

## Research Article

# Dim-Small Target Detection Based on Adaptive Pipeline Filtering

Biao Li,<sup>1,2,3,4</sup> Xu Zhiyong ,<sup>1,3</sup> Jianlin Zhang ,<sup>1,3</sup> Xiangru Wang ,<sup>2</sup> and Xiangsuo Fan<sup>5</sup>

<sup>1</sup>Institute of Optics and Electronics, Chinese Academy of Sciences, Chengdu 610209, China

<sup>2</sup>School of Optoelectronic Science and Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China

<sup>3</sup>University of Chinese Academy of Sciences, Beijing 100049, China

<sup>4</sup>Key Laboratory of Optical Engineering, Chinese Academy of Sciences, Chengdu 610209, China

<sup>5</sup>School of Electrical and Information Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China

Correspondence should be addressed to Jianlin Zhang; [jlin@ioe.ac.cn](mailto:jlin@ioe.ac.cn)

Received 11 January 2020; Revised 15 March 2020; Accepted 7 April 2020; Published 30 May 2020

Academic Editor: Luca Chiapponi

Copyright © 2020 Biao Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In order to improve the robustness of the pipeline target detection algorithm against strong noises and occlusion, this paper presents an *adaptive pipeline filtering algorithm* (APFA). In APFA, the velocity and the center of the target are firstly predicted based on the smooth motion trajectory after background suppression. Then, time-domain energy enhancement of targets is adopted to improve the obscure target detection, and adaptively updating the center and radius of the pipeline filter are carried out for targets' motion variation. Experiments on five different typical scenes show that APFA can improve the robustness of the pipeline filter against strong noises and even when targets are temporarily obscured partially or completely. Simultaneously, APFA can significantly improve the energy and signal-to-noise ratio of targets, and as a result, the target detection rate is significantly promoted on all experiments.

## 1. Introduction

In complex scenes, signal-to-noise ratios and contrasts between dim-small targets and their background are low due to their small scale and weak energy. Therefore, background suppression is needed before target detection [1, 2]. After background suppression, the signal-to-noise ratios and contrasts between targets and their background are improved in difference images, and it is conducive to improve the effect of target detection. Traditional dim-small target detection algorithms can be classified into two categories, namely, detect before track (DBT) [3–5] and track before detect (TBD) [6–8]. The *pipeline target detection algorithm* (PTDA) first proposed by Wang et al. [3] is a classic algorithm of DBT. Later, other improved algorithms were proposed [9–11]. They are collectively referred to as the *traditional pipeline filtering algorithm* (TPFA) in this paper. The TPFA has the advantages of easy implementation and strong real-time performance. The basis of TPFA is that trajectories of small targets are continuously smooth, and the

energy intensity of targets is higher than that of their surrounding backgrounds [3]. Under this condition, TPFA has three assumptions: first, the speed of the targets is limited to about 1 pixel/frame [9, 11, 12]; second, the maximum target loss rate in the detection pipeline is one frame per five frames [9, 11, 12]; third, the signal-to-noise ratio of targets should not be lower than 5 dB [2]. In other words, it requires that targets' motion trajectories must be continuous, smooth, and without fracture, and targets' velocities must be almost immobilized. So, TPFA has some shortcomings. First, it is sensitive to noises. When strong noises exist in the pipeline area, the algorithm may be invalid. For example, when there are continuous strong noises in fixed positions in the pipeline area or there are strong noises at the edge of the pipeline, the algorithm may take the noise points as the targets, which will lead to misjudgment and affect the accuracy of subsequent detection. Second, it would not enhance the target energy. So, when noises are as strong as targets, there will be a mistake in target detection. Third, its robustness is poor. In the case of strong noise interference or

temporary occlusion, the target detection would fail. Fourth, the determination of pipeline centers and pipeline radius is too mechanical, so it cannot be adaptive to the target moving variation. For the first shortcoming, some related research studies have been done. Among them, Liu and Ji [9] proposed to correct the center of the pipeline by using the deviation between the center of the pipeline and that of the target, in order to avoid the noises at the edge of the pipeline affected the determination of the center of pipeline; in [10], a threshold of a dot occurrence frequency was set to eliminate the fixed noises in the pipeline; in [11], the continuity of the target trajectory was used to estimate the target motion direction, so as to suppress the interference of noises that were outside the area of the target motion direction in the pipeline. Essentially, all of these methods have an important shortcoming, that is, they did not make full use of the information that is related to target motion and target pixels. So, these algorithms would fail when targets are weak or targets temporarily disappear. For example, these algorithms would fail if targets are temporarily obscured or completely submerged in background clutters. Therefore, in order to effectively overcome these shortcomings, an improved pipeline target detection algorithm named as the *adaptive pipeline filtering algorithm* (APFA) is presented in this paper.

The contribution of this paper is to present the APFA that has remarkable robustness. It can be mainly reflected in two aspects: first, APFA has time-domain energy enhancement performance. Time-domain energy enhancement improves the contrast between the target and the residual background and improves the robust performance of the algorithm when targets are obscured. Second, APFA can adaptively update the center and radius of the pipeline, so that it is robust against noise interference and target motion variation.

Section 2 of this paper mainly presents the improved method that can enhance the energy of targets in the time domain and adaptively update the center and radius of the pipeline. In Section 3, experiments on five different typical scenes are demonstrated to verify and evaluate the performance of APFA, which includes the test of time-domain energy enhancement of targets and the test of target detection robustness against strong noises and occlusion. Section 4 is to draw a conclusion of this paper.

## 2. The Adaptive Pipeline Filtering Algorithm

The three main parameters of the pipeline filtering algorithm are the center coordinate  $G$ , the radius  $R$ , and the length  $L$  of the pipeline. Among them,  $L$  affects the judgment of the target, while the other two parameters,  $G$  and  $R$ , directly determine whether the target can be successfully retrieved. The research of APFA is focused on  $G$  and  $R$ .

