applied sciences

MDPI

Article

On-Board Flickering Pixel Dynamic Suppression Method Based on Multi-Feature Fusion

Liangjie Jia [1,2,3], Peng Rao [1,2,*], Xin Chen [1,2] and Shanchang Qiu [1,2,*]

[1] Shanghai Institute of Technical Physics, Chinese Academy of Sciences, Shanghai 200083, China; jialiangjie@mail.sitp.ac.cn (L.J.); chenxin@mail.sitp.ac.cn (X.C.)
[2] Key Laboratory of Intelligent Infrared Perception, Chinese Academy of Sciences, Shanghai 200083, China
[3] University of Chinese Academy of Sciences, Beijing 100049, China
* Correspondence: peng_rao@mail.sitp.ac.cn (P.R.); qiushanchang@126.com (S.Q.)

Abstract: The blind pixel suppression is the key preprocess to guarantee the real-time space-based infrared point target (IRPT) detection and tracking. Meanwhile, flickering pixels, as one of the blind pixels, is hard to suppress because of randomness. At present, common methods adopting a single feature generally need to accumulate dozens or hundreds of frames to ensure detection accuracy, which cannot update flickering pixels frequently. However, with low detection frequency, the flickering pixels are easily miss detected. In this paper, we propose an on-board flickering pixel dynamic suppression method based on multi-feature fusion. The visual and motion features of flickering pixels are extracted from the result of IRPT detection and tracking. Then, the confidence of flickering pixel evaluation strategy and selection mechanism of flickering pixel are introduced to fuse the above features, which achieves accurate flickering pixel suppression using a dozen frames. The experimental results evaluated on the real image of four scenarios show that the blind pixel false detection rate of the proposed method is no more than 1.02%. Meanwhile, evaluated on the simulated image, the flickering pixel miss suppression rate is no more than 2.38%, and the flickering pixel false suppression rate is 0. The proposed method could be an addition to most other IRPT detection methods, which guarantees the near-real-time and reliability of on-board IRPT detection applications.

Keywords: flickering pixel suppression; infrared point target; multi-feature fusion

check for updates

Citation: Jia, L.; Rao, P.; Chen, X.; Qiu, S. On-Board Flickering Pixel Dynamic Suppression Method Based on Multi-Feature Fusion. *Appl. Sci.* **2022**, *12*, 198. https://doi.org/10.3390/app12010198

Academic Editors: Qizhi Xu, Jin Zheng and Feng Gao

Received: 1 December 2021
Accepted: 23 December 2021
Published: 25 December 2021

1. Introduction

Space-based infrared point target (IRPT) detection is one of the most important applications of the on-board intelligent information processing technology, which has been wildly used in both the civil and military fields. Blind pixel suppression is the essential preprocess for all infrared camera applications. Owing to the limitation of the material, craftwork, and working environment, some infrared pixels will experience permanent or temporary functional degradation, namely blind pixels. Especially, blind pixels with a visual feature similar to that of IRPT can easily lead to false detection of IRPT and obstruct the real-time performance of the IRPT detection algorithm. Indeed, threshold filter [1], coding mechanism [2], and support vector machine [3], etc., are proposed to eliminate false alarm sources and have obtained good results in the IRPT detection algorithm. However, these sophisticated methods consume most computing resources to eliminate more indistinguishable false alarm sources from background. These blind pixels have a catastrophic effect on on-board intelligent information processing if they are not properly suppressed. Therefore, an effective blind pixel suppression method is vitally needed for on-board IRPT detection application.

Generally, blind pixels are divided into constant blind pixels and flickering pixels [4,5]. The constant blind pixels which appear as loog-live over-bright or over-dark can be corrected by the existing method effectively. However, considering that the flickering pixels can oscillate randomly between several levels, flickering pixels suppression is a hard task. Nowadays, blind pixel suppression is divided into two steps: blind pixel detection and blind pixel compensation. The blind pixel detection method can be classified into two categories. The first category is the calibration-based method [6–8], which uses the blackbody to obtain uniform reference images and detect blind pixels by comparing each pixels' response rate, deviation factor, noise statistics, and other properties. This method is easy to implement and has a good detection effect for constant blind pixels. However, flickering pixels cannot be detected accurately by this method due to their randomicity. More importantly, this method requires additional hardware equipment, such as blackbody and electromechanical parts, which will consume valuable hardware resources of satellites.

The other category is the scene-based blind pixel detection method, which utilizes image processing and computer vision to statistically analyze the scene image. Although this method has a lower accuracy compared to the calibration-based method, it can operate during the system's normal mission and does not require other equipment. Namely, this method has better real-time performance and adaptability. Presently, one of the most important tools for scene-based blind pixel detection methods is image registration. Korchev et al. [9] proposed a method that created bad pixel masks by frame registration using simple pixel differences, and produced good results for detecting moving, low-contrast targets at the same time. Liu et al. [10] adopted a registration-based method that could correct the fixed pattern noise with less error using the estimation of scene motion. In addition, Liu et al. [11] used the correlated phase information to make the registration more precise and adopted linear mapping to estimate true pixel value. However, the efficiency of image registration methods may influence the real-time performance of these methods. For instance, the registration based on the Fourier transform would make some methods difficult to apply to real-time hardware [11]. In addition, a good image registration effect asks for rich scene details, which limits the application for some scenarios. Recently, analyzing the intensity of adjacent pixels is also efficient. Tchendjou and SIMEU [12] proposed real-time and online defective pixel detection and correction algorithms. By using basic arithmetic operations on adjacent pixels, these algorithms have been implemented on an FPGA platform. Additionally, Cao et al. [13] proposed a scene-based bad pixel dynamic correction method. According to the self-adaptive median filter and the keyframe technique, this method can search for blind pixels with high detection accuracy. Song et al. [14] also gave an adaptive median filtering method to detect real-time blind and flickering pixels, in which a human-vision-based algorithm is applied to avoid misjudging blind pixels. However, these methods also have limitations. For instance, they detect blind pixels using high spatial frequency, which can easily cause the IRPT to be falsely identified as blind pixels. Therefore, these methods need dozens or hundreds of frames to ensure the accuracy of blind pixel detection, which may cause miss detection of the flickering pixel with low flickering frequency.

The blind pixel compensation method is applied to overcome the bad effect of the blind pixel which is detected by the previous step. At present, the existing blind pixel compensation methods [15–18] have achieved great compensation effect. These methods use adversarial networks, dual-band information, or regression model to compensate the blind pixel, and the rich scene details are preserved. Despite their success in estimated accuracy, aforementioned methods also have a limitation, since they generally treat each blind pixel as the constant blind pixel. However, the flickering pixels still retain a part of the information acquisition ability [19]. Therefore, how to choose the flickering pixel that should be compensated is worth studying.

In this paper, in order to address the problems mentioned above, we proposed an on-board flickering pixel dynamic suppression method based on multi-feature fusion. The visual and motion features of flickering pixels were extracted from the result of IRPT detection and tracking, which economizes computing resources. Meanwhile, the above features are fused by the confidence of flickering pixel (CFP) evaluation strategy and selection mechanism to guarantee the accuracy of flickering pixel detection with a dozen frames. Finally, CFP and visual features of flickering pixels are combined to compensate flickering pixels dynamically and accurately. This method overcomes the drawbacks associated with the conventional scene-based method that depends on a single feature consuming large amounts of frames to ensure detection accuracy.

2. Methodology

As shown in Figure 1, the proposed method is embedded in a IRPT detection process and, the block diagram briefly summarizes the processing steps of the proposed method including visual feature extraction, motion feature extraction, multi-feature fusion, and blind pixel compensation.

