

Article

Star Sensor Denoising Algorithm Based on Edge Protection

Kaili Lu ^{1,2}, Enhai Liu ^{1,2,*}, Rujin Zhao ^{1,2}, Hui Zhang ^{1,2} and Hong Tian ^{1,2}

- ¹ Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610000, China; lukaili@ioe.ac.cn (K.L.); zhaorj@ioe.ac.cn (R.Z.); hzhang@ioe.ac.cn (H.Z.); tianhong@ioe.ac.cn (H.T.)
- ² Key Laboratory of Science and Technology on Space Optoelectronic Precision Measurement, Chinese Academy of Sciences, Chengdu 610000, China
- * Correspondence: lehioe@163.com

Abstract: Single-pixel noise commonly appearing in a star sensor can cause an unexpected error in centroid extraction. To overcome this problem, this paper proposes a star image denoising algorithm, named Improved Gaussian Side Window Filtering (IGSWF). Firstly, the IGSWF algorithm uses four special triangular Gaussian subtemplates for edge protection. Secondly, it exploits a reconstruction function based on the characteristic of stars and noise. The proposed IGSWF algorithm was successfully verified through simulations and evaluated in a star sensor. The experimental results indicated that the IGSWF algorithm performed better in preserving the shape of stars and eliminating the single-pixel noise and the centroid estimation error (CEE) value after using the IGSWF algorithm was eight times smaller than the original value, six times smaller than that after traditional window filtering, and three times smaller than that after the side window filtering.

Keywords: single-pixel noise; star sensor; denoising; subtemplates; reconstruction function



Citation: Lu, K.; Liu, E.; Zhao, R.; Zhang, H.; Tian, H. Star Sensor Denoising Algorithm Based on Edge Protection. *Sensors* **2021**, *21*, 5255. <https://doi.org/10.3390/s21165255>

Academic Editor: Marco Carratu

Received: 22 June 2021
Accepted: 30 July 2021
Published: 4 August 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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 attitude measurement instrument that takes a star as a working object and calculates multiple reference vectors using multiple stars. The centroid extraction [2,3] of the star-point region has a great influence on the final precision of a star sensor. The factor affecting accuracy of centroid extraction can be attributed to the motion blur and the background noise [4]. The motion blur is generated by the fast motion of satellite under dynamic conditions. It will bring about the blur of the stars. There have been many studies [5–8] on this aspect. Most of them focused on the method of restoration of the star image. Generally, the background noise can be divided into two categories: single-pixel noise with only a single pixel, and large-area noise with a continuous change in the gray level. Single-pixel noise can be attributed to two main sources: the nonuniform response of a complementary metal oxide semiconductor (CMOS) detector [9] and the impact of cosmic radiation particles [10]. Large-area noise with a continuous change in the gray level is usually influenced by sunlight, moonlight, and earth-atmosphere light [11]. Considering the actual engineering requirements in the aerospace field, this paper aimed to develop a single-point noise elimination method.

As shown in Figures 1 and 2, the nonuniform response of a CMOS detector and the cosmic radiation particles can cause significant dense single-pixel noise. To deal with the single-pixel noise, traditional window filter algorithms, such as BOX filtering [12], mean filtering [13], and Gaussian filtering [14,15], can be used. However, these methods have a kinetic disadvantage, and the original image can be damaged during the denoising process. This disadvantage will be discussed in detail in Section 2, and the traditional window filter algorithm will be compared with the proposed Improved Gaussian Side Window Filtering (IGSWF) algorithm in Section 3.

Recently, many studies on single-pixel noise and star-sensor applications have been conducted. Schmidt [16] proposed a method to deal with the single-pixel noise by using the

background value prediction, where the background value is obtained through multiframe accumulation and the noise is eliminated using subtraction. However, this method places high requirements on the speed of the star sensor, and it is suitable only for the nonuniform response of a CMOS detector. Zheng [17] proposed an improved method based on the Schmidt method and adjusted the scale of the correction domain at high speed, which could adapt to the higher dynamic speed. The results showed that this method could merely deal with the nonuniform response of the CMOS detector having a fixed position, but random noise caused by cosmic radiation particles could not be handled well. The two above-mentioned algorithms cannot eliminate the two types of single-pixel noise at the same time without destroying the precision of a star sensor. Hence, the methods for eliminating single-pixel noise still face challenges, and further analyses are necessary.

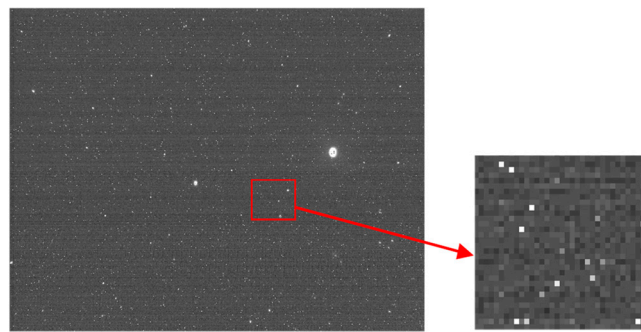


Figure 1. The noise caused by the impact of cosmic radiation particles.

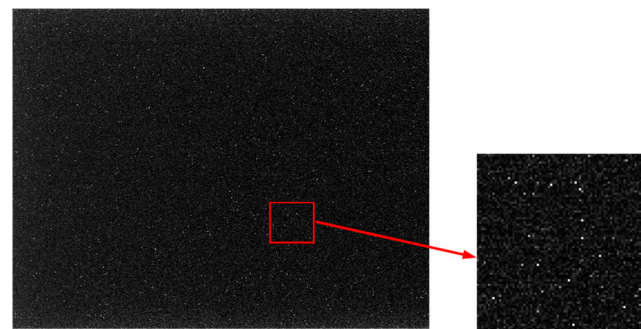


Figure 2. The noise caused by a nonuniform response of a CMOS detector.

To deal with the single-pixel noise caused by both a nonuniform response of a CMOS detector and cosmic radiation particles simultaneously, this paper proposes a denoising algorithm, named IGSWF, which is based on edge protection.

The proposed algorithm is based on the side window filtering (SWF) algorithm proposed by Hui Yin [18], which can smooth the background noise and prevent the object boundary from being damaged at the same time. In addition, the algorithm is improved for application in star images. Firstly, unlike SWF, the proposed IGSWF algorithm uses four triangular Gaussian subtemplates for noise filtering. Secondly, based on the shape and energy characteristics of the star point, background, and image noise, a suitable calculation function f_{min} for eliminating single-pixel noise in a star image was defined. The proposed algorithm was verified by experiments, and the experimental results showed that the proposed IGSWF algorithm can effectively protect the edge of the preprocessed star point from being damaged and successfully eliminate the single-pixel noise in a star image at the same time. Finally, it was verified that the proposed IGSWF algorithm was favorable to improving the precision of centroid extraction and the accuracy of star identification.

The remainder of this paper is organized as follows: Section 2 explains the SWF principles and introduces the proposed IGSWF algorithm. Section 3 presents the experimental results. Finally, Sections 4 and 5 conclude the paper and present future directions for work.

2. Materials and Methods

2.1. Side Window Filtering Principle

In this section, the principle of side window filtering (SWF) is introduced.

The coordinates (x_0, y_0) of the center of mass can be expressed in the following way:

$$\begin{cases} x_0 = \frac{\sum_{x=1}^m \sum_{y=1}^n (F_{sc}(x,y) + F_{se}(x,y) + F_b(x,y))x}{\sum_{x=1}^m \sum_{y=1}^n (F_{sc}(x,y) + F_{se}(x,y) + F_b(x,y))} \\ y_0 = \frac{\sum_{x=1}^m \sum_{y=1}^n (F_{sc}(x,y) + F_{se}(x,y) + F_b(x,y))y}{\sum_{x=1}^m \sum_{y=1}^n (F_{sc}(x,y) + F_{se}(x,y) + F_b(x,y))} \end{cases} \quad (1)$$

where $F_{sc}(x, y)$ denotes the central pixel of a star, $F_{se}(x, y)$ denotes the pixel on the star edge, and $F_b(x, y)$ represents the background.

