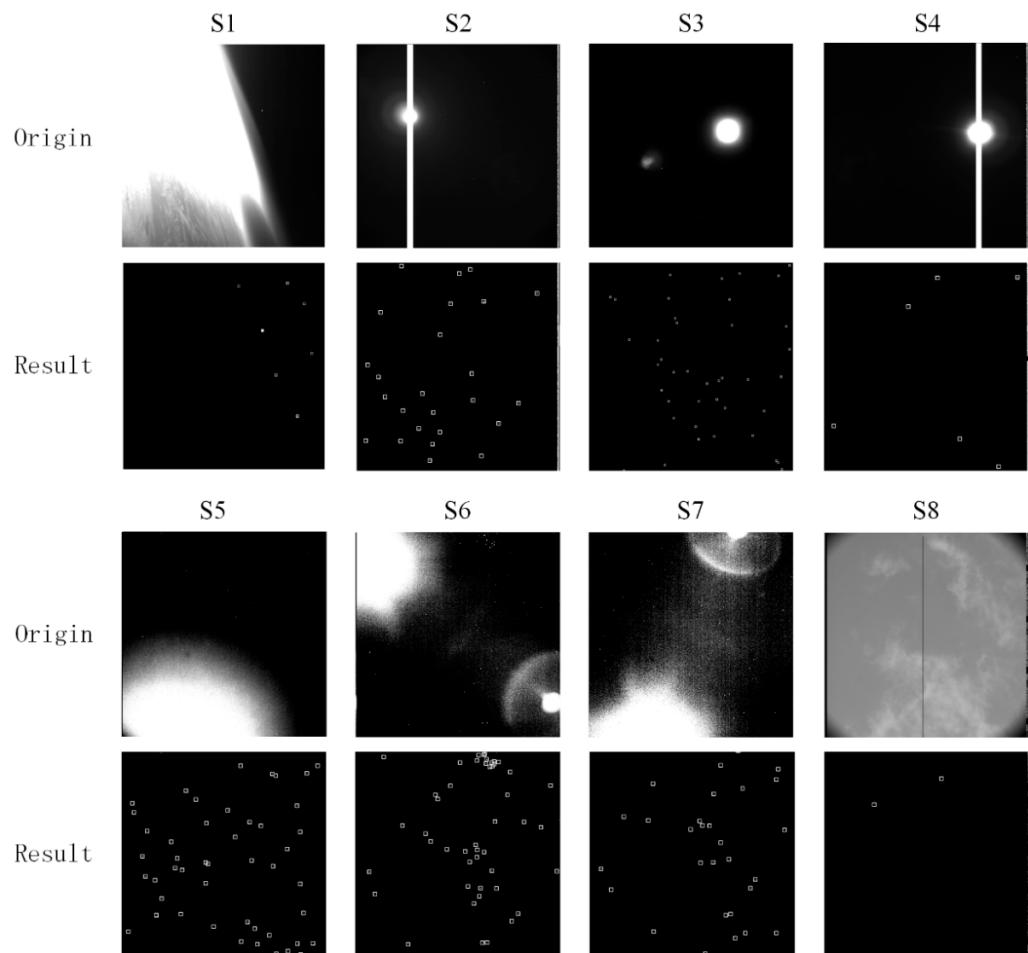


Figure 18. The detection result for eight different sequences.

In order to guarantee the accuracy of calculations, at least four of the brightest stars are chosen for performing subsequent attitude calculations. As shown in Figure 18, the algorithm proposed in this work effectively extracts the brightest stars in the star image under the stray-light interference. This also guarantees the development of subsequent identification algorithms.

3.2.2. The Analysis of Resource Consumption and Delay

First, we implement the algorithm in different FPGA platforms and analyze the resource consumption of the proposed algorithm. The three platforms are commonly used in the development of star sensors.

Table 1 shows the resource consumption results of the algorithm proposed in this paper under three FPGA platforms. The resource consumption also includes the driver of the detector and the logic of data acquisition and storage. As shown in Table 1, the logic consumption does not exceed half of the chip itself, and the RAM consumption is caused by the FIFO cache of image rows and the storage of dual-port RAM. According to

the maximum clock frequency, the algorithm can run at a higher frequency, and it meets the timing requirements of the star sensor.