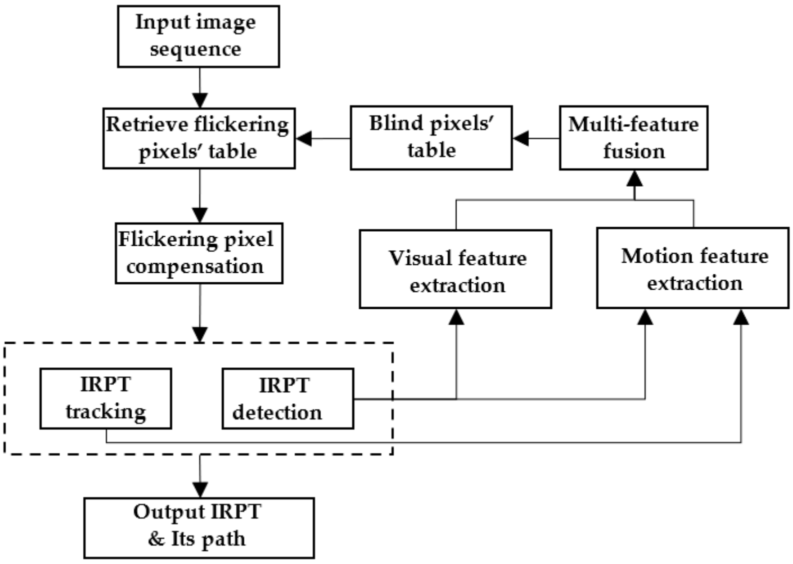

Input image sequence
Retrieve flickering pixels' table
Blind pixels' table
Multi-feature fusion
Flickering pixel compensation
Visual feature extraction
Motion feature extraction
IRPT tracking
IRPT detection
Output IRPT & Its path

Figure 1. The block diagram of proposed algorithm.

2.1. *Visual Feature Extraction Based on Facet Model*

A point target is a target whose actual image area on the surface of the detector is smaller than one pixel of the detector [20]. However, the imaging process of the practical optical system is influenced by diffraction and aberration. Therefore, IRPT imaging is not an ideal point but a speckle. Conversely, blind pixels are mainly caused by the limitations of the sensor, and their image is entirely unrelated to the optical system. Hence, blind pixels appear as bright or dark points in the infrared focal plane array (IRFPA).

From the above results and analysis, we propose a visual feature extraction method based on the facet model using the result of IRPT detection as input to distinguish blind pixels from suspected IRPT. The facet model was proposed by Robert M. Haralick [21]. This model considers the discrete pixel response as surface, and a polynomial function is used to estimate the surface in a small neighborhood [22]. Therefore, the spatial domain of the image can be partitioned into connected smooth regions called facets. Obviously, the blind pixels cannot be estimated by the facet model accurately because of its pulse shape. On the contrary, the intensity of IRPT decreased gradually from the center to the edge, which could be estimated more accurately by using the facet model. Therefore, we regard the detected IRPTs as potential blind pixels, and estimate them based on the facet model. Then, the estimation error of the facet model is segmented by the threshold to extract blind pixels. A detailed description of the steps is provided bellow.

Firstly, we use a bivariate cubic function f in canonical form as Equation (1) to fit the 3 × 3 neighborhood.

$$f(r,c) = \sum_{i=1}^{8} k_i p_i, \tag{1}$$

where, $(r,c) \in R \times C$, $R,C = \{-1,0,1\}$ are symmetric index sets of row and column respectively. $p_i \in \{1, r, c, r^2 - 2/3, rc, c^2 - 2/3, r(c^2 - 2/3), c(r^2 - 2/3)\}$ are 2-D discrete orthogonal Chebyshev polynomials, and k_i is the fitting coefficient which could be deduced by the least-squares algorithm as shown in Equation (2).

$$k_i = \frac{\sum_{(r,c)\in R\times C} P_i(r,c)I(r,c)}{\sum_{(r,c)\in R\times C} {P_i}^2(r,c)}, \tag{2}$$

where $I(r,c)$ is the response intensity of pixel (r,c). Obviously, $I(r,c)$ is independent of the remainder in Equation (2). Therefore, we can define this remainder as fixed filters ω_i in Equation (3) so that k_i could be calculated by convolution.

$$\omega_i = \frac{P_i(r,c)}{\sum_{(r,c)\in R\times C} {P_i}^2(r,c)}, \tag{3}$$

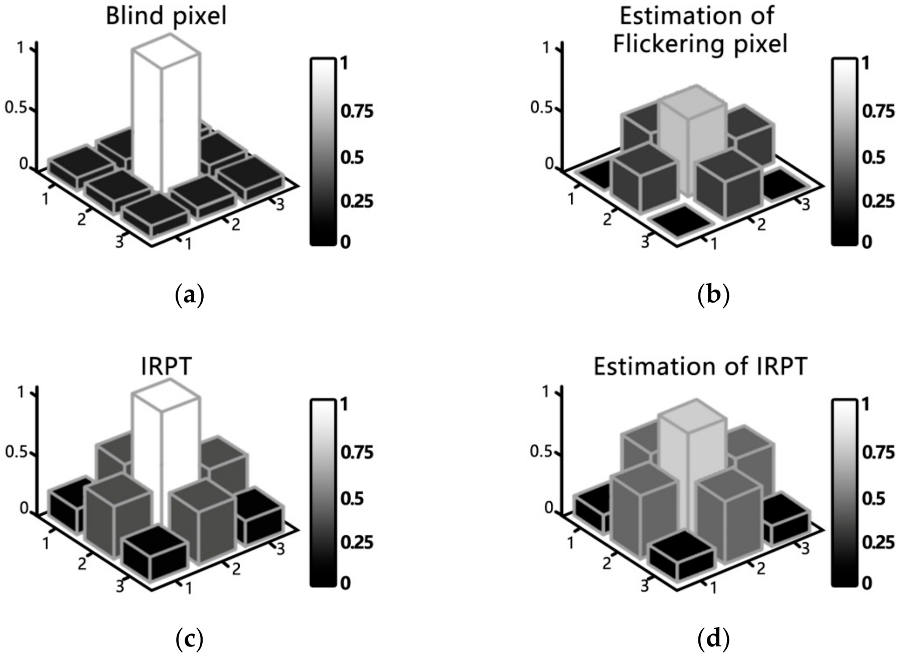
Then, the estimation of the facet model can be deduced through Equations (1) and (2) as follows:

$$f_{facet}(i,j) = k_1 - \frac{2}{3}k_4 - \frac{2}{3}k_6, \tag{4}$$

where (i,j) is the coordinate of potential blind pixels, and $f_{facet}(i,j)$ is the estimation of the facet model. The non-zero fitting coefficients could be calculated by the Equation (3) as follows:

$$\begin{cases} \omega_1 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \\ \omega_4 = \frac{1}{6}\begin{bmatrix} 1 & 1 & 1 \\ -2 & -2 & -2 \\ 1 & 1 & 1 \end{bmatrix} \\ \omega_6 = {\omega_4}^T \end{cases}, \tag{5}$$

Up to now, we could estimate the response intensity of the 3 × 3 neighborhood of potential blind pixels quickly. Figure 2 shows simulated blind pixel and IRPT and their estimation by the above-mentioned facet model. It can be clearly seen that the estimation of IRPT is more accurate.



(a)

(b)

(c)

(d)

Figure 2. The estimation of facet model (**a**) simulated blind pixel, (**b**) estimation of blind pixel by facet model, (**c**) simulated IRPT, (**d**) estimation of IRPT by facet model.

Subsequently, the estimation error of the facet model could be determined based on the following criterions as

$$Dev(i,j) = \frac{\sum_{m=-1}^{1}\sum_{n=-1}^{1}\sqrt{\left[I(m+i,n+j) - f_{facet}(m+i,n+j)\right]^2}}{\sum_{m=-1}^{1}\sum_{n=-1}^{1}\sqrt{\left[I(m+i,n+j) - mean(pic)\right]^2}}, \tag{6}$$

where $Dev(i,j)$ is the estimation error of the facet model while $mean(pic)$ represents the mean of the current image. Note that the denominator of Equation (6) is used in the normalization of the estimation error for easy threshold comparison.