In the field of visual processing, window-smoothing filtering is a common method for denoising, and some of the most common filtering methods are box filtering, mean filtering, and Gaussian filtering. Window-smoothing filtering can be expressed as:

$$I_o = \sum_{\omega_j \in B} \omega_j I_j, \quad (2)$$

where B represents the filtering window centered at the current pixel j , I_j is the input image region, I_o is the output pixel, and ω_j denotes the weight value.

To demonstrate the filtering effect, traditional Gaussian filtering with a window size r of 3 and a scale σ of 0.8 was applied to a star image, and the results are presented in Figure 3.

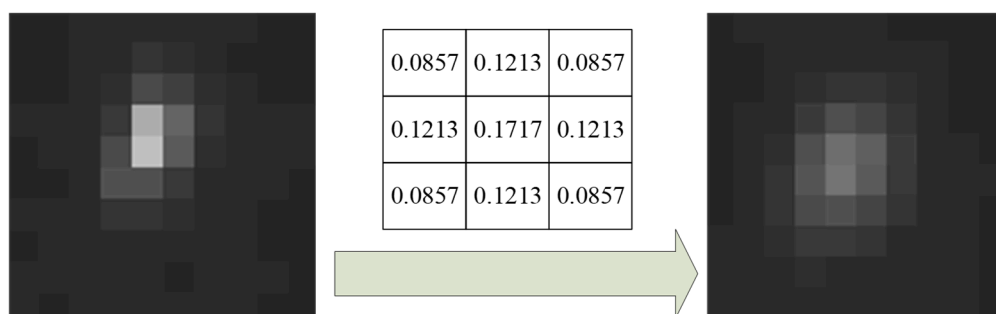


Figure 3. The result of traditional Gaussian filtering.

As shown in Figure 3, after denoising by Gaussian filtering, the inherent shape of the star point was destroyed, the energy of the star point was dispersed, and the edge became blurred. Thus, there was a significant drop in the values of $F_{sc}(x, y)$ and $F_{se}(x, y)$ in Equation (1). Accordingly, the weight values of the central pixel and the pixel on the star edge decreased, which resulted in a deviation of (x_0, y_0) . Hence, a method to preserve the energy and edge of the star point is urgently needed.

Furthermore, Taylor's expansion [19] of a pixel (x, y) can be expressed as:

$$f(x, y) = f(x_0, y) + \frac{f'(x_0, y)}{1!} (x - x_0) + \frac{f''(x_0, y)}{2!} (x - x_0)^2 + \dots, \quad (3)$$

where $f(x, y)$ denotes the intensity value at the pixel (x, y) .

For an edge point (x, y) , the left limit $(x - \varepsilon, y)$ and the right limit $(x + \varepsilon, y)$ can be defined when $\varepsilon > 0$.

Assuming that $x_0 = x - \varepsilon$ and function $f(x, y)$ is differentiable at x_0 , then Taylor's expansion of a pixel (x, y) can be expressed as:

$$f(x - 2\varepsilon, y) \approx f(x_0) + f'(x_0)(x - x_0) = f(x - \varepsilon, y) + f'(x - \varepsilon, y)(-\varepsilon). \quad (4)$$

Similarly, when $x_0 = x + \varepsilon$, Taylor's expansion of a pixel (x, y) becomes:

$$f(x + 2\varepsilon, y) \approx f(x_0) + f'(x_0)(x - x_0) = f(x + \varepsilon, y) + f'(x + \varepsilon, y)(\varepsilon). \quad (5)$$

According to Equations (4) and (5), the pixel value on one side of the edge point can be obtained by the pixel value in the field on that side. Therefore, it is necessary to ensure that the boundary cannot be crossed during the filtering process. In other words, when a pixel j is positioned on the image edge, the edge of a filter B should be aligned with the center pixel j . Therefore, the SWF algorithm takes each pixel as a potential edge point and generates several different filtering subwindows. Then, it aligns the edge or a corner position of the filtering window with the edge point. Finally, the best reconstruction result after filtering is selected as the final filtering result.

2.2. The Proposed Improved Gaussian Side Window Filtering Algorithm

As mentioned before, the proposed IGSWF algorithm is based on SWF. The main objective of IGSWF is to remove the single-pixel noise from a star image while ensuring that the shape and energy characteristics of the stars are not affected.

The main innovation of the proposed IGSWF algorithm was that four triangular Gaussian subtemplates were designed, and a suitable reconstruction function f_{min} was defined to obtain the best-possible final filtering results. The proposed IGSWF was more suitable than the SWF for star image denoising.

2.2.1. Filter Template Design

According to the description in Section 2.1, a series of filter templates should be designed when the current processing pixel is aligned with the edge or corner position of a filter subtemplate. Considering that the distribution of star points on a detector is similar to the Gaussian distribution, the Gaussian filter template was chosen as the final filter template in this work.

Aiming at a full consideration of the edge feature and processing capacity of a processing chip (which is usually a field programmable gate array) at the same time, the traditional Gaussian filter template is separated into four subtemplates, namely, up, down, left, and right subtemplates, as shown in Figure 4. The shape of the subtemplates is the same (a triangle). The four filtering templates are used simultaneously during the filtering process.

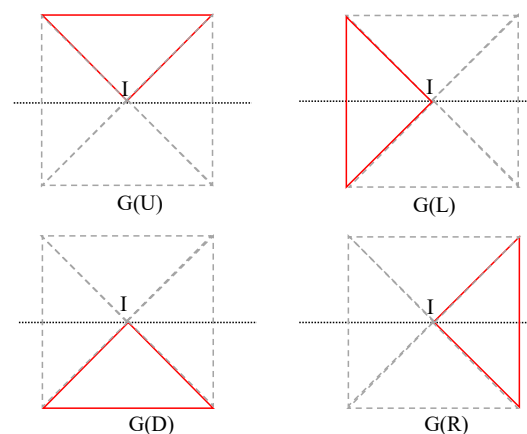


Figure 4. The four triangular Gaussian subtemplates.

2.2.2. Improved Gaussian Side Window Filtering Steps

The block diagram of the IGSWF algorithm is presented in Figure 5, where it can be seen that it includes three main steps: Gaussian subtemplate convolution, calculation of relative minimum of filtering values, and the final image determination.

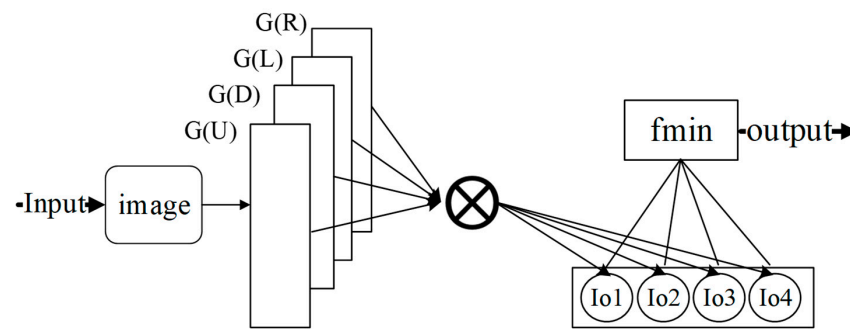


Figure 5. The block diagram of the IGSWF algorithm.