**2.1. Image Preprocessing.** In the complex background, the contrast between the dim-small target and the background is low because the target is small and weak. Therefore, in order to enhance the target, image preprocessing, namely, background suppression, should be carried out before the target

detection. In this paper, background suppression is implemented by the *statistical region low-rank background modeling algorithm* (SRLBMA). SRLBMA (under review) is an improved low-rank background modeling algorithm:

$$I = L + S. \quad (1)$$

The existing mature low-rank background modeling algorithms [13, 14] model the whole video image as a kind of superimposition that some sparse components, namely, targets and random noises, interfere on the low-rank background, just as shown in equation (1), where  $I$  is the original video image,  $L$  is the low-rank background, and  $S$  represents the sparse components, i.e., targets and random noises. However, SRLBMA does not directly establish the model of the whole video image as shown in equation (1) but models and solves the statistical clustering regions of the video image. The purpose of SRLBMA is to eliminate excessive nonstationary residues of background suppression and improves the contrast between the target and the residual background.

Firstly, SRLBMA performs statistical clustering of original video images. Secondly, a low-rank background model as shown in equation (2) is established for each statistical clustering region. Thirdly, the optimal solution of equation (2) is solved and the low-rank background  $B$  of each statistical clustering region is obtained by using equations (3) to (5) [13–16]. Equations (3) to (5) are the equivalent improved algorithm for the augmented Lagrange multiplier algorithm [15, 17]. Fourthly,  $B$  is superimposed to each other to obtain the low-rank background  $L$  of the original video image. Finally, the difference image  $S$  is obtained by subtracting the low-rank background  $L$  from the video image  $I$ :

$$F = B + P, \quad (2)$$

$$B_{k+1} = H_{(1/\mu)}(F - P_k + \mu^{-1}Y_k), \quad (3)$$

$$P_{k+1} = P_{(\lambda/\mu)}(F - B_{k+1} + \mu^{-1}Y_k), \quad (4)$$

$$Y_{k+1} = Y_k + \mu(F - B_{k+1} - P_{k+1}). \quad (5)$$

In equation (2),  $F$  is the statistical region image matrix,  $B$  is the low-rank matrix of  $F$ , and  $P$  is the sparse matrix of  $F$ . The parameters  $\lambda$  and  $\mu$  are specified in Section 3.1.

**2.2. Adaptive Pipeline.** Figure 1 represents a curve of a function  $f(\cdot)$  that describes an event. Figure 1(a) represents the intact curve of  $f(\cdot)$ , and Figure 1(b) represents the curve where the information of  $f(\cdot)$  is lost at  $D$  and  $E$  for some reason. So, whether the lost information at  $D$  and  $E$  of function  $f(\cdot)$  can be retrieved by prediction? The answer is obvious: if the function  $f(\cdot)$  is continuous and smooth, that is, the function  $f(\cdot)$  is regular, the lost information at  $D$  and  $E$  can be predicted and retrieved by the information before  $D$  and  $E$ , respectively. So, it can be seen that unknown information of regular events can be predicted by using the existed information. Actually, the motion of the natural object is continuous and smooth in a certain time range. Taking advantage of this objective fact, the

position and velocity of the moving target can be predicted, and then the pipeline target detection algorithm can be improved by using the prediction information. At the same time, the prediction information can be combined with the target pixel information to enhance the energy of the target. In this way, the target can also be predicted and detected even when the target trajectory is temporarily discontinuous, that is, when the target is temporarily obscured or lost. Therefore, the continuous and smooth property of the trajectory of the target are extended to the predictability of the information of the motion target, which is the basis for the establishment of the APFA.

The motions of targets in the video frame images are represented by three-dimensional data  $D(x, y, t)$ , as shown in Figure 2. In Figure 2, investigating targets and their neighborhood region, video frame images can be divided into three categories. The first are those where the target pixels are stronger than their neighboring nontarget pixels, such as the image frames before the moment  $t_A$ , that is, frames before the frame  $A$ ; the second category are those where noise pixels are stronger than target pixels, for example, the  $B$ ,  $C$ , and  $D$  frames corresponding to the moment of  $t_B$ ,  $t_C$ , and  $t_D$ , and the strong noise points in the figure are represented by red dots. The strong noise point  $n_B$  of frame  $B$  is closer to the target, the strong noise point  $n_C$  in frame  $C$  is near the pipeline edge, and in frame  $D$ , the target is weaker than the noise. The third category are those where targets are temporarily obscured, e.g., images of frame  $E$  to  $F$  corresponding to time  $t_E$  to  $t_F$ . When the pipeline filter is applied to detect targets, the strong noises in the second type images may cause false detection, that is, noises are taken as targets, which will affect the determination of the center positions of the pipeline filter and make the subsequent detection unable to continue. For example, under the influence of the noise  $n_C$ , the center of the pipeline filter in the next frame may be mistakenly moved to the position where  $n_C$  is, namely, the edge position of the pipeline. As a result, the pipeline filter in the next frame fails to contain the target, so the target cannot be detected forward. Images of the third type have no target points, so the detection cannot continue effectively. Since the motion of targets is continuous and smooth, the positions and velocities of targets can be predicted by making full use of the motion information such as the speed and direction of targets. In order to improve the robustness of the pipeline filter detection algorithm, after target positions and velocities are predicted, works of two aspects need be done. The first is to enhance the time-domain energy of targets by using the information of target pixels; the second is to use the information of positions and velocities of targets to adaptively update  $G$  and  $R$  of the pipeline filter. In this way, the influence of random strong noises and temporary obscure on target detection can be effectively solved.