Finally, we set up a visual feature model of blind pixels (VFBP model) and define $\mathrm{Th}_{\mathrm{IM}}$ as the threshold of estimation error. The results of matching VFBP model are expressed by $X_{IM}(i,j)$, which is written as

$$X_{IM}(i,j) = \begin{cases} 1, & Dev(i,j) < \mathrm{Th}_{\mathrm{IM}} \\ 0, & Dev(i,j) \geq \mathrm{Th}_{\mathrm{IM}} \end{cases}, \tag{7}$$

If $Dev(i,j)$ is smaller than $\mathrm{Th}_{\mathrm{IM}}$, $X_{IM}(i,j)$ will be assigned to 0, which means this potential blind pixel might be an IRPT. On the contrary, $X_{IM}(i,j)$ will be set to 1, which means this pixel might be a blind pixel. In this paper, $\mathrm{Th}_{\mathrm{IM}}$ is set to 0.30. According to the change of practical optical system, this threshold needs to be adjusted.

2.2. Motion Feature Extraction

In this paper, the IRPTs are divided into two categories: space target and aircraft. The movement of the space target is restricted to a specific orbit. Meanwhile, the movement of aircraft is also constrained by specific flight envelop [23]. Therefore, the moving path of IRPT can be treated as smooth curvilinear motion within a short time. In terms of blind pixel, although blind pixels cannot move, the inter-frame correlation may cause an error path formed by adjacent blind pixels. Therefore, the blind pixels appear as static points or winding paths.

According to the above analysis, we set up two kinds of motion feature models of blind pixels (MFBP model) to extract the motion feature of blind pixels. The static MFBP model with IRPT detection results as input is used to detect isolated blind pixels. Meanwhile, the dynamic MFBP model with IRPT tracking results as input detects adjacent blind pixels by evaluating the curvature degree of the path.

The static MFBP model firstly calculates the frequency that a pixel is detected as the blind pixel. Then, compare this frequency with the threshold to detect isolated blind pixels, as shown in Equation (8).

$$X_{MOst}(i,j) = \begin{cases} 0, & \sum_{f=F-F_{length}+1}^{F} X_d(f,i,j)/F_{length} < \mathrm{Th}_{MO\mathrm{sst}} \\ 1, & \sum_{f=F-F_{length}+1}^{F} X_d(f,i,j)/F_{length} \geq \mathrm{Th}_{MO\mathrm{sst}} \end{cases}, \tag{8}$$

where $X_d(f,i,j)$ is the result of IRPT detection. If $X_d(f,i,j)$ is 1, it means that pixel (i,j) is detected as potential blind pixel in fth frame. F presents current frame number while F_{length} presents the frame range of testing. Finally, $X_{MOst}(i,j)$ is the result of static MFBP model matching. If $X_{MOst}(i,j)$ equals to 1, it means that pixel (i,j) may be blind pixel. Conversely, if $X_{MOst}(i,j)$ equals to 0, this pixel may be normal. In terms of parameter selection, the threshold of static MFBP model ($\mathrm{Th}_{MO\mathrm{sst}}$) is set to 0.3. However, this threshold needs to be adjusted according to frame frequency and moving state of IRPT. It is necessary to point out that, considering computational complexity and reliability, F_{length} is set to 10 to 20 in this paper.

Dynamic FPMF model also evaluates within F_{length} frames and takes the path of IRPT tracking as input. Suppose that (i_{fn}, j_{fn}) denotes the path points forms a certain path, and the subscript fn means the frame number. Using the result of IRPT tracking as input, pick up direction vectors from the path, we now describe the procedure as follows. Firstly, traverse all path points in frame order as the starting point of each direction vector. Then, traverse the path points whose distance to the starting point is greater than $\sqrt{2}$ and the frame number is greater than the starting point, and choose the path point with the smallest frame number among them as the end point of the corresponding direction vector. Let $\overrightarrow{D_n}$ be direction vectors and the subscript n means order number of direction vectors. Finally, the threshold of the dynamic MFBP model ($\mathrm{Th}_{MO\mathrm{dy}}$) was introduced to evaluate the winding degree of a path, as shown in Equation (9).

$$X_{MOdy}(i_{fn}, j_{fn}) = \begin{cases} 0, & \frac{1}{n-1}\sum_{j=1}^{n-1} \frac{\overrightarrow{D_{j+1}}\cdot\overrightarrow{D_j}}{\left|\overrightarrow{D_{j+1}}\right|\cdot\left|\overrightarrow{D_j}\right|} < \mathrm{Th}_{MO\mathrm{dy}} \\ 1, & \frac{1}{n-1}\sum_{j=1}^{n-1} \frac{\overrightarrow{D_{j+1}}\cdot\overrightarrow{D_j}}{\left|\overrightarrow{D_{j+1}}\right|\cdot\left|\overrightarrow{D_j}\right|} \geq \mathrm{Th}_{MO\mathrm{dy}} \ or \ n \leq 3 \end{cases}, \tag{9}$$

where $X_{MOdy}(i_{fn}, j_{fn})$ presents the matching result of the dynamic MFBP model. The left side of the inequality denotes the mean value of the angle transformation of the path direction vector. If this value is larger than or equal to $\mathrm{Th}_{MO\mathrm{dy}}$, $X_{MOdy}(i_{fn}, j_{fn})$ will be set to 1, which means this path is tortuous and these path points may be blind pixels. Furthermore, if the number of available direction vectors (n) is less than or equal to 3, $X_{MOdy}(i_{fn}, j_{fn})$ will be set to 1 also, which means the pixels forming this path are too close and they might be blind pixel blocks. Note that $\mathrm{Th}_{MO\mathrm{sdy}}$ is set to 30. This threshold also needs to be adjusted according to frame frequency and moving state of IRPT.

2.3. The Strategy of Flicking Pixel Suppression

The confidence evaluation mechanism based on multi-features fusion was introduced to evaluate the blinking frequency of a pixel. We use $C(i, j)$ to represent the confidence of flickering pixel of pixel (i, j). The value range is 0 to 1. The greater the confidence of flickering pixel of a pixel is, the higher the blinking frequency would be. In addition, the step size of updating confidence is defined as c. Considering the iteration speed and reliability of the confidence, we set c ranging from 0.02 to 0.05. The higher the c, the faster the confidence update, but the lower the reliability.

All the pixels' confidences are assigned to 0 at the beginning. If a certain pixel is judged as a flickering pixel by the VFBP model and one of the MFBP models at the same time, the confidence of this pixel will be increased a unit c, as shown in Equation (10).

$$C(i,j) = \begin{cases} 1 & C(i,j) + c > 1 \\ C(i,j) + c & C(i,j) + c \leq 1 \end{cases} \ if(X_{IM}(i,j) = 1)\&\&\left(X_{MOst}(i,j) = 1\middle|\middle|X_{MOdy}(i,j) = 1\right), \tag{10}$$

If a certain pixel is judged as normal pixel by VFBP model or MFBP model, the confidence of this pixel will be decreased a unit c, as shown in Equation (11).

$$C(i,j) = \begin{cases} 0 & C(i,j) - c \leq 0 \\ C(i,j) - c & C(i,j) - c > 0 \end{cases} \ if(X_{IM}(i,j) = 0)\left|\middle|\left(X_{MOst}(i,j) = 0\&\&X_{MOdy}(i,j) = 0\right)\right., \tag{11}$$

Especially, a threshold was introduced to maintain the scale of flickering pixels' table. $\mathrm{Th}_{\mathrm{re}}$ is the reserve threshold and is set to 0.5 c. When CFP of a pixel is greater than $\mathrm{Th}_{\mathrm{re}}$, the pixel will be added to the flickering pixels' table. Otherwise, it will be removed from the table. In addition, due to changes in working conditions, some flickering pixels will return to normal. Therefore, we introduce forgetting factor (F) with values from 0.96 to 0.98. If a certain pixel within the flickering pixels' table is not updated CFP in the current frame, the CFP of this pixel will be decreased by multiplying the forgetting factor. The block diagram of proposed confidence evaluation mechanism of flickering pixel is shown in Figure 3.