Step 1: Gaussian Subtemplate Convolution

The pixels are convolved using the Gaussian subtemplates in the four directions: up, down, left, and right. Based on Equation (2), the four following expressions can be obtained as follows:

$$\begin{aligned}
 I_{o2} &= \sum_{(i,j) \in G(L)} \omega_{(i,j)} I_{(i,j)}, \\
 I_{o3} &= \sum_{(i,j) \in G(D)} \omega_{(i,j)} I_{(i,j)}, \\
 I_{o4} &= \sum_{(i,j) \in G(R)} \omega_{(i,j)} I_{(i,j)}, \\
 &\text{and} \\
 I_{o1} &= \sum_{(i,j) \in G(U)} \omega_{(i,j)} I_{(i,j)},
 \end{aligned} \tag{6}$$

where I_{o1} , I_{o2} , I_{o3} , and I_{o4} represent the filtering values of the four templates in the up, left, down, and right directions, respectively.

Step 2: Relative Minimum of the Filtering Values

The relative minimum of the filtering values of the four templates is expressed as follows:

$$I_m = f_{min}(I_{o1}, I_{o2}, I_{o3}, I_{o4}, I_c), \tag{7}$$

where I_m represents the relative minimum value; and I_c represents the input value, which is also the center pixel. The main objective is to find the best reconstruction function f_{min} , which will be discussed in detail in Section 2.2.3.

Step 3: Final Image Determination

The output image after denoising is generated based on the calculated pixel instead of the original pixel. Step 1 is repeated until the last pixel has been calculated. As shown in Figure 6, the surrounding edge of the image (green fields) is filled with adjacent pixels (yellow fields) uniformly, and the filter templates (red dotted line) shift from left to right and from top to bottom.

2.2.3. Reconstruction Function Calculation

The calculation of the reconstruction function f_{min} is mainly based on the shape and energy characteristics of the star point, background, and image noise. Therefore, by analyzing a large amount of shooting data, pixels in a star image can be categorized into four types, as shown in Figure 7.

According to the classification in Figure 7, the reconstruction function f_{min} , which is suitable for a star image, can be defined as follows.

For the type presented in Figure 7a, the reconstruction function f_{min} satisfies condition A, which is as follows:

$$A = \{I_{ok} < I_c \ \& \ \Delta I_c < T_{dmin} \ \& \ (I_c - I_{min}) > T_{dmin}\} \tag{8}$$

For the type presented in Figure 7b, the reconstruction function f_{min} satisfies condition B, which is as follows:

$$B = \{I_{ok} < I_c \ \& \ ((I_c - I_{min}) > T_{dmax} \ | \ (I_{max} - I_{min}) > T_{dela})\} \quad (9)$$

For the types presented in Figure 7c,d, this means that the remaining pixels do not satisfy conditions A and B at the same time.

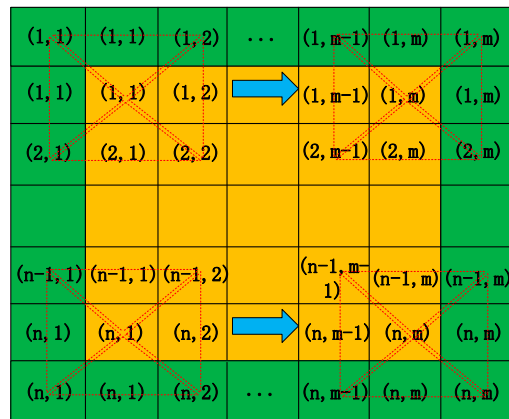


Figure 6. Filter template processing.

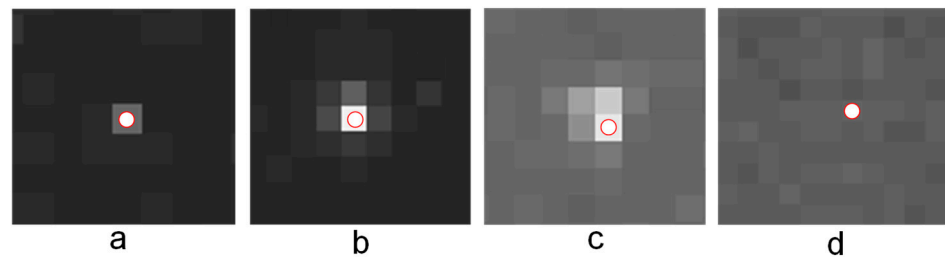


Figure 7. Four types of pixels in a star image: (a) a single pixel, which is characterized by a relatively bright pixel in the center and flat black pixels around it; (b) a standard central pixel of a star point, which is characterized by a relatively bright center pixel and approximate Gaussian distribution of the surrounding pixels; (c) an irregular central pixel of the Gaussian star point, which is characterized by a relatively bright pixel in the center and approximate Gaussian distribution on several sides, and the remaining part is the gentle background; and (d) pixels in a flat region with a small difference from the other pixels.

In the above-mentioned condition, I_{ok} ($k = 1:4$) represents the filtering value of a subtemplate in the up, left, down, or right direction; I_c represents the current pixel; ΔI_c represents the d-value around the current pixel I_c ; I_{max} represents the maximum L_2 distance; I_{min} represents the minimum L_2 distance; T_{dmin} is the lower limit for the d-value of smooth background; T_{dmax} is the lower limit for the d-value of I_c and I_{min} ; and lastly, T_{dela} is the lower limit for the d-value of I_{max} and I_{min} .

The maximum L_2 distance I_{max} , the minimum L_2 distance I_{min} , and the d-value ΔI_c are calculated as follows:

$$\begin{aligned} I_{max} &= \operatorname{argmax}_{k \in \{1,2,3,4\}} \|I_{ok} - I_c\|_2^2 \\ I_{min} &= \operatorname{argmin}_{k \in \{1,2,3,4\}} \|I_{ok} - I_c\|_2^2 \\ &\text{and} \\ \Delta I_c &= \begin{cases} I_c(i-1, j) - I_c(i-2, j) \\ I_c(i+1, j) - I_c(i+2, j) \\ I_c(i, j-1) - I_c(i, j-2) \\ I_c(i, j+1) - I_c(i, j+2). \end{cases} \end{aligned} \quad (10)$$

In addition, several constants, namely, T_{dmin} , T_{dmax} , and T_{dela} are set as follows:

$$T_{dmin} = 8; T_{dmax} = 65; T_{dela} = 20. \quad (11)$$

The above three constants are related to the gray distribution of the background and star. They are chosen according to empirical values, and they have been tested using different scenes. The effect of IGSWF on noise was also verified by the experiment, as presented below.

When a pixel is under condition A, it represents single-pixel noise. As mentioned before, single-pixel noise should be eliminated. Therefore, it is necessary to maximize the distance between the input and output pixels, which means that the output pixel obtained after filtering should be as far as possible from the input pixel and as close as possible to the background pixel. In this work, the maximum L_2 distance denoted as I_{max} was calculated and multiplied by 0.9 (it represented an approximation coefficient used to make the result closer to the background than I_{max}); in this way, the noise is weakened to bring it to the level of the background.

When a pixel is under condition B, it represents the brightest pixel; to maintain the star centroid, the output value is set to be the same as the input.

When a pixel is under other conditions, it mainly represents the star edges and background. To preserve both edges and energy, it is necessary to minimize the distance between the input and output pixels, which means that the output pixel obtained after filtering should be as close as possible to the input pixel. Thus, the minimum L_2 distance I_{min} is chosen as the output value in this case.

According to the characteristic given above, function f_{min} can be obtained as:

$$f_{min} = \begin{cases} 0.9 * I_{max} & \text{condition A} \\ I_c & \text{condition B} \\ I_{min} & \text{other} \end{cases}. \quad (12)$$

By using function f_{min} , the optimum output value can be obtained, which can replace the input value.