### 2.2.1. Position and Velocity Prediction of Targets.

Classical optimal estimation algorithms include least square method, maximum likelihood method, Wiener filter method, and Kalman filter method. The Kalman filter method is a time-domain filtering algorithm, which describes the system with state space and adopts the recursive

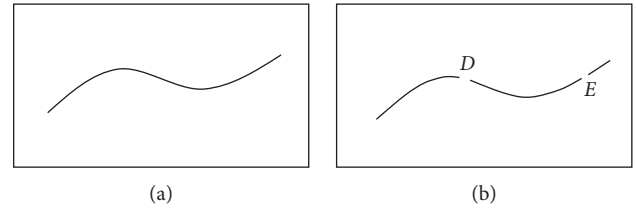


FIGURE 1: A curve of the function  $f(\cdot)$  that describes an event. (a) The intact curve of  $f(\cdot)$ ; (b) the curve of  $f(\cdot)$  where parts of the information are lost at  $D$  and  $E$ .

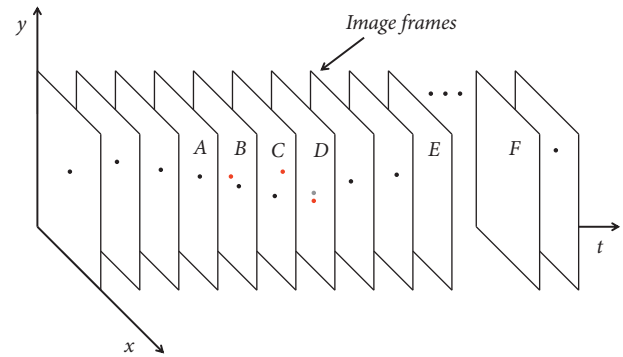


FIGURE 2: Three types of target points. The target points are represented by small black dots, the strong noise points are represented by red dots, and frames  $E$  to  $F$  have no target points.

iteration method for optimization. The calculation amount and storage capacity of the Kalman filter method are relatively less than those of other optimal estimation algorithms, and it is applicable to multidimensional and various random processes [18]. Therefore, the Kalman filtering method is widely applied in various fields. So, the Kalman filter algorithm is used in this paper to predict the centroids and velocities of targets.

If the time of examining the motion state of targets is relatively short, the motion of targets can be approximately regarded as a uniform rectilinear motion. The state equation and the observation equation of the Kalman filter system can be expressed as follows:

$$\begin{aligned} X(k+1) &= F * X(k) + Q * W(k), \\ Z(k+1) &= H * X(k+1) + V(k+1), \end{aligned} \quad (6)$$

where  $x(k)$ ,  $y(k)$ , and  $v_x(k)$ ,  $v_y(k)$  in the state variable  $X(k) = [x(k), y(k), v_x(k), v_y(k)]^T$  are the centroid coordinates and velocities of targets in the  $x$  and  $y$  directions, respectively;  $F$  is the state transition matrix;  $Q$  is the noise-driven matrix;  $H$  is the observation matrix;  $W(k)$  and  $V(k)$  are the process noise and observation noise, respectively; and the observation variable  $Z(k)$  denotes the observed centroid coordinates of targets in the  $x$  and  $y$  directions.

**2.2.2. Time-Domain Energy Enhancement of Targets.** The center of the pipeline filter is mainly determined by the centroid coordinates of targets. However, under the

influence of strong noises, targets may be undetected. For instance, in Figure 2, under the influence of strong noise  $n_C$ , the pipeline filter may take  $n_C$  as the target, and then in the detection of the next frame, the pipeline filter would take the centroid of  $n_C$  as the center of the pipeline filter. As another example in Figure 2, targets are disappeared in frames at the moment from  $t_E$  to  $t_F$ . Under this situation, due to the absence of targets, the pipeline filter may select a strong noise as the target point, so the centroid of this strong noise will be taken as the center of the pipeline filter in the next frame. The deviation of the center position of the pipeline will affect the target detection of the pipeline filter. In order to improve the robustness of the pipeline filter against strong noise interference or targets' obscuration, a time-domain target energy enhancement algorithm is proposed in this section.

The energy enhancement of targets can be realized by using the predicted positions of targets. Because dim-small targets are very small in the field of vision, the size and motion information of targets are relatively stable in a short time. Therefore, after the centroid of the target of the following frame is predicted out, the time-domain energy enhancement operation of the target can be carried out. That is, the target pixels' gray value information of several previous frames could be superimposed on the target of the prediction frame, so as to effectively enhance the target energy of the prediction frame. The specific formula is as follows:

$$T'_N = T_{N-m} + T_{N-(m-1)} + \cdots + T_{N-2} + T_{N-1} + T_N, \quad (7)$$

where  $m$  is the number of frames that are superimposed on the frame that needs to be predicted,  $T_N$  is the original energy of the target of the frame to be predicted,  $T_{N-m}, T_{N-(m-1)}, \cdots, T_{N-2}, T_{N-1}$  are the target energy of frames  $N-m$  to  $N-1$  before the predicted frame, and  $T'_N$  is the enhanced target energy of the frame to be predicted.

This enhancement operation can be classified into three scenarios. In the first case, there is complete target information in the frame to be predicted. In this case, the target energy of the frame to be predicted can be greatly enhanced through the time-domain energy enhancement operation. In the second case, only partial information of the target is contained in the frame to be predicted, that is, the target is partially obscured temporarily. In this case, the lost target information can be properly compensated through the time-domain energy enhancement operation, and the whole target energy can be effectively enhanced in the prediction frame. In the third case, there is no target in the prediction frame, that is, the target is completely obscured and lost temporarily. In this case, the target information can be estimated at the predicted position of the target through the time-domain energy enhancement operation, and the energy of the estimated target has been enhanced.