Second, we analyze the output delay for star coordinates. As presented in Figure 19, *Data_i* denotes the original star image output from the detector. Moreover, *star_gray_data*, *star_x_data*, and *star_y_data* denote the accumulations of gray values and coordinates, which can be used to calculate the precise star coordinates. From Figure 19, it is evident that the delay between the last line of the image and the accumulations is 18.256 μ s (the master clock is 80 MHZ). This means that the centroid coordinates of stars can be quickly calculated as soon as the star image completes the readout, guaranteeing a sufficient time margin for subsequent matching and identification operations.

Table 1. The resource consumption results of the algorithm under three platforms.

Platform	Resource Consumption
xc3s700an-4fgg484	Slice Flip Flops: 50% 4 input LUTs: 65% BRAMs: 20% MULT18×18SIOS: 10% Maximum frequency: 174.642 MHZ CORE: 30.87% GLOBAL: 33.33% RAM/FIFO: 48.21%
A3pe3000L-484FBGA	Maximum frequency: 179.856 MHZ LUT: 6.57% DFF: 5.34% RAM64×18: 3.57% RAM1K18: 13.76% Chip Global: 37.50%
M2S090T-1FG484I	Maximum frequency: 172.1 MHZ

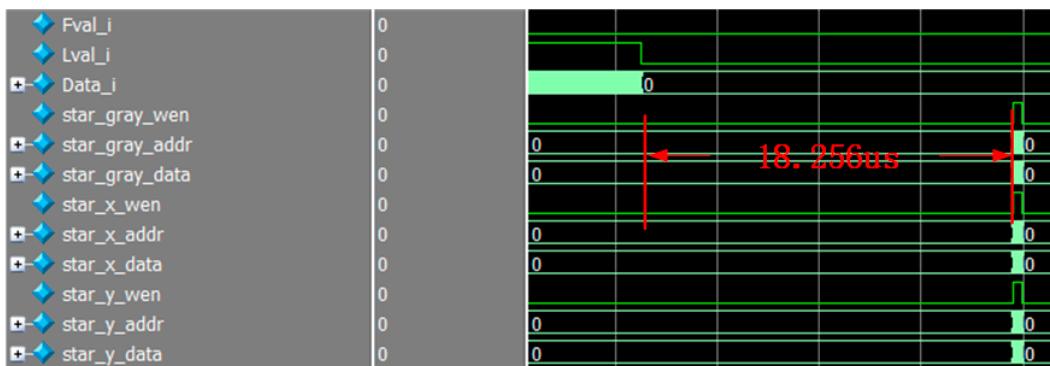


Figure 19. The output delay for star coordinates.

3.2.3. The Field Experiment

The field experiment is performed in Hami, Xinjiang. As presented in Figure 20a, we apply the algorithm in the self-developed star sensor, which is based on the FPGA chip A3PE3000L-484FBGA. The FPGA completes the star-detection algorithm, and the adjacent ARM chip completes the matching and identification processes. Finally, as shown in Figure 20b, the star sensor outputs the effective attitude quaternion and exposure time.

In this experiment, the star sensor faces the moon, and the three-axis attitude accuracy is calculated by using the effective attitude quaternion and exposure time. We calculate the attitude accuracy with the proposed algorithm when the two-axis rotation table operated at the speed of 0 and 1°/s.

As presented in Figures 21 and 22, the attitude accuracy of the X and Y axes is better than 5''(3 σ) when the speed of the star sensor reaches zero. The attitude accuracy of the X

and Y axes is better than $20''(3\sigma)$ when the speed of star sensor reaches $1^\circ/\text{s}$. Therefore, the attitude accuracy meets the requirement of the star sensor, and it proves that the proposed algorithm can guarantee the attitude accuracy at different speeds.