2.2.4. Filter Window Parameter Selection

As shown in Figure 4, the collection S can be obtained as:

$$S = \{G(U), G(L), G(D), G(R)\} = \text{gauss}(r, \sigma) = \frac{1}{2\pi\sigma} e^{-((x-\frac{r-1}{2}-1)^2 + (y-\frac{r-1}{2}-1)^2)/2\sigma^2} \quad (13)$$

In other words, the four subtemplates can be combined into a two-dimensional Gaussian template, and thus the next key point is to select proper values of the Gaussian window size r and scale σ .

The size of the star points on the focal plane of a CMOS detector is usually between 2×2 and 7×7 . Hence, the range of the Gaussian window size should be from 3×3 to 7×7 .

Because single-pixel noise should be eliminated as much as possible, σ should be as large as possible. However, a larger σ will weaken the energy distribution of a star, and appropriate values of σ were found to be 0.8, 1.0, and 1.2.

As shown in Figure 8, at the same value of σ , the increase in the filtering window size r results in an abnormal energy distribution of star points. Considering that the high-brightness area is usually small and the Gaussian window size r is larger, the star edge can be destroyed, as shown in Figure 8, where $r = 5$ and $r = 7$.

Therefore, when the Gaussian window size r was set to 3 and the scale σ was set to 1.2, the ideal original shape of the star point could be obtained, and the noise could be suppressed to a greater extent.

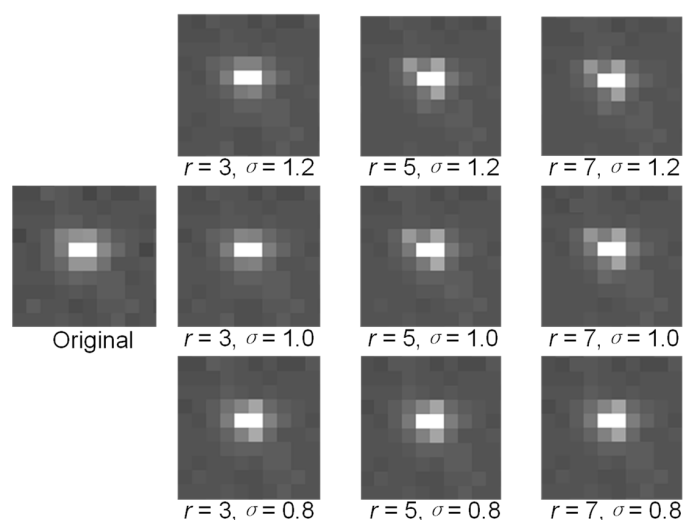


Figure 8. Results for different values of r and σ .

3. Results

3.1. Experimental Conditions

In this section, the performance of the proposed IGSWF algorithm in protecting the shape of stars and eliminating the single-pixel noise was evaluated experimentally.

Firstly, a simulation experiment was carried out. An actual star image with a resolution of 1536×1536 that was taken by a certain type of star sensor was used in the experiment. The operating platform was a 2.5 GHz Intel I7 CPU with 16 GB of memory, and the simulation software was MATLAB R2012b. The proposed IGSWF algorithm was used to substantiate the effect in eliminating single-pixel noise. It was also compared with Gaussian filtering, box filtering, mean filtering, and SWF in preserving the shape of a star point during the denoising process.

Secondly, an application experiment was carried out. The IGSWF algorithm was applied to a star sensor when it docked with the dynamical star simulator. The star sensor was made using the AM3358 processor and the CMV4000 detector. The resolution of the detector in the star sensor was 1536×1536 , and the field of view was $15^\circ \times 15^\circ$. The star points in the dynamic star simulator moved at a rate of $0.06^\circ/s$, which is the normal operating rate of a satellite. The programming IDE tool was CCS v5.4. The centroid estimation error (CEE) [20] for six kinds of stars with different sizes was analyzed. In addition, the proposed IGSWF algorithm was compared with Gaussian filtering, box filtering, mean filtering, Zheng processing [17], and SWF in the value of the centroid estimation error (CEE). The accuracy of star identification [21] was also tested when we chose different algorithms.

The specific coefficients of the four subtemplates are shown in Figure 9. The values of these coefficients could be obtained from Equation (13). The coefficients were guaranteed to meet the normalization conditions.

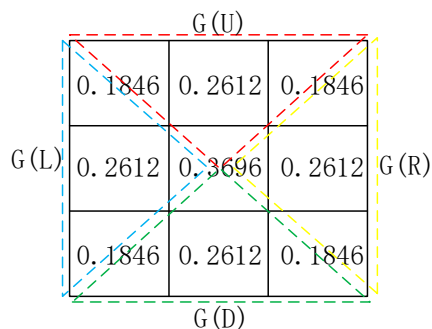


Figure 9. Final filtering parameters.

3.2. Simulation Experiment

3.2.1. Denoising Effect on the Star Image

Firstly, the denoising effect of IGSWF on a star image with single-pixel noise was analyzed. As shown in Figure 10, the single-pixel noise position in the star image was not fixed in the original image. The local star-point region before and after IGSWF denoising was compared.

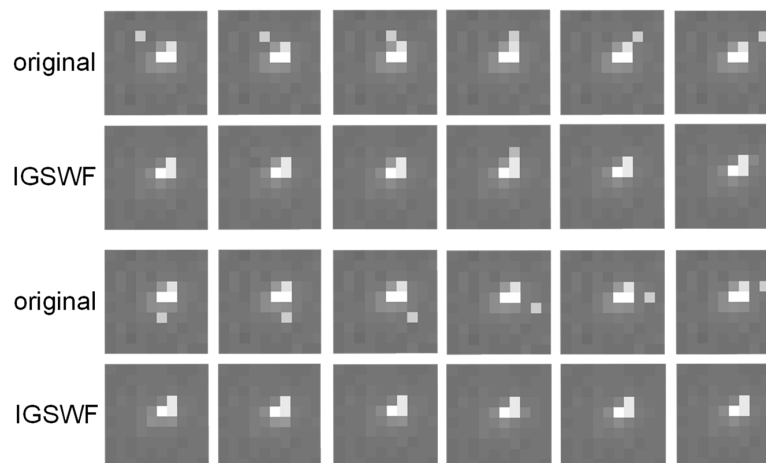


Figure 10. Star-point region before and after denoising.

The results in Figure 10 indicated that wherever the single-pixel noise existed in a star image, the IGSWF algorithm could effectively filter the noise without destroying the star shape. Therefore, by applying the proposed algorithm, the star-point region could be kept smooth while mitigating the influence on the star shape, which was beneficial for the calculation of the center of mass in the star-point region.

Secondly, as shown in Figures 11 and 12, the actual star image, which was taken in orbit by a certain type of star sensor, was simulated.

When a star sensor is in strong radiation zones, such as the South Atlantic Anomaly (SAA area) [22], there will be much random single-pixel noise in a star image, while fixed single-pixel noise is also generated when a star sensor is running at a high temperature. Therefore, the real effect in the original and denoised images was compared. The three-dimensional distribution of the grayscale in the star image was also used to analyze the performance of IGSWF.

The results shown in Figures 11 and 12 indicate that the star image became much cleaner and smoother after IGSWF processing regardless of the star sensor's position. Moreover, the denoising effect of IGSWF was not affected by the earth-atmosphere light. Furthermore, a large amount of single-pixel noise encountered in orbit was effectively suppressed by IGSWF, while the gray level of the star point remained unchanged.

Therefore, based on the results, it can be concluded that the IGSWF algorithm is suitable for the application in orbit and can deal with a severe single-pixel noise problem in the SAA area. The IGSWF algorithm can be extended to other scenarios that include radiation.