**2.2.3. Pipeline Center Adaptive.** Adaptive pipeline is another work to improve the robustness of the pipeline filter, which mainly includes pipeline center adaptively tuning and

pipeline radius adaptively tuning. The center position of the pipeline is mainly determined by the centroid coordinate of the target. In the ideal case without random noises, the coordinate of the target centroid can be directly taken as the coordinate of the pipeline center. But in fact, pipeline filter detection is susceptible to random noises in the pipeline area, especially the strong noise at the edge of the pipeline. Therefore, in order to improve the robustness of the pipeline filtering detection algorithm against strong noises, the pipeline center adaptive tuning algorithm is given as follows:

$$\begin{aligned} C_k^x &= C_{k-1}^x + a_x, \\ C_k^y &= C_{k-1}^y + a_y, \end{aligned} \quad (8)$$

where  $C_k^x$  and  $C_k^y$  are the centroid coordinates of the pipeline in  $x$  and  $y$  directions of frame  $k$ , respectively,  $C_{k-1}^x$  and  $C_{k-1}^y$  are the centroid coordinates of the pipeline in  $x$  and  $y$  directions of frame  $k-1$ , respectively, and  $a_x$  and  $a_y$  are the correction factors, and they can be determined by the following formula:

$$\begin{aligned} a_x &= \begin{cases} x_{k-1} - C_{k-1}^x, & x_{k-1} - C_{k-1}^x \leq V_x^{k-1}, \\ \frac{x_{k-1} - C_{k-1}^x}{10b_x}, & x_{k-1} - C_{k-1}^x > V_x^{k-1}, \end{cases} \\ a_y &= \begin{cases} y_{k-1} - C_{k-1}^y, & y_{k-1} - C_{k-1}^y \leq V_y^{k-1}, \\ \frac{y_{k-1} - C_{k-1}^y}{10b_y}, & y_{k-1} - C_{k-1}^y > V_y^{k-1}, \end{cases} \end{aligned} \quad (9)$$

where  $V_x^{k-1}$  and  $V_y^{k-1}$  are the target velocities in  $x$  and  $y$  directions of frame  $k-1$ , respectively,  $x_{k-1}$  and  $y_{k-1}$  are the centroid coordinates of the target of frame  $k-1$ , respectively, and  $b_x$  and  $b_y$  are the order of the magnitude of  $(x_{k-1} - C_{k-1}^x)$  and  $(y_{k-1} - C_{k-1}^y)$ , respectively.

$V_x^{k-1}$  and  $V_y^{k-1}$  are taken as thresholds, and correction factors  $a_x$  and  $a_y$  are self-adaptively evaluated according to the distance of the target moved in  $x$  and  $y$  directions in the front and rear frames, respectively. Under normal circumstances, the moved distance of the target in the front and rear frames in  $x$  and  $y$  directions will not exceed the moving speed  $V_x^{k-1}$  and  $V_y^{k-1}$  of the target in  $x$  and  $y$  directions, so the moved distance of the target is considered reasonable. If the moved distance of the target exceeds the moving speed of the target, it may be caused by strong noises. In this case, the correction factor should be reduced correspondingly to decrease the interference of strong noises. Since the velocity and centroid coordinates of the target have been obtained during prediction and detection operation, velocities  $V_x^{k-1}$  and  $V_y^{k-1}$  are taken as the threshold and correction factors  $a_x$  and  $a_y$  will be adjusted automatically when the velocity changes, which can make the determination of the coordinates of the pipeline center be more adaptive.

**2.2.4. Pipeline Radius Adaptive.** The radius of the pipeline is mainly determined by the velocity of the target, which is generally an integer multiple of the target velocity. Since the



TPFA assumes that the target speed does not exceed 1 pixel per frame, the pipeline radius is directly set a fixed constant value. But in fact, the moving speed of the target may change or exceed 1 pixel per frame. Therefore, it is necessary to adjust the pipeline radius according to the changed speed. Otherwise, when the target moves beyond the range of the pipeline area, the target will not be detected. Therefore, in order to make the pipeline radius adaptive to the changed speed and improve the robustness of the detection, the pipeline radius adaptive algorithm is given as the following equation:

$$\begin{aligned} R_k &= R_{k-1} + d, \\ R_0 &= cV_0, \end{aligned} \quad (10)$$

where  $R_k$  is the pipeline radius of frame  $k$ ,  $R_{k-1}$  is the pipeline radius of frame  $k-1$ , and  $R_0$  is the initial pipeline radius;  $c$  is a constant, that is, it is a positive integer;  $V_0 = \max(|V_x^0|, |V_y^0|)$  is the maximum speed in  $x$  and  $y$  directions of the target of the initial frame; and  $d$  is the correction factor, which can be determined by the following formula:

$$d = \begin{cases} V_\Delta, & V_\Delta \leq V_{k-1}, \\ \frac{V_\Delta}{10e}, & V_\Delta > V_{k-1}, \end{cases} \quad (11)$$

where  $V_\Delta = \max(|V_x^{k-1} - V_x^{k-2}|, |V_y^{k-1} - V_y^{k-2}|)$  is the maximum velocity difference in  $x$  and  $y$  directions of targets in frames  $k-1$  and  $k-2$ ,  $V_{k-1} = \max(|V_x^{k-1}|, |V_y^{k-1}|)$  is the maximum velocity in  $x$  and  $y$  directions of targets in frame  $k-1$ , and  $e$  is the order of magnitude of  $V_\Delta$ .

In the pipeline radius adaptive algorithm, the pipeline radius  $R$  is mainly determined by  $R_{k-1}$ , and it is adaptively adjusted according to the velocity variation  $V_\Delta$ . Under normal circumstances,  $V_\Delta$  does not exceed  $V_{k-1}$  at the  $k-1$  frame, so it is considered reasonable. When the speed suddenly changes significantly leading to  $V_\Delta$  greater than  $V_{k-1}$ , it indicates that the detection result may be influenced by strong noises, so the value of the correction factor  $d$  should be reduced to mitigate the interference of strong noises. Similarly, the velocity of the target has been obtained during prediction and detection operation, and  $V_{k-1}$  is taken as the threshold, so the correction factor  $d$  will be adjusted automatically when the velocity changes, which can make the determination of the pipeline radius be more adaptive.