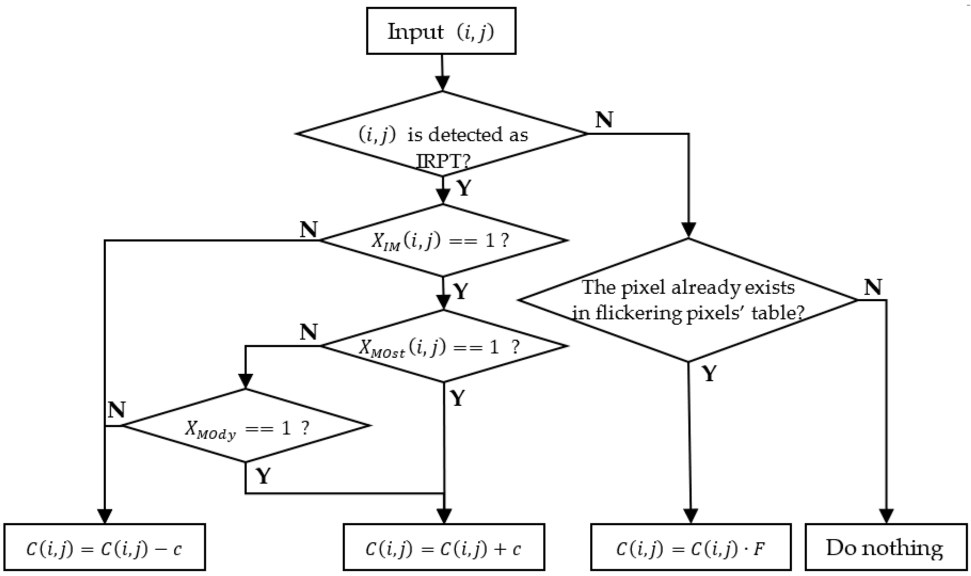
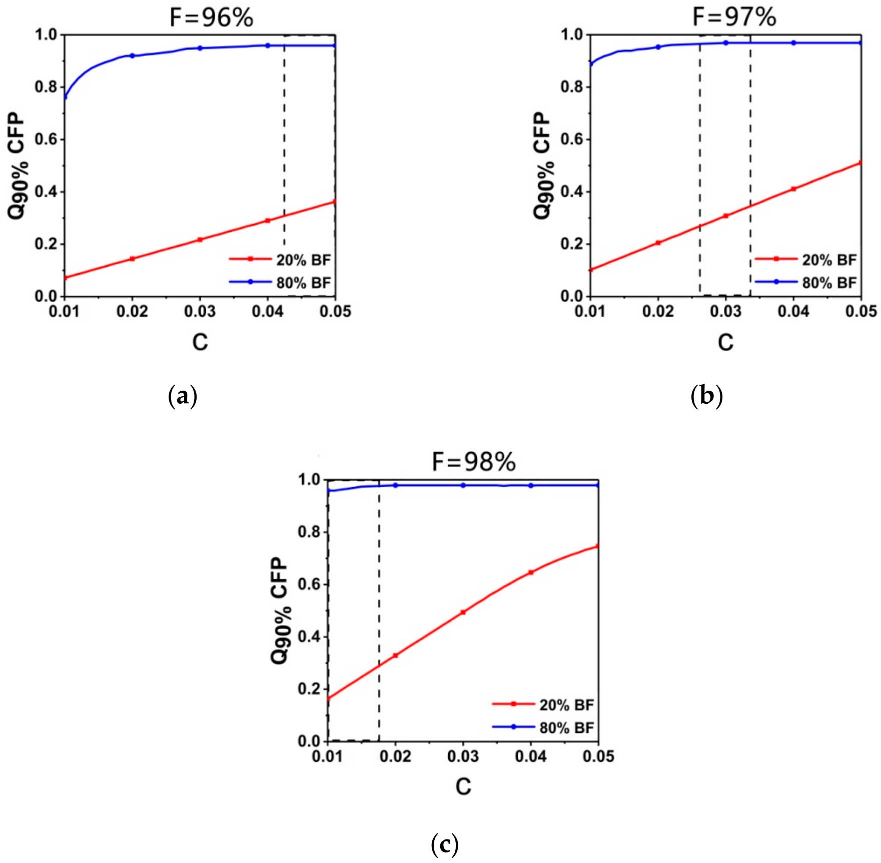


Figure 3. Confidence evaluation mechanism of flickering pixel.

Subsequently, the dynamic compensation strategy based on confidence was proposed to compensate the flickering pixel accurately to frame. The direct compensation threshold $\mathrm{Th_D}$ and the compensation after validation threshold $\mathrm{Th_V}$ are introduced. If the CFP of a certain pixel is equal to or greater than $\mathrm{Th_D}$, this pixel will be compensated directly. Correspondingly, if the confidence of a certain pixel is equal to or greater than $\mathrm{Th_V}$ but less than $\mathrm{Th_D}$, this pixel will be validated by the VFBP model again. If it is judged as flickering pixel by the VFBP model in current frame, this pixel will be compensated. Otherwise, this pixel will not be compensated. In this paper, for saving computing resources, flickering pixels would be compensated by the mean filter or median filter. Alternatively, the flickering pixels are simply suppressed by skipping the IRPT detection process for these pixels.

Finally, the above parameters and thresholds selection are discussed. Indeed, these values are important because they determine the effect of flickering pixel suppression. The CFP fluctuates nonlinearly with the forgetting factor and steps size of updating CFP. Therefore, it is hard to give a clear interpretation of fluctuations of CFP. Moreover, the relation between blinking frequency of flickering pixel and the corresponding evolution of CFP should be determined to select the thresholds $\mathrm{Th_V}$ and $\mathrm{Th_D}$. Firstly, we introduce the 90th-percentile CFP ($\mathrm{Q_{90\%CFP}}$) to determine thresholds. The probability that the CFP of a pixel with a certain blinking frequency is greater than $\mathrm{Q_{90\%CFP}}$ is 90%. Then, we define that the flickering pixel with blinking frequency equal to or greater than 80% will be suppressed directly, which corresponds to $\mathrm{Th_D}$. The blinking frequency corresponding to $\mathrm{Th_V}$ is 20%. Subsequently, we use uniformly distributed random numbers to simulate the blinking frequency of flickering pixel. Finally, through ten million iterations, the distribution of $\mathrm{Q_{90\%CFP}}$ with different blinking frequency (BF), forgetting factor (F) and step size of updating CFP (c) is observed, as shown in Figure 4.

In this paper, the selection principle of parameters and thresholds is to make the blue line ($\mathrm{Th_D}$) and the red line ($\mathrm{Th_V}$) showed in Figure 4 uniformly distributed within the range of CFP. For instance, when the forgetting factor is equal to 0.98 and the step size of updating CFP is equal to 0.05, there will be a huge space under the red line ($\mathrm{Th_V}$), as shown in Figure 4c. It shows that flickering pixel needs more frames to accumulate CFP. Correspondingly, when forgetting factor is equal to 0.96 and step size of updating CFP is equal to 0.01, there will be a relatively huge space above the blue line ($\mathrm{Th_D}$), as is shown in Figure 4a. It denotes the flickering pixel needs more frames to weaken CFP when returning normal briefly. Following these principles, some recommended ranges of parameters and thresholds are selected and highlighted in the dotted box in Figure 4. Meanwhile, some typical selection of parameters and thresholds are listed in Table 1.



(**a**) (**b**)

(**c**)

Figure 4. The distribution of $Q_{90\%CFP}$ with different blinking frequency (BF) and step size of updating CFP (**a**) forgetting factor = 0.96, (**b**) forgetting factor = 0.97, (**c**) forgetting factor = 0.98.

Table 1. Typical selection of parameters and thresholds.