3.2.2. Comparison with Other Algorithms

The performance of IGSWF was compared with those of the traditional Gaussian filter, box filter, and mean filter for the case of a star image with the single-pixel noise. The star image, the three-dimensional distribution of the grayscale, and the local star-point region were compared for different algorithms.

From Figure 13a–e, it can be found that the star image after the IGSWF algorithm was cleaner than the original image. Compared with the other algorithms, the background area after IGSWF was smoother. In addition, the gray value of the star after IGSWF changed less than after the other filtering algorithms, and the gray value of the single-pixel noise was

practically invisible. In addition, the IGSWF algorithm could keep the original shape and energy distribution of the star points relatively unchanged while denoising. This advantage was not observed in the traditional algorithms. Therefore, the IGSWF algorithm had a stronger ability to protect the star shape and eliminate the single-pixel noise from a star image than the traditional Gaussian filtering, box filtering, and mean filtering algorithms.

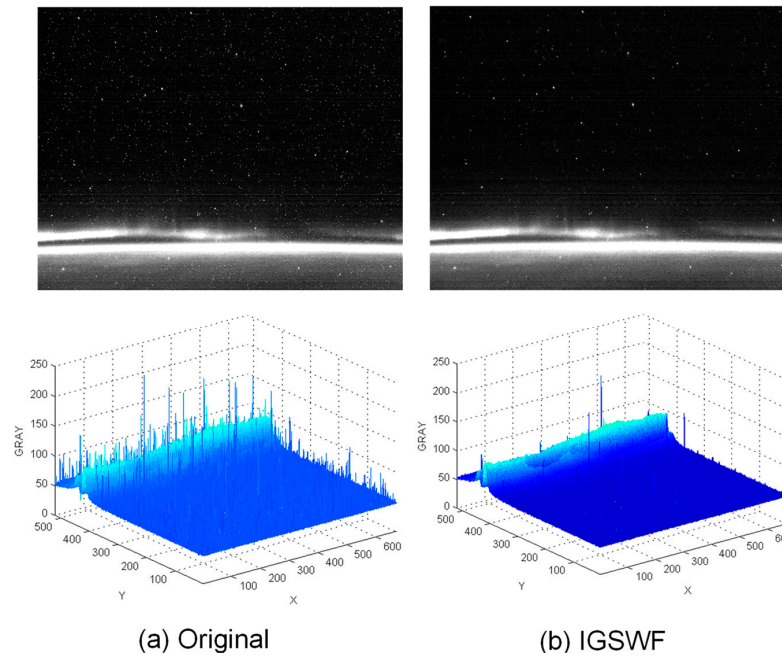


Figure 11. Comparison before and after denoising in SAA area 1: (a) the star image and the three-dimensional distribution of the grayscale before denoising; (b) the star image and the three-dimensional distribution of the grayscale after IGSWF processing.

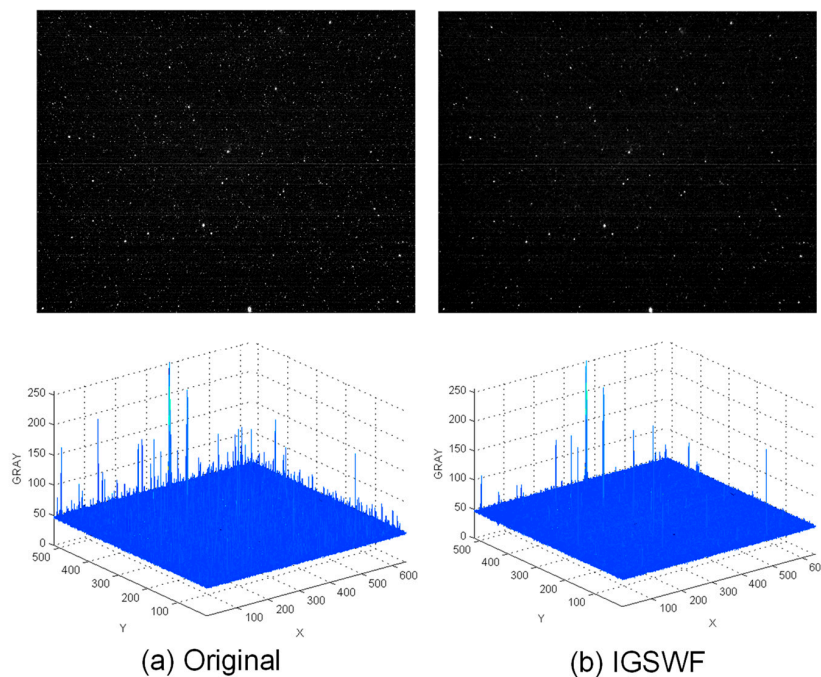


Figure 12. Comparison before and after denoising in SAA area 2: (a) the star image and the three-dimensional distribution of the grayscale before denoising; (b) the star image and the three-dimensional distribution of the grayscale after IGSWF processing.