**2.3. The APFA Algorithm.** The APFA, namely, Algorithm 1, is an organic combination of target prediction, time-domain energy enhancement of targets, pipeline center adaptive tuning, and pipeline radius adaptive tuning.

After the backgrounds of original video images are suppressed by SRLBMA to get sequence difference images  $S$ , Algorithm 1 is carried out. Firstly, the centroid and velocity of the target of the  $N$ th frame are predicted by using the information of targets in previous frames; secondly, the target energy of frame  $N$  is enhanced in the time domain. After the time-domain energy enhancement operation, if

there is a target in the original  $N$ th frame, the target will be significantly enhanced; if there is only part signal of the target in the original  $N$ th frame, the target information will be compensated and the energy will be effectively enhanced; if there is no target in the original  $N$ th frame due to obscured or other reasons, the target information will be estimated and the energy will be enhanced. Thirdly, the obtained information of centroids and velocities of targets is used to adaptively adjust the center and radius of the pipeline filter to eliminate the interference caused by strong noises. Finally, the adaptive pipeline filter is used to detect targets.

In order to understand the algorithm and demonstrate its difference from TPFA intuitively, both the flow diagrams of APFA and TPFA are shown in Figure 3. Comparing with TPFA, it can be seen from Figure 3 that based on the prediction information, the detection of APFA can be just confined in the prediction area. In addition, it is more robust.

### 3. Experiments

In this section, five different scenes are given, deep space scene  $A$ , complex sky scene  $B$ , complex forest scene  $C$ , complex space scene  $D$ , and sky scene  $E$ . Among them, except scenes  $B$  and  $C$ , the other three scenes are the field application signals. Scene  $A$  was got by an optical imaging system with 1-meter aperture and 5-meter focal length, scene  $D$  was got by an optical imaging system with 0.5-meter aperture and 4-meter focal length, and scene  $E$  was got by a digital camera. The five scenes are used to carry out experimental verification of the APFA algorithm. First, background suppression is carried out. Second, robust performance experiments of APFA are conducted. It includes the time-domain energy enhancement experiment of targets and target detection experiments when targets are temporarily partial or entire obscured. Third, comparison experiments of target detection are conducted for the whole motion process of targets to verify the overall target detection performance of APFA.

**3.1. Background Suppression.** The SRLBMA algorithm is used for the background suppression experiment. In SRLBMA, the parameters  $\lambda = 1/\max(o, q)$  and  $\mu = o * q/4\|F_1\|$ ;  $o$  and  $q$  are the size of the statistical region image matrix  $F \in R^{o \times q}$ ; and the convergence condition is  $\|F - B - P_F\| \leq \varphi\|F_F\|$ , where  $\varphi = 10^{-7}$  [15].

Scene  $A$  is a deep space scene. There are two targets in each frame, where target 2 is below target 1. The energy and size of target 2 are obviously larger than those of target 1, and the energy of target 1 is very weak. They flew from the lower right corner to the upper left corner in the frame. Scene  $B$  is a complex sky scene. Moving from the upper right to the lower left, the target in scene  $B$  is eventually submerged in the turbulent layer of the atmosphere above the clouds. Scene  $C$  is a complex forest scene. The target is moved from the right to the left at the edge of the forest and is constantly, partially, or completely obscured by the trees as it is moving. In scene  $D$ , the target moves from the bottom to the top of the image.