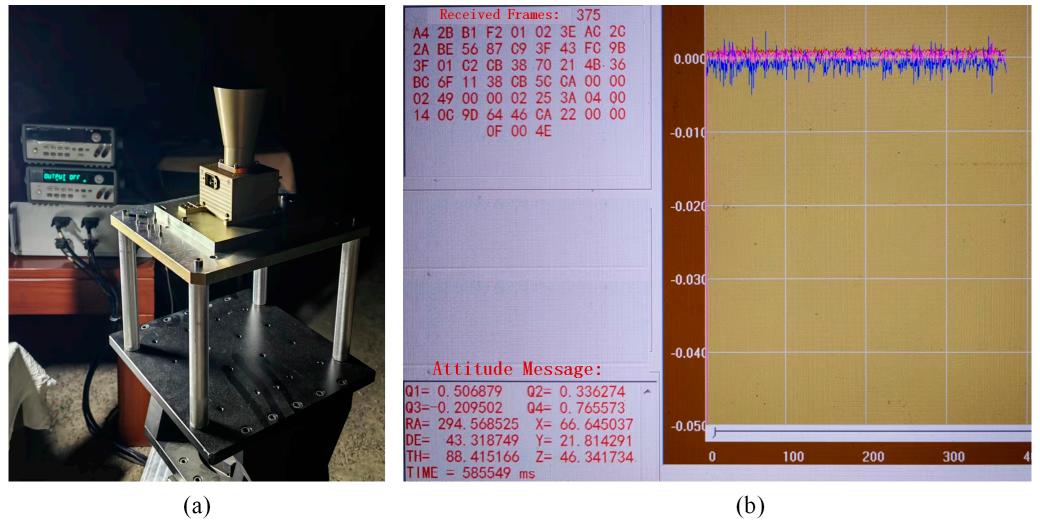


Figure 20. The field experiment: (a) the star sensor installed on the two-dimensional rotation table; (b) the output effective attitude accuracy.

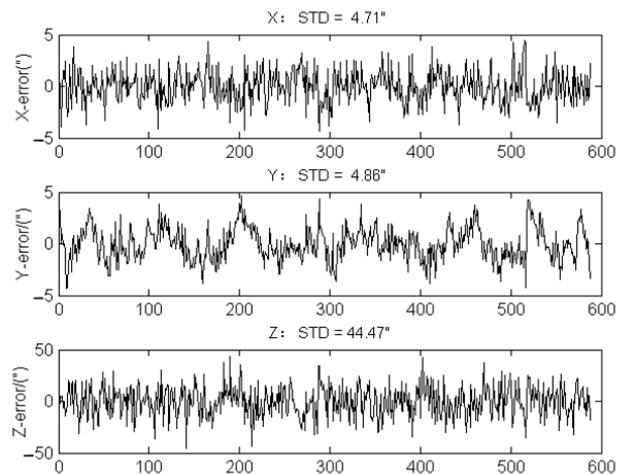


Figure 21. The attitude accuracy when the dynamic speed of the star sensor reaches zero.

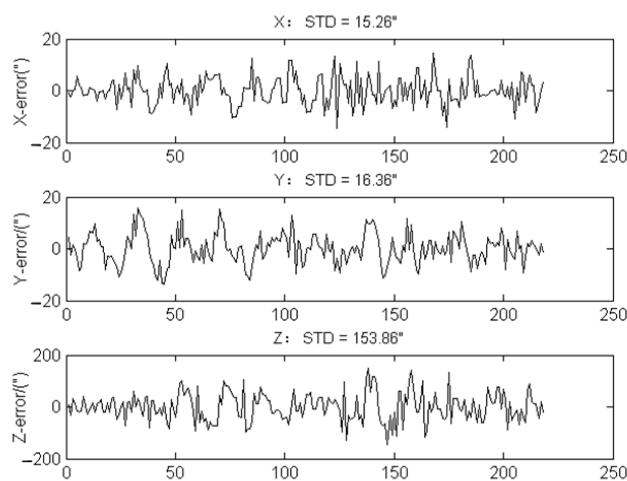


Figure 22. The attitude accuracy when the dynamic speed of the star sensor reaches $1^\circ/\text{s}$.

Except the accuracy calculation, in the long-term moon alignment test (the speed of the star sensor reaches zero), we also analyze the successful identification rate when the moon enters the field of view. As shown in Table 2, using the proposed algorithm, the successful identification rate reaches 98%. While using the local threshold segmentation algorithm, the successful identification rate is only 53%.

Table 2. The successful identification rate obtained using different algorithms.

Situation	The Successful Identification Rate
Local threshold segmentation algorithm	53%
The proposed algorithm	98%

The experimental results show that the algorithm proposed in this work significantly improves the successful identification rate of the star sensor and guarantees the accuracy requirements of attitude when the star sensor is operated at different speeds.

4. Discussion

In order to meet the strict launch requirements of star networks, the development trend of star sensors is focused on smaller sizes and lower power consumption. A smaller size means a smaller lens hood and a worse shading effect. Therefore, star-detection algorithms should consider the interference of stray light.