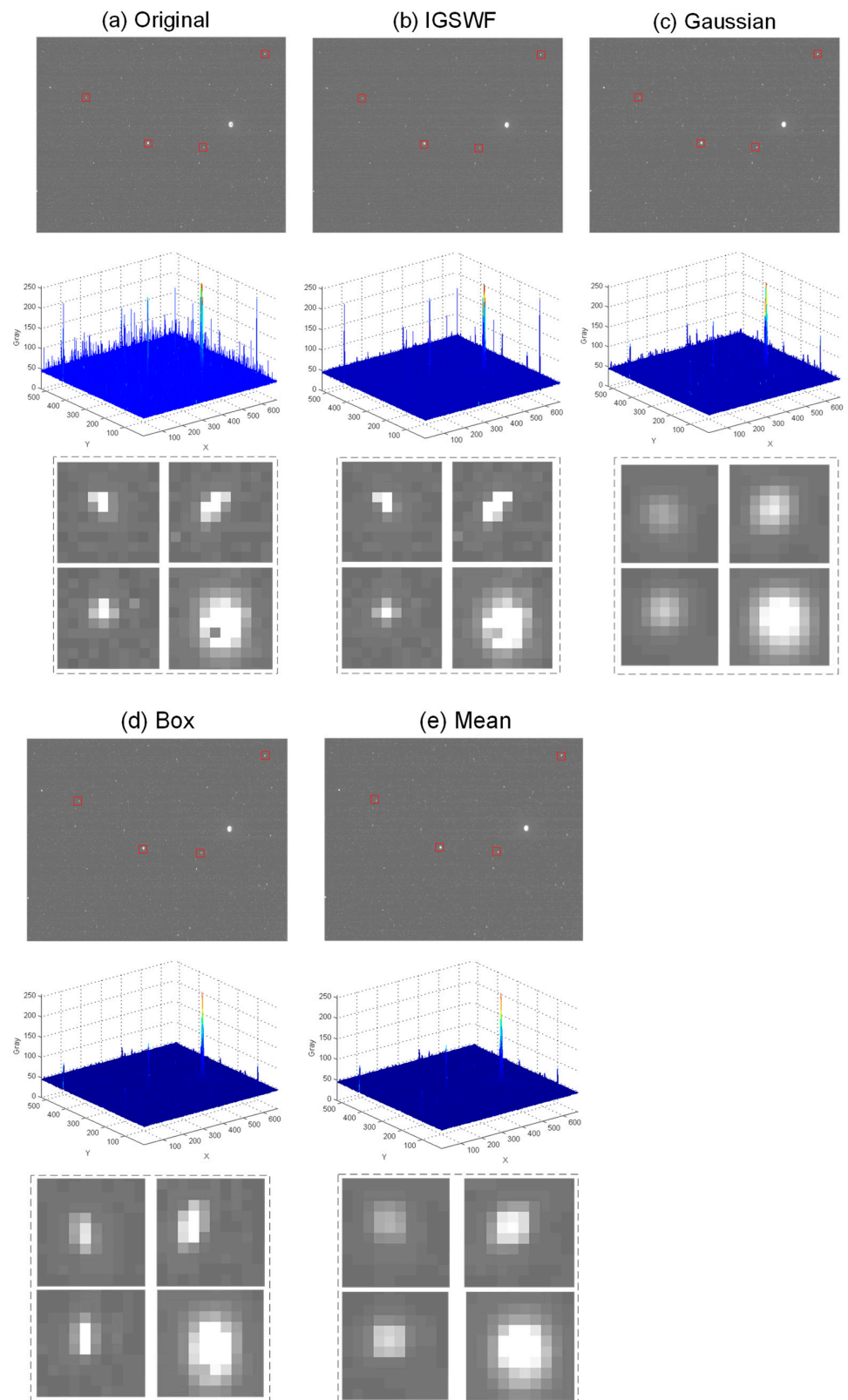


Figure 13. Comparison of the IGSWF algorithm with Gaussian filtering, box filtering, and mean filtering: (a) before denoising; (b–e) after denoising by the IGSWF algorithm, Gaussian filtering, box filtering, and mean filtering, respectively. The first lines are star images, the second lines are the three-dimensional distribution of the grayscale in the star image, and the third lines are the local star points.

At the same time, the proposed IGSWF was compared with SWF regarding energy distribution and precision. Because SWF had a similar effect in protecting the star shape as the IGSWF algorithm, only the differences in the energy distribution and precision of star points were analyzed. In Figure 14, the first column represents the original energy distribution, the second column represents the energy distribution after IGSWF, and the third column represents the energy distribution after SWF. The relative proportion of the grayscale in the original star image was the closest to that after the IGSWF, which indicated that IGSWF maintained the energy distribution better than the SWF.

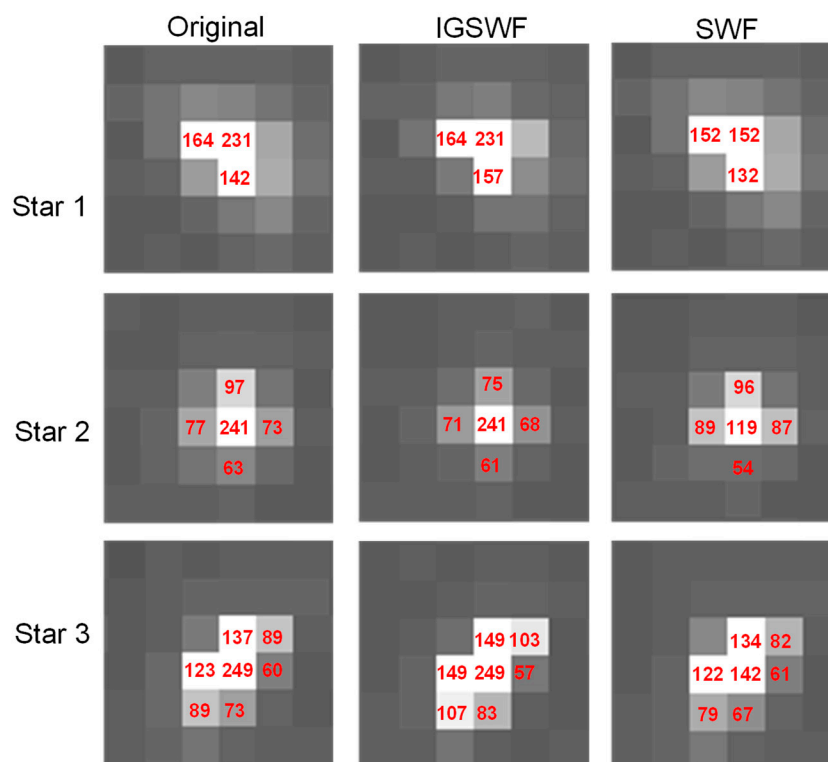


Figure 14. Comparison between IGSWF and SWF.

In Figure 14, the center of mass was calculated, and the calculation results are shown in Table 1.

Table 1. The center of mass.

Center of Mass	Origin (x_o, y_o)	IGSWF (x_n, y_n)	SWF (x'_n, y'_n)
Star 1	(36.883, 133.291)	(36.898, 133.258)	(36.910, 133.378)
Star 2	(446.025, 292.821)	(446.003, 292.860)	(446.000, 92.723)
Star 3	(608.824, 43.866)	(608.798, 43.882)	(608.766, 43.829)

Then, the deviation between the different centers of mass was calculated by:

$$\varepsilon_n = \sqrt{(x_n - x_o)^2 + (y_n - y_o)^2}$$

and

$$\varepsilon'_n = \sqrt{(x'_n - x_o)^2 + (y'_n - y_o)^2}, \quad (14)$$

and it was obtained that:

$$\varepsilon_1 = 0.036249, \varepsilon_2 = 0.044777, \varepsilon_3 = 0.030529$$

and

$$\varepsilon'_1 = 0.091093, \varepsilon'_2 = 0.101139, \varepsilon'_3 = 0.068797 \quad (15)$$

By comparing ε_n with ε'_n , it was found that the center of mass after IGSWF was closer to the original center of mass than that after the SWF. Consequently, the proposed IGSWF algorithm can protect the energy distribution better than the SWF, and IGSWF is more favorable for centroid extraction than SWF.

3.3. Application Experiment

In Figures 15 and 16, the dynamic star simulator was used to generate star points, and the star sensor was aligned with the optical center of the dynamic star simulator for testing.

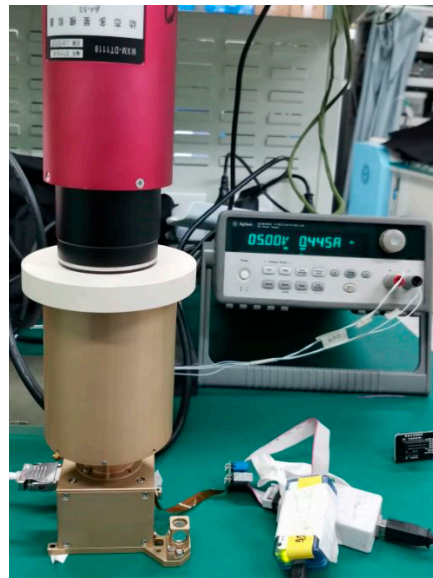


Figure 15. The verification platform.

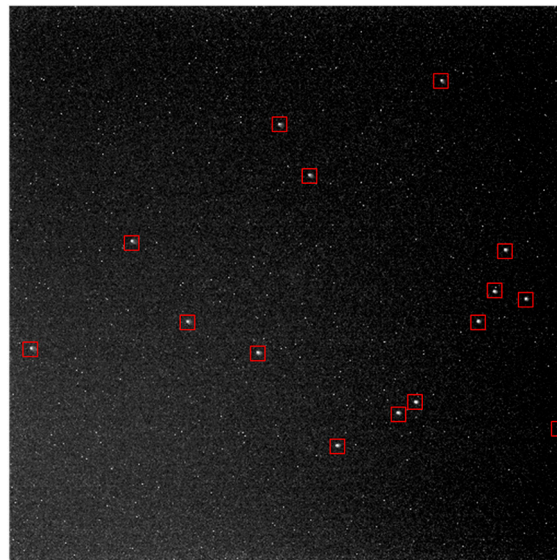


Figure 16. Star image created by the dynamic star simulator.