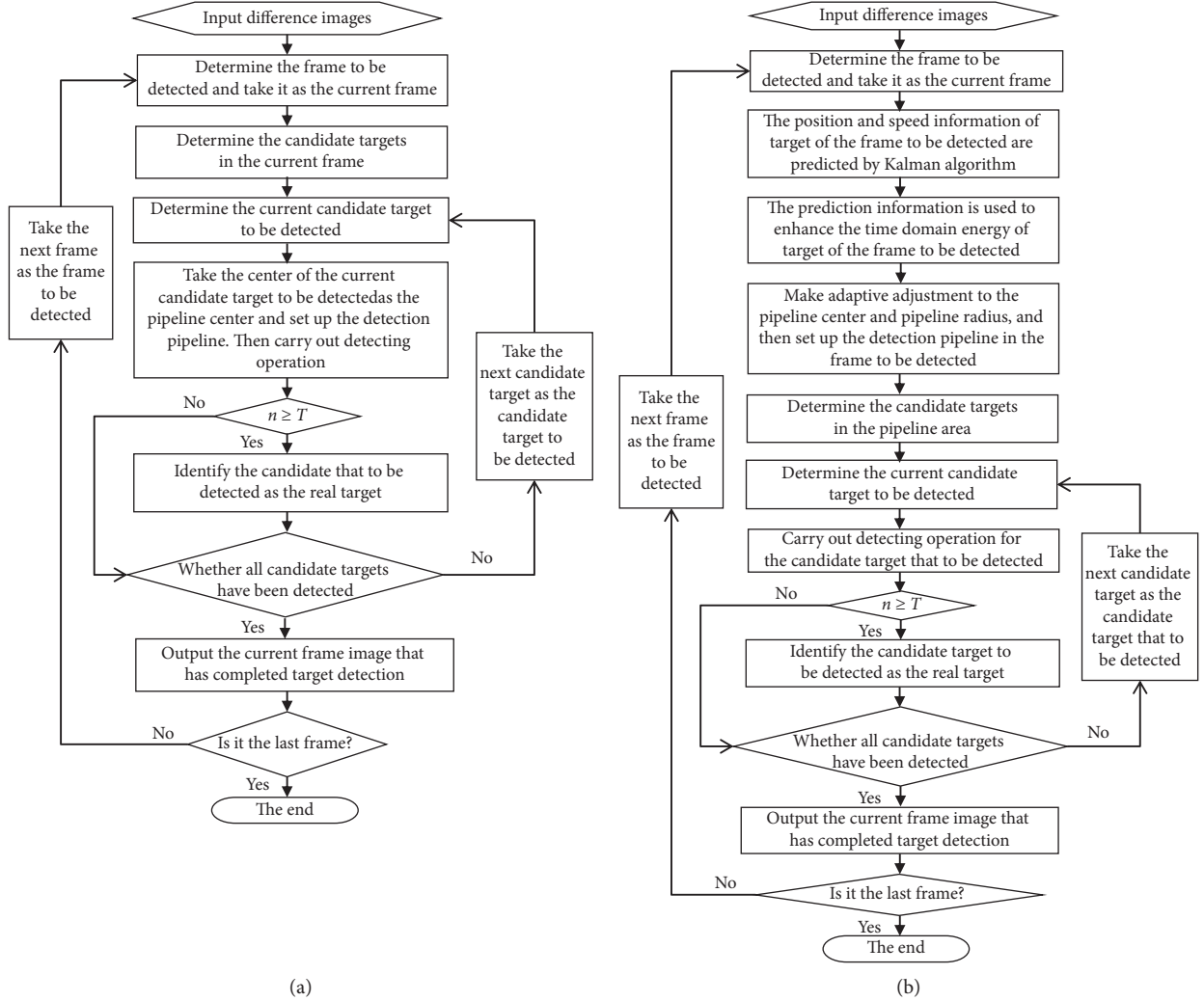


FIGURE 3: The flow diagram of the TPGA and APFA methods. (a) The TPGA flow diagram. (b) The APFA flow diagram.  $n$  denotes the number of times that the candidate target is detected in the pipeline area, and  $T$  denotes the judgment threshold.

From frames 103 to 123, the target is submerged in the highlight background. In scene  $E$ , the target is moved from the left to the right of the image. For scenes  $A, B, C, D$ , and  $E$ , the difference images that were obtained by background suppression are shown in Figure 4. Figure 4 shows only frame 19 of scene  $A$ , 64 of scene  $B$ , 115 of scene  $C$ , 249 of scene  $D$ , and 61 of scene  $E$ .

Based on our PC platform with the Windows7 operating system, Intel Core i5 CPU, 3G memory, and Matlab R2017b, the time consumptions of SRLBMA for scenes  $A, B, C, D$ , and  $E$  are given in Table 1.

**3.2. Robust Performance Experiments of APFA.** In this section, firstly, the energy enhancement experiments of targets in the time domain are carried out, and then the next experiments are about the detection of targets that are temporarily partially or completely obscured.

**3.2.1. Energy Enhancement of Targets in Time Domain.** The relationship between the number  $m$  of superimposed frames and the effect of time-domain energy enhancement of targets is discussed by carrying out the experiment for scene  $B$ . Since the TPGA starts to be unstable and large false detection occurred after frame 331, images before frame 331 of scene  $B$  are used for comparing analysis in order to better illustrate the problem. The input signal-to-noise ratio ( $SNR_i$ ), output signal-to-noise ratio ( $SNR_o$ ), signal-to-noise ratio gain ( $SNR_g$ ), and the average gray scale value of the target ( $Ave\_gray$ ) are used as evaluation indexes. They are shown in equations (12) to (15):

$$SNR_i = 10 \lg \left( \frac{\text{target\_in}_{\text{mean}} - \text{original\_image}_{\text{mean}}}{\text{original\_image}_{\text{std}}} \right), \quad (12)$$

$$SNR_o = 10 \lg \left( \frac{\text{target\_out}_{\text{mean}} - \text{out\_image}_{\text{mean}}}{\text{out\_image}_{\text{std}}} \right), \quad (13)$$