Compared with the local adaptive threshold method or the multiscale patch-based contrast measure, the output delay of the proposed algorithm is much smaller due to its simple implementation architecture. Furthermore, other algorithms, including the window filtering methods, are limited to the size of the filtering template and weighted value. Therefore, the existing algorithms cannot deal with the stray-light interference in different scenarios.

In this study, we mainly analyze the algorithm performance from three metrics. They are the intuitive capability of star detection, the resource consumption and output delay in engineering applications, and the attitude accuracy and successful identification rate of the star sensor.

The experimental results show that the proposed algorithm possesses excellent stray-light-suppression and star-detection abilities. Apart from these factors, the proposed algorithm also has strong engineering application characteristics. The proposed algorithm fully combines the characteristics of FPGA pipeline and parallelization technology and adopts a small amount of FIFO and row-pixel processing technology. The experimental results show that the resource consumption of the proposed algorithm is small, and the output delay of the algorithm is only 18.256 μs . Using the proposed algorithm, the success-

ful identification rate reaches 98%. The attitude accuracy of the X and Y axes is better than $5''(3\sigma)$ when the speed of the star sensor reaches 0 and better than $20''(3\sigma)$ when the speed of the star sensor reaches $1^\circ/\text{s}$.

However, the proposed algorithm has certain limitations. When the sunlight enters the field of view with a tiny incident angle, the gray levels of most pixels in the image are close to saturation and the stars are completely covered. In this situation, the proposed algorithm cannot detect stars exactly. Therefore, further research is necessary to deal with these special scenarios and further improve the ability to detect stars.

5. Conclusions

This work proposed a new algorithm for extracting stars under the interference of stray light in an efficient manner. The proposed algorithm innovatively unifies the highlighted pixels and performs horizontal erosion and dilation operations based on a large template. The background containing stray light is obtained after erosion and dilation operations, and the stray light is suppressed by subtracting the background from the unified image. In addition, the proposed algorithm marks the connected domain based on the preordered pixels and calculates the centroid coordinates of the stars in each connected domain.

The experimental results show that the proposed algorithm possesses star-detection abilities even when there is interference caused by different stray-light sources. The proposed algorithm also consumes less resources and has a smaller output delay. Moreover, the proposed algorithm is beneficial for improving the successful identification rate and guaranteeing the attitude accuracy of the star sensor at different speeds.

In the future, the proposed algorithm will be applied to an on-orbit task for further verification, to improve its stray-light-suppression capability in the star sensor. This will improve the adaptability of the star sensor in different maneuvering states.

Author Contributions: Conceptualization, K.L.; methodology, K.L.; software, K.L.; validation, K.L. and L.L.; formal analysis, H.L.; investigation, K.L.; resources, R.Z. (Renjie Zhao); data curation, K.L.; writing—original draft preparation, K.L.; writing—review and editing, R.Z. (Rujin Zhao); visualization, K.L.; supervision, E.L.; project administration, K.L.; funding acquisition, R.Z. (Rujin Zhao). All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Program of China under Grant No. 2019YFA0706001.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was supported by the Sichuan Outstanding Youth Science and Technology Talent Project (2022JDJQ0027). This research was also supported by CAS “Light of West China” Program, and Special support for talents from the Organization Department of Sichuan Provincial Party Committee.

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

References

1. Liebe, C.C. Accuracy performance of star trackers—A tutorial. *IEEE Trans. Aerosp. Electron. Syst.* **2002**, *38*, 587–599. [[CrossRef](#)]
2. Clermont, L.; Michel, C.; Stockman, Y. Stray Light Correction Algorithm for High Performance Optical Instruments: The Case of Metop-3MI. *Remote Sens.* **2022**, *14*, 1354. [[CrossRef](#)]
3. Roger, J.C.; Santer, R.; Herman, M.; Deuzé, J.L. Polarization of the solar light scattered by the earth-atmosphere system as observed from the U.S. shuttle. *Remote Sens. Environ.* **1994**, *48*, 275–290. [[CrossRef](#)]
4. Liu, W.D. Lens-Hood Design of Starlight Semi-Physical Experimental Platform. *Laser Optoelectron. Prog.* **2012**, *49*, 162–167. [[CrossRef](#)]
5. Xu, M.Y.; Shi, R.B.; Jin, Y.M.; Wang, W. Miniaturization Design of Star Sensors Optical System Based on Baffle Size and Lens Lagrange Invariant. *Acta Opt. Sin.* **2016**, *36*, 0922001. [[CrossRef](#)]