To show the accuracy of centroid extraction, the CEE value was introduced. The CEE value represented the variance between the theoretical and actual centers of mass, and it was calculated by:

$$CEE = \frac{1}{n} \sum_{i=1}^n \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \quad (16)$$

where n denoted the actual number of available star points obtained in each frame of the star image, (x_i, y_i) was the centroid position of the i th star point calculated in the current frame image, and (x_c, y_c) represented the theoretical centroid position of the i th star point in the current frame image. The theoretical centroid position was obtained using the star simulator. In the meantime, the centroid extraction with the threshold was used as a centroid extraction method.

Firstly, in Figure 17, the dynamic star simulator generated six kinds of stars with different sizes. The six kinds of sizes corresponded to six kinds of star magnitude. Star 1 to Star 6 corresponded to 1 Mv, 2 Mv, 3 Mv, 4 Mv, 5 Mv, and 6 Mv, respectively. We calculated the CEE value ($n = 1$) for each star after IGSWF processing when the stars were rolling.

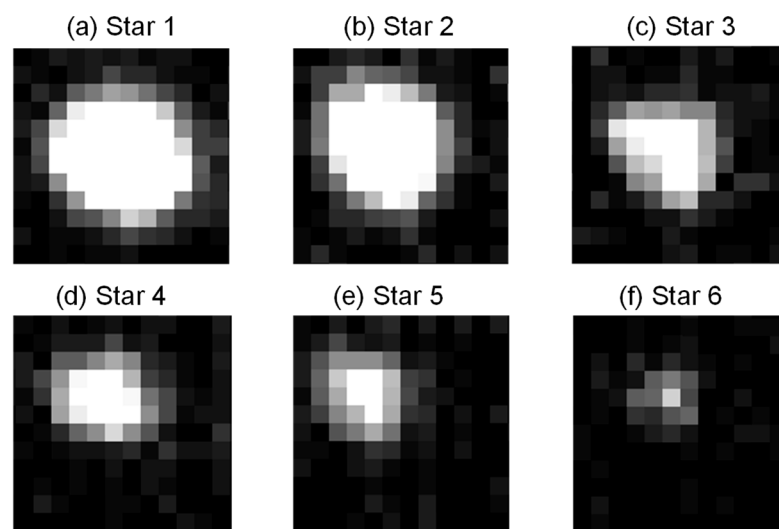


Figure 17. The six kinds of stars with different sizes: (a–f) Star 1 to Star 6 corresponded to 1 Mv, 2 Mv, 3 Mv, 4 Mv, 5 Mv, and 6 Mv, respectively.

As shown in Figure 18, it could be found that the CEE value was smaller if the size of the star was larger. This means that the center of mass of a star with larger size was much closer to original state after IGSWF filtering.

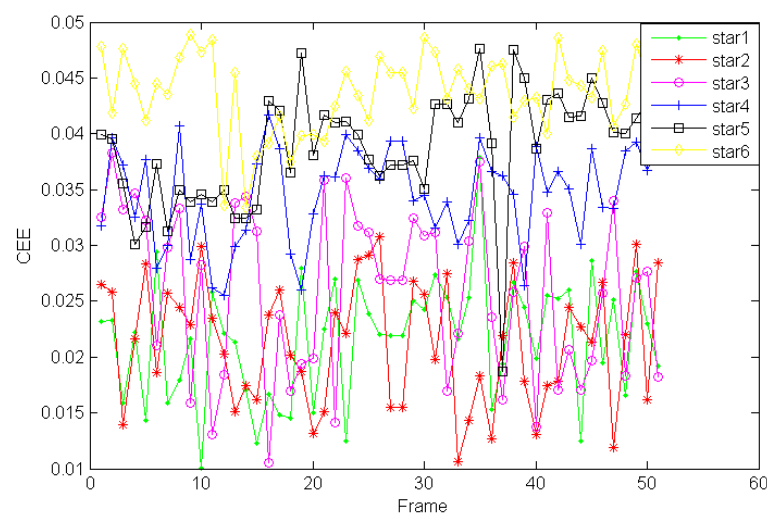


Figure 18. The CEE curves of the different stars.

Secondly, in this experiment, the performance of centroid extraction was evaluated for the star image in the original state and the star image after Gaussian filtering, box filtering, mean filtering, Zheng processing [17], SWF processing, and IGSWF processing. The CEE

curves for 1000 continuous frames of images were obtained, and the error curves before and after correction were drawn.

As shown in Figure 19, the deviation of the centroid extraction result was very large before denoising, and the CEE value fluctuated around 0.5. When traditional Gaussian filtering was adopted, the CEE value was superior to the original value and fluctuated around 0.35. When box filtering or mean filtering was adopted, the CEE value fluctuated around 0.4. After denoising by the SWF method, the deviation of the centroid extraction result was reduced quickly, and the CEE value fluctuated around 0.18. After processing according to Zheng [17], the CEE value fluctuated around 0.15, which was close to the SWF method. Lastly, after denoising by the IGSWF method, the deviation of the centroid extraction result was significantly reduced, and the CEE value fluctuated around 0.06. Based on the results, the CEE value after using the IGSWF algorithm was eight times smaller than the original value; six times smaller than that after traditional Gaussian filtering, box filtering, and mean filtering; and nearly three times smaller than that after the SWF and Zheng processing.

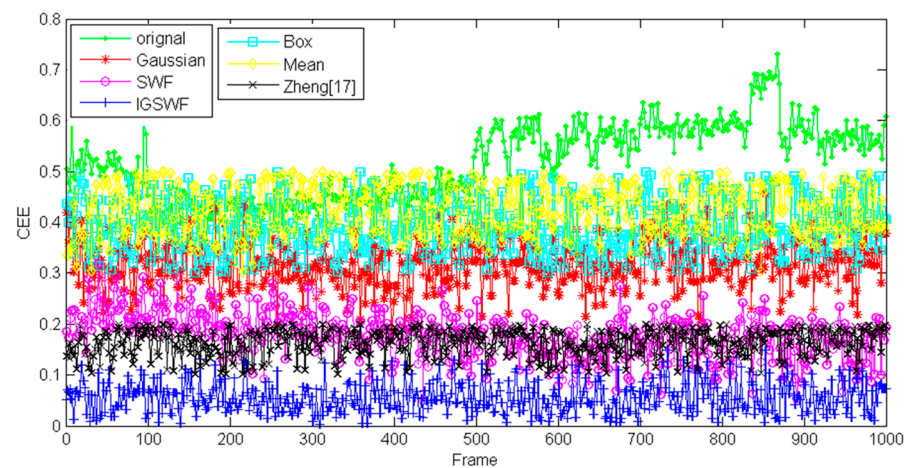


Figure 19. Comparison of the CEE curves of the different algorithms.

The comparison results show that the SWF and Zheng algorithm were superior to the traditional window-filtering algorithms. Although traditional Gaussian filtering could suppress the single-pixel noise, it could not improve the accuracy of the centroid extraction because it could not protect the shape and energy distribution of stars, which led to the deviation of the centroid extraction result. It should be noted that in this experiment, the proposed IGSWF algorithm achieved higher accuracy and better stability than the other compared methods. Moreover, the proposed IGSWF could improve the centroid extraction accuracy by several times and acquire a more precise attitude for the satellite.

Thirdly, the centroid extraction accuracy could improve the accuracy of star identification in the following process. Therefore, to verify the effect of the proposed IGSWF algorithm on the step of star identification, we compared the identification rate when we chose different algorithms, such as Gaussian filtering, SWF, and IGSWF. Typically, the triangular-based star-identification algorithm was selected.