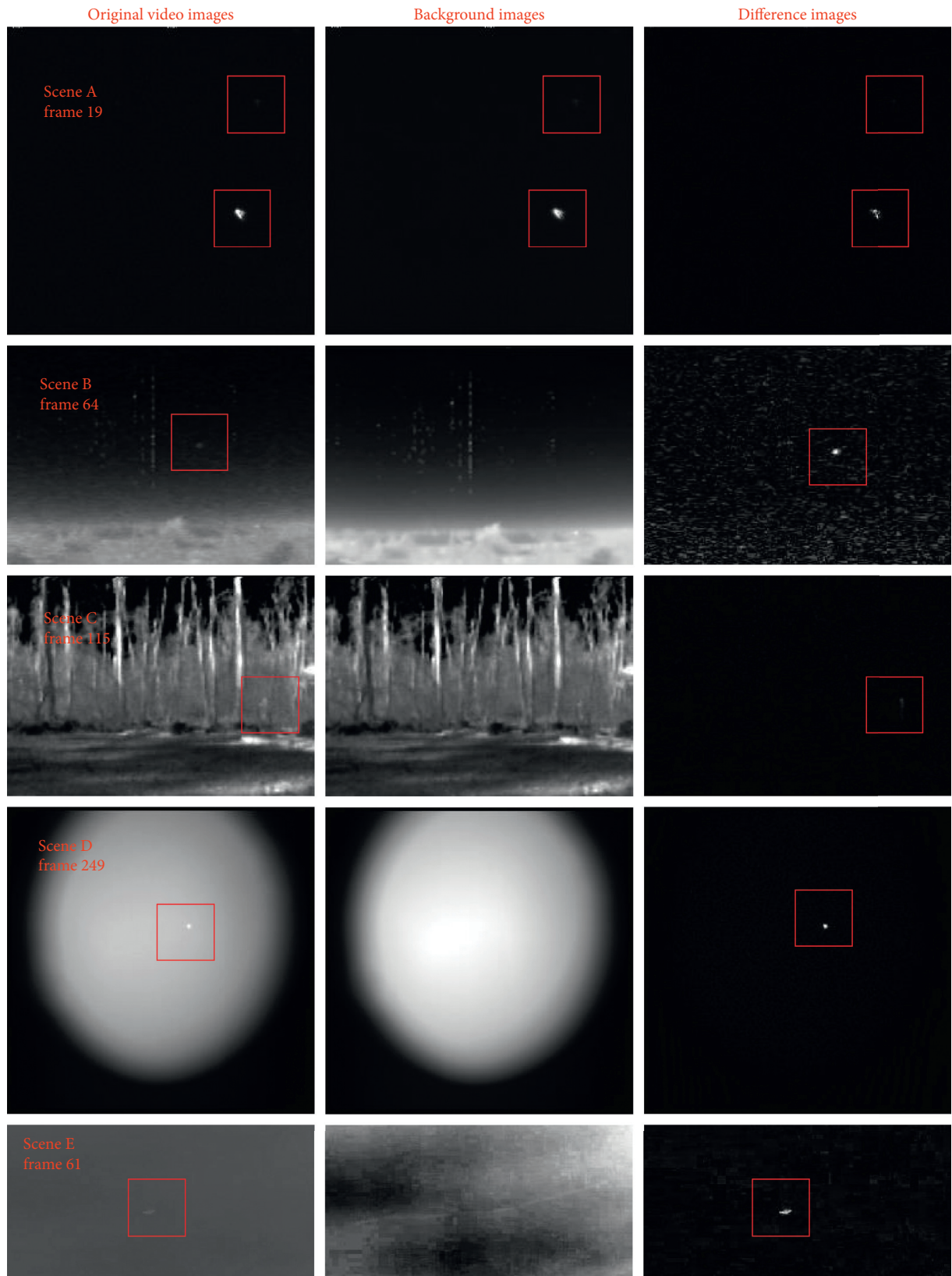


FIGURE 4: Background suppression results of scenes A, B, C, D, and E. The first to third columns correspond to the original video images, background images, and difference images, respectively.

```

(1) Input: Sequence difference images  $S \in R^{w \times z}$  that were obtained by the SRLBMA algorithm.
    Parameters  $m, F, H, Q, W(k), V(k), c,$  and  $V_0$ 
(2) while not the last image do
(3) // Centroids and velocities prediction of targets
     $X(k+1) = F * X(k) + Q * W(k)$ 
     $Z(k+1) = H * X(k+1) + V(k+1)$ 
(4) // Time-domain energy enhancement of targets
     $T_N^i = T_{N-m} + T_{N-(m-1)} + \dots + T_{N-2} + T_{N-1} + T_N$ 
(5) // Pipeline centers adaptive and pipeline radiuses adaptive
     $C_k^x = C_{k-1}^x + a_x$ 
     $C_k^y = C_{k-1}^y + a_y$ 
     $R_k = R_{k-1} + d$ 
(6) Pipeline target detecting
(7) end while
(8) Output: target images  $T$ 

```

ALGORITHM 1: The adaptive pipeline filtering algorithm (APFA).

$$\text{SNR}_g = 10 \lg \left( \frac{\text{SNR}_o}{\text{SNR}_i} \right), \quad (14)$$

$$\text{Ave\_gray} = \frac{\text{gray}_{\text{sum}}}{\text{Num}_{\text{pixel}}}, \quad (15)$$

where  $\text{target\_in}_{\text{mean}}$  and  $\text{target\_out}_{\text{mean}}$  are the targets' mean gray value of input images and output images, respectively;  $\text{original\_image}_{\text{mean}}$  and  $\text{original\_image}_{\text{std}}$  are the gray scale value mean and standard deviation of the neighborhood areas of targets of input images, respectively;  $\text{out\_image}_{\text{mean}}$  and  $\text{out\_image}_{\text{std}}$  are the mean gray value and standard deviation of the neighborhood areas of targets of output images, respectively;  $\text{gray}_{\text{sum}}$  is the sum of the gray value of target pixels; and  $\text{Num}_{\text{pixel}}$  is the total number of target pixels.

It is known from the previous discussion that for the current frame to be detected, the target centroid will be predicted and the targets' energy in previous  $m$  frames will be superimposed onto the detected frame to enhance the target energy. Obviously, the energy of the target increases with the increase in the number  $m$  of superimposed frames, which is indeed true from the experimental results. Frames 131, 236, and 315 of scene B are randomly selected to calculate  $\text{SNR}_i$ ,  $\text{SNR}_o$ ,  $\text{SNR}_g$ , and  $\text{Ave\_gray}$  of targets before and after time-domain energy enhancement, and the data are filled in Tables 2 and 3. It can be seen from Tables 2 and 3 that  $\text{SNR}_o$ ,  $\text{SNR}_g$ , and  $\text{Ave\_gray}$  of targets of output images are obviously higher than those without superimposed enhancement, and  $\text{SNR}_o$ ,  $\text{SNR}_g$ , and  $\text{Ave\_gray}$  of targets of output images are increased with the increase in  $m$ .

However, is the larger the number  $m$  of superimposed frames, the better the result of energy enhancement? If only from the perspective of energy enhancement, it is true that the larger the  $m$ , the more significant the energy enhancement effect will be obtained. However, with the increasing value of  $m$ , two serious problems arise: the first problem is that the target is extended, that is, the area that is occupied by the superimposed target is larger than that of the real target.

TABLE 1: The time consumptions of SRLBMA for scenes A, B, C, D, and E.

Scenes	Total frames	Image size	Total time	Average time per frame
A	110	250 * 250	746.4270	6.7857
B	532	250 * 180	4339.5772	8.1571
C	521	250 * 180	3171.1707	6.0867
D	399	256 * 256	3582.6609	8.9791
E	115	300 * 150	943.8970	8.2078

The time unit is second.