6. Kwang-Yul, K.; Yoan, S. A Distance Boundary with Virtual Nodes for the Weighted Centroid Localization Algorithm. *Sensors* **2018**, *18*, 1054. [[CrossRef](#)]
7. Fialho, M.; Mortari, D. Theoretical Limits of Star Sensor Accuracy. *Sensors* **2019**, *19*, 5355. [[CrossRef](#)] [[PubMed](#)]
8. He, Y.Y.; Wang, H.L.; Feng, L.; You, S.H.; Lu, J.H.; Jiang, W. Centroid extraction algorithm based on grey-gradient for autonomous star sensor. *Opt.-Int. J. Light Electron Opt.* **2019**, *194*, 162932. [[CrossRef](#)]
9. Seyed, M.F.; Reza, M.M.; Mahdi, N. Flying small target detection in ir images based on adaptive toggle operator. *IET Comput. Vis.* **2018**, *12*, 527–534. [[CrossRef](#)]
10. Gonzalez, R.C.; Woods, R.E.; Masters, B.R. Digital Image Processing, Third Edition. *J. Biomed. Opt.* **2009**, *14*, 029901. [[CrossRef](#)]
11. Zhang, Y.; Du, B.; Zhang, L. A spatial filter based framework for target detection in hyperspectral imagery. In Proceedings of the 2013 5th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), Gainesville, FL, USA, 26–28 June 2013; pp. 1–4. [[CrossRef](#)]
12. Yu, L.W.; Mao, X.N.; Jin, H.; Hu, X.C.; Wu, Y.K. Study on Image Process Method of Star Tracker for Stray Lights Resistance Filtering Based on Background. *Aerospace Shanghai* **2016**, *33*, 26–31. [[CrossRef](#)]
13. Wang, H.T.; Luo, C.Z.; Wang, Y.; Wang, X.Z.; Zhao, S.F. Algorithm for star detection based on self-adaptive background prediction. *Opt. Tech.* **2009**, *35*, 412–414. [[CrossRef](#)]
14. Wei, Y.; You, X.; Li, H. Multiscale patch-based contrast measure for small infrared target detection. *Pattern Recognit.* **2016**, *58*, 216–226. [[CrossRef](#)]
15. Lu, R.T.; Yang, X.G.; Li, W.P.; Ji, W.F.; Li, D.L.; Jing, X. Robust infrared small target detection via multidirectional derivative-based weighted contrast measure. *IEEE Geosci. Remote Sens. Lett.* **2020**, *1*, 1–5. [[CrossRef](#)]
16. Lu, K.L.; Liu, E.H.; Zhao, R.J.; Zhang, H.; Lin, L.; Tian, H. A Curvature-Based Multidirectional Local Contrast Method for Star Detection of a Star Sensor. *Photonics* **2022**, *9*, 13. [[CrossRef](#)]
17. Perri, S.; Spagnolo, F.; Corsonello, P. A Parallel Connected Component Labeling Architecture for Heterogeneous Systems-on-Chip. *Electronics* **2020**, *9*, 292. [[CrossRef](#)]
18. Wan, X.W.; Wang, G.Y.; Wei, X.G.; Li, J.; Zhang, G.J. Star Centroiding Based on Fast Gaussian Fitting for Star Sensors. *Sensors* **2018**, *18*, 2836. [[CrossRef](#)]
19. Chen, W.; Zhao, W.; Li, H.; Dai, S.; Han, C.; Yang, J. Iterative Decoding of LDPC-Based Product Codes and FPGA-Based Performance Evaluation. *Electronics* **2020**, *9*, 122. [[CrossRef](#)]
20. Han, J.L.; Yang, X.B.; Xu, T.T.; Fu, Z.Q.; Chang, L.; Yang, C.L.; Jin, G. An End-to-End Identification Algorithm for Smearing Star Image. *Remote Sens.* **2021**, *13*, 4541. [[CrossRef](#)]
21. Schiattarella, V.; Spiller, D.; Curti, F. A novel star identification technique robust to high presence of false objects: The multi-poles algorithm. *Adv. Space Res.* **2017**, *59*, 2133–2147. [[CrossRef](#)]
22. Rijlaarsdam, D.; Yous, H.; Byrne, J.; Oddenino, D.; Furano, G.; Moloney, D. Efficient star identification using a neural network. *Sensors* **2020**, *20*, 3684. [[CrossRef](#)] [[PubMed](#)]

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