In Table 2, it can be seen that the identification rate without a denoising algorithm was 85%. Due to the various errors of centroid extraction, the process of identification was instable. After Gaussian filtering and SWF processing, the identification rate was improved to 89.5% and 94.3%, respectively. After IGSWF processing, the identification rate was improved to 99.2%. So, it can be concluded that the proposed IGSWF algorithm was beneficial in improving the accuracy of star identification.

Table 2. The identification rate.

Situations	Identification Rate (%)
Original	85.0%
Gaussian	89.5%
SWF	94.3%
IGSWF	99.2%

The results of the experiments listed above showed that the proposed IGSWF algorithm had a better ability than other algorithms in protecting the shape of stars and eliminating the single-pixel noise, and could improve the centroid extraction accuracy and the identification rate while maintaining high stability.

4. Discussion

In this paper, the IGSWF algorithm for star-image denoising based on the idea of edge protection was proposed. The proposed algorithm used four triangular Gaussian subtemplates, which is convenient for edge protection and engineering applications. In addition, based on the shape and energy characteristics of the star point, background, and image noise, a suitable calculation function f_{min} for eliminating the single-pixel noise in a star image was introduced.

The proposed algorithm was verified by a simulation experiment, and the experimental results showed that the proposed IGSWF algorithm could maintain the edge characteristics of star points better than the traditional Gaussian filtering, box filtering, and mean filtering. It was also verified that IGSWF had higher precision than SWF in centroid extraction. Moreover, when processing the star image in-orbit, the IGSWF algorithm was adaptive to various environments of a satellite in orbit, and had an outstanding performance in single-pixel noise suppression.

Finally, through the application experiment, the accuracy and stability of the IGSWF algorithm were verified by comparing the CEE curves of centroid extraction for the original star image and the star image after Gaussian filtering, box filtering, mean filtering, Zheng processing, SWF processing, and IGSWF processing. The comparison results showed that, when the IGSWF algorithm was used, the accuracy (CEE value) of the centroid extraction was improved by nearly eight times compared to the original image, six times compared to the traditional window filtering, and three times compared to the SWF and Zheng. It also showed that the proposed IGSWF algorithm could improve the identification rate of the star sensor.

5. Conclusions

We concluded that the proposed IGSWF algorithm was better than the traditional window filter algorithms, Zheng, and SWF in preserving the shape of stars and eliminating the single-pixel noise, which was favorable to improving the precision of centroid extraction and the accuracy of identification rate.

In the future, the IGSWF algorithm will be applied to an on-orbit task for further verification and to improve the capability of the star sensor.

Author Contributions: Conceptualization, E.L. and K.L.; methodology, R.Z. and K.L.; software, K.L.; validation, K.L. and H.Z.; formal analysis, H.T.; investigation, K.L.; resources, R.Z.; data curation, K.L.; writing—original draft preparation, K.L.; writing—review and editing, R.Z.; visualization, K.L.; supervision, E.L.; project administration, K.L.; funding acquisition, R.Z. 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 Nos. 2019YFA0706001 and 2016YFB0501105.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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. Kwang-Yul, K.; Yoan, S. A Distance Boundary with Virtual Nodes for the Weighted Centroid Localization Algorithm. *Sensors* **2018**, *18*, 1054. [[CrossRef](#)]
3. 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](#)]
4. Ni, Y.M.; Dai, D.K.; Tan, W.F.; Wang, X.S.; Qin, S.Q. Attitude-correlated frames adding approach to improve signal-to-noise ratio of star image for star tracker. *Opt. Express* **2019**, *27*, 15548–15564. [[CrossRef](#)]
5. Sun, T.; Xing, F.; You, Z.; Wei, M. Motion-blurred star acquisition method of the star tracker under high dynamic conditions. *Opt. Express* **2013**, *21*, 20096–20110. [[CrossRef](#)]
6. Mu, Z.; Wang, J.; He, X.; Wei, Z.; He, J.; Zhang, L.; Lv, Y.; He, D.L. Restoration method of a blurred star image for a star sensor under dynamic conditions. *Sensors* **2019**, *19*, 4127. [[CrossRef](#)]
7. Wang, S.; Zhang, S.; Ning, M.; Zhou, B. Motion blurred star image restoration based on MEMS gyroscope aid and blur kernel correction. *Sensors* **2018**, *18*, 2662. [[CrossRef](#)]
8. Zhang, C.; Zhao, J.; Yu, T.; Yuan, H.; Li, F. Fast restoration of star image under dynamic conditions via lp regularized intensity prior. *Aerosp. Sci. Technol.* **2016**, *61*, 29–34. [[CrossRef](#)]
9. Bao, J.Y.; Xing, F.; Sun, T.; You, Z. CMOS imager non-uniformity response correction-based high-accuracy spot target localization. *Appl. Opt.* **2019**, *58*, 4560–4568. [[CrossRef](#)] [[PubMed](#)]
10. Brau, J.; Igonkina, O.; Potter, C.; Sinev, N. Investigation of radiation damage in the SLD CCD vertex detector. *IEEE Trans. Nucl. Sci.* **2004**, *51*, 1742–1746. [[CrossRef](#)]
11. 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](#)]
12. Pires, B.R.; Singh, K.; Moura, J.M.F. Approximating image filters with box filters. In Proceedings of the 2011 18th IEEE International Conference on Image Processing IEEE, Brussels, Belgium, 11–14 September 2011; pp. 85–88. [[CrossRef](#)]
13. Zhou, Z.; Lam, E.Y.; Lee, C. Nonlocal Means Filtering Based Speckle Removal Utilizing the Maximum a Posteriori Estimation and the Total Variation Image Prior. *IEEE Access* **2019**, *99*, 99231–99243. [[CrossRef](#)]
14. Hu, H.; Zhang, B.; Xu, D.; Xia, G. Battery Surface and Edge Defect Inspection Based On Sub-Regional Gaussian and Moving Average Filter. *Appl. Sci.* **2019**, *9*, 3418. [[CrossRef](#)]
15. Terejanu, G.; Singla, P.; Singh, T.; Scott, P.D. Adaptive Gaussian Sum Filter for Nonlinear Bayesian Estimation. *IEEE Trans. Autom. Control* **2011**, *56*, 2151–2156. [[CrossRef](#)]
16. Schmidt, U. Intelligent error correction method applied on an active pixel sensor based star tracker. In Proceedings of the Detectors and Associated Signal. Processing II, Jena, Germany, 14 October 2005. [[CrossRef](#)]
17. Zheng, X.; Ye, Z.; Yang, Q.; Sun, S. An error correction method for pixel inhomogeneity of star sensor based on multi-frame correlation filtering is presented. *Aerosp. Control Appl.* **2017**, *43*, 31–36.
18. Yin, H.; Gong, Y.; Qiu, G. Side Window Filtering. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 8750–8758. [[CrossRef](#)]
19. Zhang, K.W.; Hao, G.; Sun, S.L. Weighted Measurement Fusion Particle Filter for Nonlinear Systems with Correlated Noises. *Sensors* **2018**, *18*, 3242. [[CrossRef](#)] [[PubMed](#)]
20. Luo, L.; Xu, L.; Zhang, H. Improved centroid extraction algorithm for autonomous star sensor. *Image Process. Lett.* **2015**, *9*, 901–907. [[CrossRef](#)]
21. Abderrahim, N.; Zoubir, A.; Mohammed, E.C. Improved triangular-based star pattern recognition algorithm for low-cost star trackers. *J. King Saud Univ. Comput. Inf. Sci.* **2021**, *33*, 258–267. [[CrossRef](#)]
22. Hussien, F.; Ghamry, E.; Fathy, A. A Statistical Analysis of Plasma Bubbles Observed by Swarm Constellation during Different Types of Geomagnetic Storms. *Universe* **2021**, *7*, 90. [[CrossRef](#)]