Step Size of Updating CFP (*c*)	0.01	0.03	0.05
Forgetting factor	0.98	0.97	0.96
Th_V	0.17	0.30	0.37
Th_D	0.95	0.97	0.96
Th_{re}		0.5 *c*	

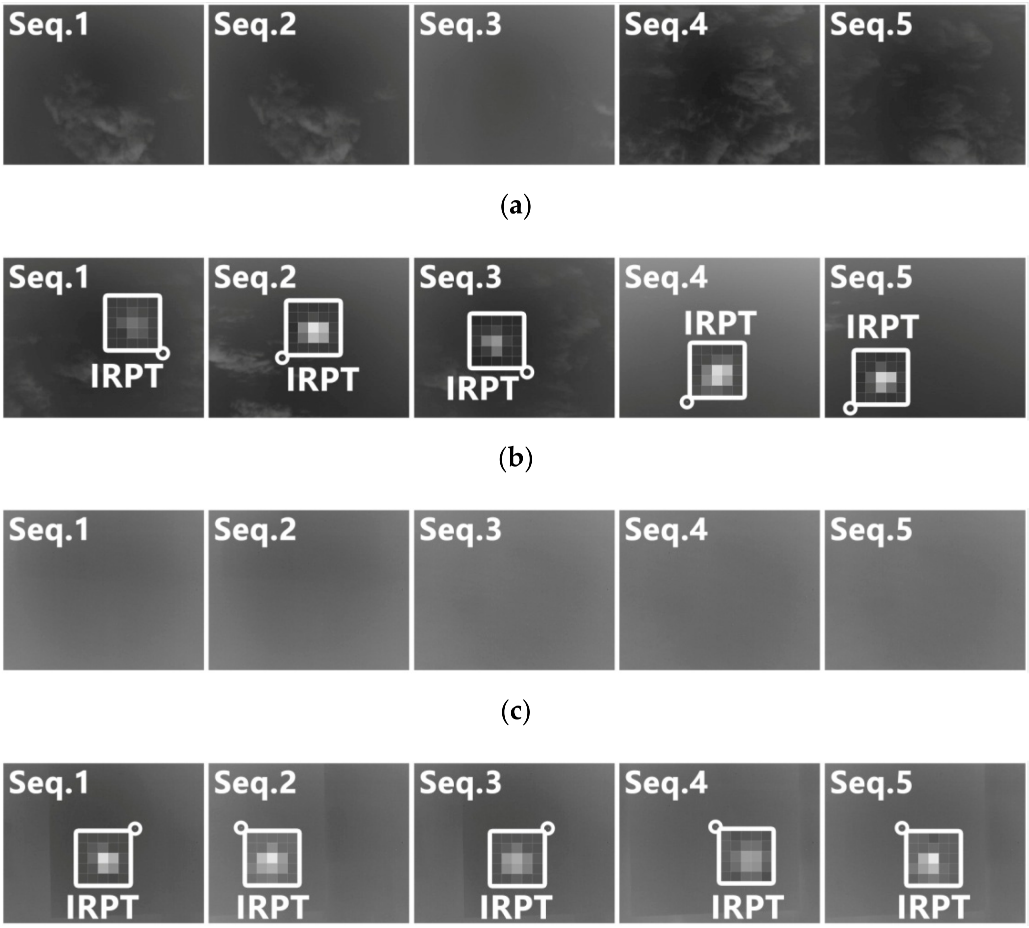
3. Experimental Results and Discussion

3.1. Experimental Data

In order to measure the performance of the proposed method in this paper, we used MWIR 640 × 512 pixels Integrated Detector Dewar Cooler Assembly to obtain experimental data. Table 2 provides an overview of its specifications. In this letter, IRPT we mainly study is space target or aircraft. Therefore, we obtained sky background and deep space background with their IRPT as experimental data, as shown in Figure 5. In Figure 5, the scenarios from the top row to bottom row are sky background, sky background with IRPT, deep space background, and deep space background with IRPT, respectively. Each data set has five sequences, and each sequence includes 150–200 frames. Figure 6 shows the details about the acquisition of experimental data. In these data sets, the sky background and its IRPT are the real shots of the sky and the aircraft as shown in Figure 6a; However, the deep space background and its IRPT are simulated by blackbody and target board as shown in Figure 6b. In Figure 6c, the blackbody penetrates the target hole to form an IRPT. Then, we can control the field of view movement in a certain direction by moving the camera. In the image, the IRPT will move in the opposite direction of the camera movement.

Table 2. Specifications of the MWIR detector used to obtain experimental data.

Format	640 × 512
Pixel size	15 µm
Spectral Range	3–5 µm
F-number	4
Noise Equivalent Temperature Difference (NETD)	30 mk
Framerate	50 Hz
Bits per pixel	14 bits



(**a**)

(**b**)

(**c**)

(**d**)

Figure 5. The real experimental data (**a**) sky background, (**b**) sky background with IRPT, (**c**) space background, (**d**) space background with IRPT.

Due to the limited number of flickering pixels of real experimental data, the simulated flickering pixels with random blinking frequency were added to the real experimental data to make a quantitative analysis on the effect of flickering pixel suppression.

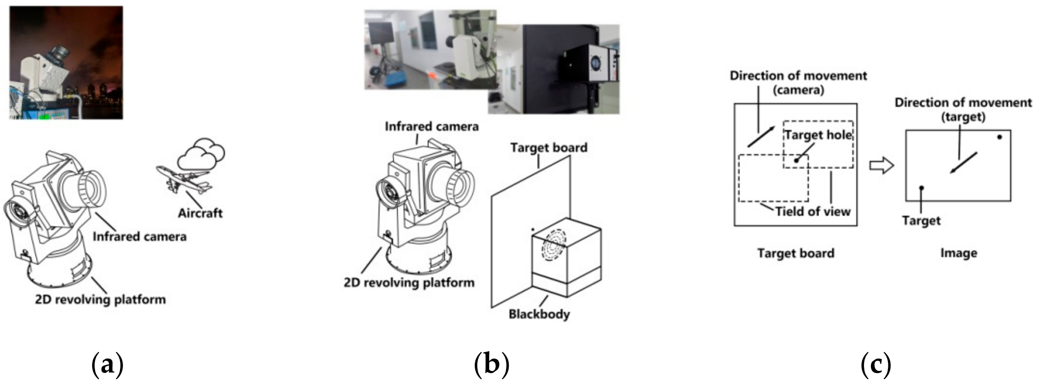


Figure 6. Experimental data acquisition details (**a**) gathering method of the sky background, (**b**) semi-physical simulation method of deep space background, (**c**) the detail of how the IRPT moves in deep space background of semi-physical simulation.

3.2. Evaluation Criterion

In this paper, the evaluation criteria consist of two parts, blind pixel detection evaluation criterion and flickering pixel suppression evaluation criterion.

We adopt the number of detected blind pixels, the false detection rate of detected blind pixels, and the average running time per frame to evaluate the effect of blind pixel detection. The number of detected blind pixels (N_{bp}) denotes the number of constant blind pixels and flickering pixels detected by a certain method. The false detection rate of detected blind pixels is given by Equation (12).

$$R_{fbp} = \frac{N_{fbp}}{N_{bp}} \cdot 100\%, \tag{12}$$

where N_{fbp} is the number of normal pixels that are falsely detected as blind pixels by a method. In this paper, we obtain calibration data, and N_{fbp} can be measured by the method given in national standard [24]. Finally, the average running time per frame (T_{pf}) can present the operating efficiency of a method. The proposed method and other methods are implemented under MATLAB R2018a with an Intel Core 2.80 GHz processor and 8 GB of physical memory.

The number of suppressed flickering pixels, the miss suppression rate of flickering pixels, and the false suppression rate of flickering pixels form the evaluation criterion for flickering pixel suppression. The number of suppressed flickering pixels (N_{sf}) means the total number of frames suppressed for all flickering pixels. The miss suppression rate of flickering pixel (R_{msf}) is defined in Equation (13).

$$R_{msf} = \frac{\sum_{n=1}^{N} N_{msf}(n)}{\sum_{n=1}^{N} N_{rf}(n)} \cdot 100\%, \tag{13}$$

where N is the flickering pixel amount, and $N_{msf}(n)$ refers to the frame amount that should be suppressed but not be suppressed of the n th flickering pixels. $N_{rf}(n)$ is the frame amount of blinking of n th flickering pixel. Similarly, false suppression rate of flickering pixel (R_{fsf}) is given by Equation (14).

$$R_{fsf} = \frac{\sum_{n=1}^{N} N_{fsf}(n)}{\sum_{n=1}^{N} N_{rf}(n)} \cdot 100\%, \tag{14}$$

where $N_{fsf}(n)$ refers to the frame amount that false suppressed of the n th flickering pixel. As is mentioned above, the evaluation of flickering pixel suppression uses simulation data.

Therefore, the $N_{rf}(n)$, $N_{fsf}(n)$, and $N_{msf}(n)$ in Equations (13) and (14) can be recorded during simulation.

3.3. Results and Discussion

According to the experimental data and evaluation criterion above, the proposed method is compared with the scene-based bad pixel dynamic correction (SBBPDC) method [13] and the calibration-based adaptive threshold bad pixel detection (CBATBPD) method [9]. In this experiment, we used the median filter and threshold segmentation to detect IRPT and use the interframe matching to track IRPT, in which our proposed method was embedded. The above IRPT process methods are one of the most common and the simplest methods in IRPT application. It is necessary to point out that our proposed method can also be embedded in any IRPT detection and tracking method.

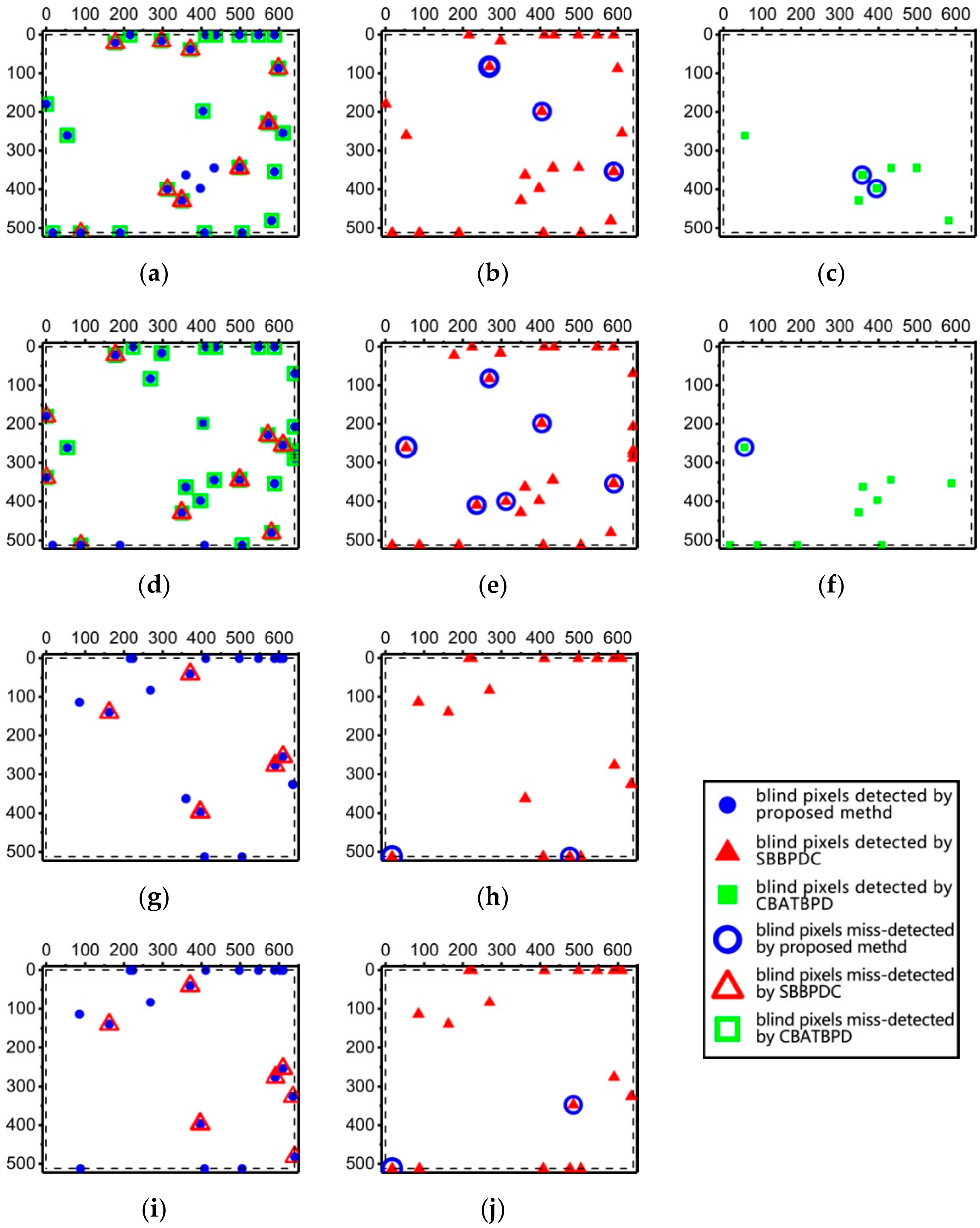
Figure 7 shows the blind pixel maps of the first sequence of four scenarios, which are detected by SBBPDC, CBATBPD, and our proposed method. The top row to bottom row of Figure 7 is sky background, sky background with IRPT, deep space background, and deep space background with IRPT, respectively. The left column to right column of Figure 7 is the proposed method, SBBPDC, and CBATBPD, respectively. The blue points denote the blind pixels detected by our method. The red triangular rims and green square rims represent the blind pixels that are not detected by SBBPDC and CBATBPD, respectively. It is clear in the left column of Figure 7 that our method has detected many blind pixels that are not detected by other methods. Similarly, the red triangles and green squares denote the blind pixels detected by SBBPDC and CBATBPD, respectively, and the blue round rims are the blind pixels that are miss detected by our method. Obviously, there are only a few blind pixels that cannot be detected by the proposed method compared with other methods. As the middle column of Figure 7 shown, the SBBPDC cannot detect some adjacent blind pixels compared with our proposed method. This is because the adaptive threshold segmentation of this method erroneously suppresses a part of adjacent blind pixels. Furthermore, due to the movement of the background, some edges of the image usually move out quickly and do not have the same part for inter-frame registration. Therefore, the CBATBPD cannot detect the blind pixels at some edges of the field of view as shown in Figure 7. In conclusion, our method adopts multi-feature to accurately identify the real targets from potential targets. It can detect more real blind pixels compared with other methods. The blind pixel maps of other sequences have similar results as the first sequence. However, space does not permit their presentation.

In Figure 8, we respectively plotted the number of blind pixels detected by the above-mentioned methods in each frame. Figure 8a–d illustrates the data of the first sequence of sky background, sky background with IRPT, deep space background, and deep space background with IRPT, respectively. It is obvious that the proposed method can detect more blind pixels with fewer frames compared with other methods. SBBPDC uses the keyframe to improve the detection accuracy and efficiency, which depends on the change of background. Therefore, the detection rate of SBBPDC tends to slow down in deep space because the change of background is not obvious. Additionally, CBATBPD needs hundreds of frames to ensure detection accuracy.

Subsequently, we calculate the N_{bp}, R_{fbp}, and T_{pf} mentioned in Section 3.2 of the above three methods in Table 3. The false detection rate of the proposed method is less than 1.02% in different scenarios, while the average running time per frame of the proposed method is less than 0.46 s. The proposed method could detect more blind pixels with less false detection rate and less time. It can be clearly seen that the proposed method is superior to others in four application scenarios.

Table 3. The situation of blind pixels detected in four application scenarios.

Blind Pixel Detection Method	**Application Scenarios**											
	Cloud			**Cloud with Target**			**Deep Space**			**Deep Space with Target**		
	N_{bp}	R_{fbp} **(%)**	T_{pf} **(s)**	N_{bp}	R_{fbp} **(%)**	T_{pf} **(s)**	N_{bp}	R_{fbp} **(%)**	T_{pf} **(s)**	N_{bp}	R_{fbp} **(%)**	T_{pf} **(s)**
SBBPDC	62	1.61	0.71	69	1.45	0.99	37	0	0.71	32	0	0.89
CBATBPD	37	10.8	1.17	34	8.82	1.02	—	—	—	—	—	—
Proposed	**98**	**1.02**	**0.36**	**100**	**1.00**	**0.46**	**40**	**0**	**0.34**	**38**	**0**	**0.34**



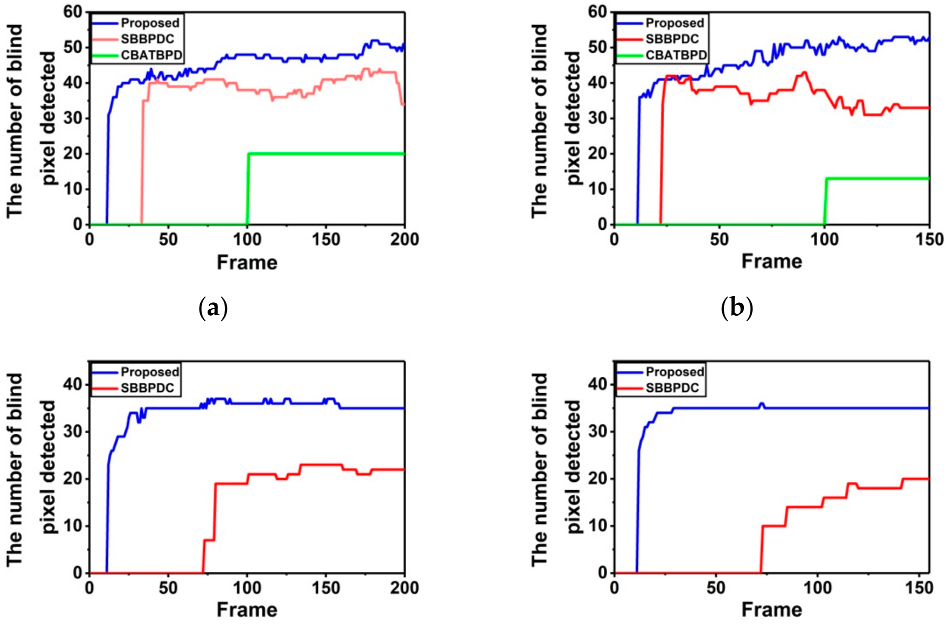
(**a**) (**b**) (**c**)

(**d**) (**e**) (**f**)

(**g**) (**h**)

(**i**) (**j**)

Figure 7. The blind pixel maps of the first sequence of each scenario detected by different methods (**a**) Background: sky Method: the proposed method; (**b**) Background: sky Method: SBBPDC; (**c**) Background: sky Method: CBATBPD; (**d**) Background: sky with IRPT Method: the proposed method; (**e**) Background: sky with IRPT Method: SBBPDC; (**f**) Background: sky with IRPT Method: CBATBPD; (**g**) Background: deep space Method: the proposed method; (**h**) Background: deep space Method: SBBPDC; (**i**) Background: deep space with IRPT Method: the proposed method; (**j**) Background: deep space with IRPT Method: SBBPDC.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. The number of blind pixels detected by different methods in each frame (**a**) sky background; (**b**) sky background with IRPT; (**c**) deep space background; (**d**) deep space background with IRPT.

In order to discuss the time consumption of our proposed method in practical applications, we recorded the average frame time consumption of each step in different scenarios in Table 4. The average frame time consumption of blind pixels detection and compensation is about 0.09 s. After code optimization and parallel acceleration, the time-consuming situation of our method can be further improved. Furthermore, in practical applications, our method can work once every two to three frames to ensure the near-real-time suppression of blind pixels. Moreover, it is important to note, owing to the limitation of the transmission bandwidth and computing resource, the current frame rate of IRPT detection and tracking application usually only reaches 5~20 frames per second.

Table 4. The average frame time consumption of each step of our method used in the above experiment in different scenarios.

Scenarios **Time Consumption**	**Cloud**	**Cloud with Target**	**Deep Space**	**Deep Space with Target**
IRPT detection (Median filter and threshold segmentation)	0.111 s	0.121 s	0.109 s	0.112 s
IRPT tracking (Interframe matching)	0.156 s	0.247 s	0.135 s	0.141 s
Blind pixels detection and compensation	0.091 s	0.089 s	0.091 s	0.090 s
Total	0.358 s	0.457 s	0.335 s	0.343 s

In the terms of flickering pixel suppression, our proposed method is to achieve frame-accurate dynamical flickering pixel suppression. For instance, the pixel (347,487) was detected as a flickering pixel in the second sequence of cloud scenario and Figure 9 shows the suppression situation of this pixel. The CFP of this pixel was less than Th_V before the 18th frame, so this pixel was not suppressed. After the 18th frame, our method accurately suppresses this pixel when the response is too low.

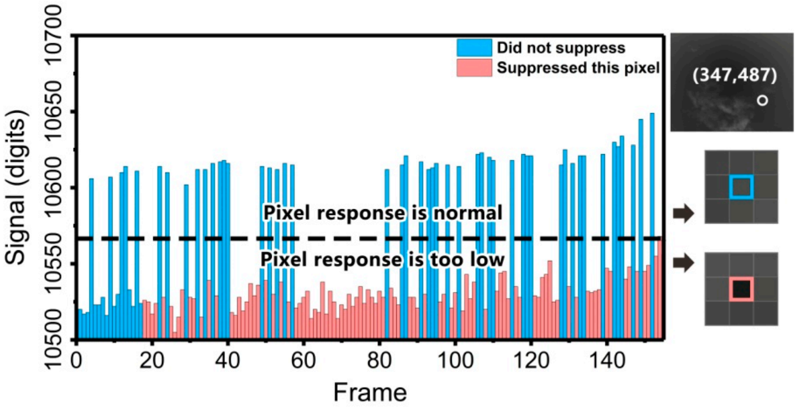


Figure 9. The suppression situation of pixel (347,487) in the second sequence of cloud scenario.

Meanwhile, we use the simulated data to make a quantitative analysis on the flickering pixel suppression. Figure 10 shows the randomly added flickering pixels and miss-suppressed flickering pixels of our method. Additionally, Figure 10a–d is sky background, sky background with IRPT, deep space background, and deep space background with IRPT, respectively. Blue points show the flickering pixels added randomly, red circles indicate the flickering pixels which miss-suppressed by our proposed method. We calculate the N_{sf}, R_{fbp}, and R_{fsf} mentioned in Section 3.2 of different simulated scenarios in Table 5. The results showed that there is no false suppression of flickering pixels in all scenarios. In addition, the miss suppression rate of flickering pixels is less than or equal to 2.38% in all scenarios.

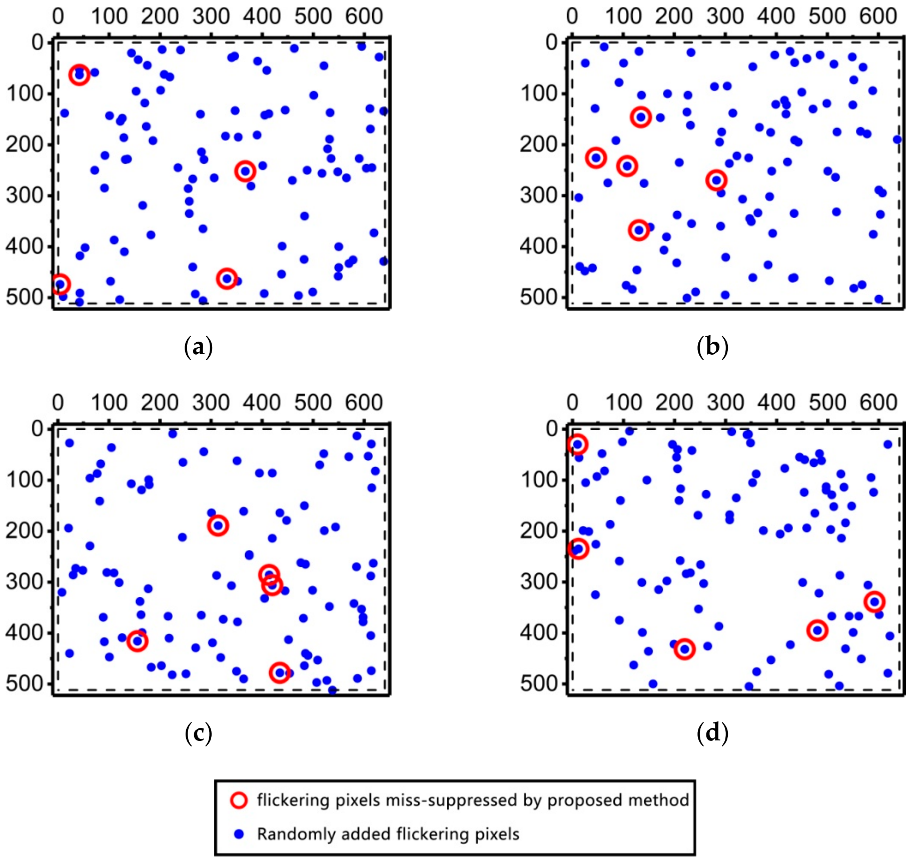


Figure 10. The randomly added flickering pixels and miss-suppressed flickering pixels of different scenarios (**a**) sky background; (**b**) sky background with IRPT; (**c**) deep space background; (**d**) deep space background with IRPT.

Table 5. The situation of flickering pixels suppressed by proposed method.

Application Scenarios	N_{sf}	R_{msf} (%)	R_{fsf} (%)
Cloud	10,080	1.77	0
Cloud with target	9500	2.07	0
Deep space	7331	2.38	0
Deep space with target	7519	2.07	0

4. Conclusions

In this paper, an on-board flickering pixel dynamic suppression method based on multi-feature fusion is proposed. This method is embedded in the IRPT detection process, and the results of IRPT detection and tracking are used to extract the visual and motion features of the flickering pixel. A small amount of extra computing resource was consumed. Therefore, our proposed method can achieve real-time uninterrupted blind pixel detection without interrupting IRPT detection. Furthermore, the confidence of the flickering pixel evaluation strategy and the selection mechanism of flickering pixel are introduced to fuse the above features, which can achieve frame-accurate dynamical flickering pixel suppression. Experimental results have shown that our proposed method is superior to others in blind pixel detection and flickering pixel suppression. The proposed method can be an addition to most other IRPT detection processes, and it has significant meaning that the ability to suppress blind pixels guarantees near-real-time IRPT detection.

Author Contributions: All the authors contributed to this study. Conceptualization, L.J. and P.R.; Investigation, L.J. and X.C.; Methodology, P.R. and X.C.; Software, L.J. and X.C.; Data curation, L.J.; Funding acquisition, P.R.; Project administration, X.C. and S.Q.; Supervision, S.Q. Writing—Original draft preparation, L.J.; Writing—review and editing, P.R. and S.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 62175251.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

References

1. Hai-Bin, P.; Wei, M.; Ming-Yu, C. Detection algorithm for space dim moving object. *Proc. SPIE* **2007**, *6595*. [CrossRef]
2. Zhao, F.; Wang, T.; Shao, S.; Zhang, E.; Lin, G. Infrared moving small-target detection via spatiotemporal consistency of trajectory points. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 122–126. [CrossRef]
3. Wan, M.; Ye, X.; Zhang, X.; Xu, Y.; Gu, G.; Chen, Q. Infrared small target tracking via gaussian curvature-based compressive convolution feature extraction. *IEEE Geosci. Remote Sens. Lett.* **2021**, 1–5. [CrossRef]
4. Gross, W.; Hierl, T.; Schulz, M. Correctability and long-term stability of infrared focal plane arrays. *Opt. Eng.* **1999**, *38*, 862–869.
5. Wang, E.; Jiang, P.; Hou, X.; Zhu, Y.; Peng, L. Infrared stripe correction algorithm based on wavelet analysis and gradient equalization. *Appl. Sci.* **2019**, *9*, 1993. [CrossRef]
6. Ribet-Mohamed, I.; Nghiem, J.; Caes, M.; Guenin, M.; Hoglund, L.; Costard, E.; Rodriguez, J.B.; Christol, P. Temporal stability and correctability of a MWIR T2SL focal plane array. *Infrared Phys. Technol.* **2019**, *96*, 145–150. [CrossRef]
7. Arounassalame, V.; Guenin, M.; Caes, M.; Hoglund, L.; Costard, E.; Christol, P.; Ribet-Mohamed, I. Robust evaluation of long-term stability of an InAs/GaSb type II superlattice midwave infrared focal plane array. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 1–8. [CrossRef]
8. Shi, Y.; Mao, H.C.; Zhang, T.X.; Cao, Z.G. New approach of IRFPA non-effective pixel discrimination based on pixel's characteristics histogram analysis. *J. Infrared. Millim. W* **2005**, *24*, 119–124.
9. Korchev, D.; Kwon, H.; Owechko, Y. Detecting small, low-contrast moving targets in infrared video produced by inconsistent sensor with bad pixels. *Opt. Eng.* **2015**, *54*, 113102. [CrossRef]
10. Liu, Z.; Ma, Y.; Huang, J.; Fan, F.; Ma, J.Y. A registration based nonuniformity correction algorithm for infrared line scanner. *Infrared Phys. Technol.* **2016**, *76*, 667–675. [CrossRef]
11. Liu, N.; Xie, J. Interframe phase-correlated registration scene-based nonuniformity correction technology. *Infrared Phys. Technol.* **2015**, *69*, 198–205. [CrossRef]

12. Tchendjou, G.T.; Simeu, E. Detection, location and concealment of defective pixels in image sensors. *IEEE Trans. Emerg. Top. Comput.* **2021**, *9*, 664–679. [CrossRef]
13. Liu, Y.C.W.J.C.L.X. Scene-based bad pixel dynamic correction and evaluation for IRFPA device. In Proceedings of the Advances in Optoelectronics and Micro/Nano-Optics, Guangzhou, China, 3–6 December 2010.
14. Zhang, Z.S.D.Z.S. Scene-based blind and flickering pixel dynamic correction algorithm. In Proceedings of the 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP), Chongqing, China, 11–13 December 2019.
15. Chen, S.T.; Meng, H.; Pei, T.; Zhang, Y.Y. An adaptive regression method for infrared blind-pixel compensation. *Infrared Phys. Technol.* **2017**, *85*, 443–449. [CrossRef]
16. Chen, S.T.; Jin, M.; Zhang, Y.Y.; Zhang, C. Infrared blind-pixel compensation algorithm based on generative adversarial networks and Poisson image blending. *Signal Image Video Process.* **2020**, *14*, 77–85. [CrossRef]
17. Nguyen, C.T.; Mould, N.; Regens, J.L. Dead pixel correction techniques for dual-band infrared imagery. *Infrared Phys. Technol.* **2015**, *71*, 227–235. [CrossRef]
18. Hailov, Y.P.A.S.A. Hardware implementation and verification of the sensor defective pixels correction algorithm. In Proceedings of the 2020 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), Saint-Petersburg, Russia, 1–5 June 2020.
19. Liu, G.R.; Sun, S.L.; Lin, C.Q.; Lyu, P.Y. Analysis and suppression method of flickering pixel noise in images of infrared linear detector. *J. Infrared Millim. Waves* **2018**, *37*, 421–426.
20. Cao, L.H.; Wan, C.M.; Zhang, Y.F.; Li, N. Infrared radiation characteristic measure method of point target. *J. Infrared Millim. Waves* **2015**, *34*, 460–464.
21. Haralick, R.M. Digital step edges from zero crossing of second directional derivatives. *IEEE Trans. Pattern Anal. Mach. Intell.* **1984**, *PAMI-6*, 58–68. [CrossRef] [PubMed]
22. Bai, X.Z.; Bi, Y.G. Derivative entropy-based contrast measure for infrared small-target detection. *IEEE Trans. Geosci. Remote* **2018**, *56*, 2452–2466. [CrossRef]
23. Chen, Q.; Qian, W.X.; Zhang, W. *Infrared Target Detection*; Publishing House of Electronics Industry Press: Beijing, China, 2016.
24. GB/T 17444-2013*Measuring Methods for Parameters of Infrared Focal Plane Arrays*, Standardization Administration of the People's Republic of China: Beijing, China, 2014.