The reason for this problem is that there is a certain difference in the area that is occupied by the target of different frames, so the target that is superimposed by multiple frames must be expanded. Another problem is that superimposed targets can interfere with real targets, and the reason for this problem is that no prediction can avoid error. If there is a significant deviation of the centroid coordinate value between the predicted and real target, it may lead that the real target is located on the edge or outside of the superimposed target, and at the same time, if the energy of the real target is lower than that of the superimposed target, the target detection result will be unstable or even failure. At the same time, this phenomenon will be fed back to the prediction system, which further exacerbates the error of the centroid of the predicted target in the subsequent image frames. These two problems affect the detection results together, and the larger the  $m$ , the more obvious the effect. From the experimental data in Figures 5 and 6, it can be clearly seen that after the time-domain energy enhancement, the target energy is significantly enhanced, but the phenomenon of target expansion and the coordinate deviation of the target centroid are also generated. Figures 5 and 6 shows the change of target size and centroid position of the representative frames 100, 214, 274, and 322 on the three conditions that of without superposition enhancement, superposition of 2 frames, and superposition of 8 frames, respectively. It can be seen from Figures 5 and 6 that at frame 100, the overlap between the superimposed target and the real target is very good on the two conditions that of superimposed 2 frames



TABLE 2: SNR<sub>i</sub>, SNR<sub>o</sub>, and SNR<sub>g</sub> of targets of difference images for frames 131, 236, and 315 of scene B before and after time-domain energy enhancement.

	Frame 131			Frame 236			Frame 315		
	SNR <sub>i</sub>	SNR <sub>o</sub>	SNR <sub>g</sub>	SNR <sub>i</sub>	SNR <sub>o</sub>	SNR <sub>g</sub>	SNR <sub>i</sub>	SNR <sub>o</sub>	SNR <sub>g</sub>
Superposition of 2 frames	8.9275	11.0483	2.1208	6.3721	10.0170	3.6499	4.7623	9.6999	4.9375
Superposition of 8 frames	8.9275	11.6644	2.7369	6.3721	10.2355	3.8634	4.7623	10.1655	5.4032

Unit: dB.

TABLE 3: Ave<sub>gray</sub> of targets after time-domain energy enhancement of difference images for frames 131, 236, and 315 of scene B before and after time-domain energy enhancement.

	Frame 131	Frame 236	Frame 315
No superposition	94.3913	64.0227	49.0000
Superposition of 2 frames	146.8167	130.2157	121.3396
Superposition of 8 frames	166.1977	136.6744	131.2437

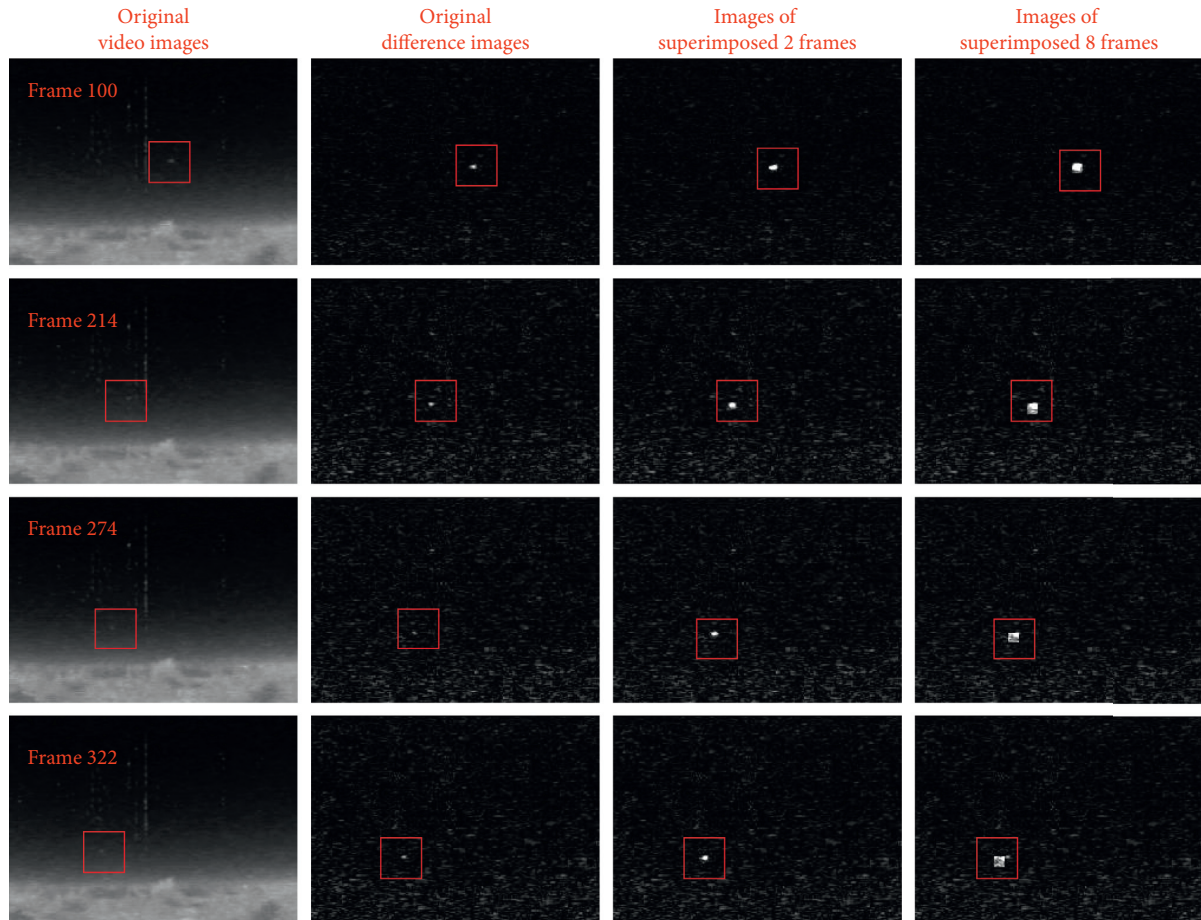


FIGURE 5: Images before and after time-domain energy enhancement of targets at frames 100, 214, 274, and 322. The first to fourth columns correspond to the original video images, original difference images, images of superimposed 2 frames, and images of superimposed 8 frames, respectively.

and superimposed 8 frames; at frames 214, 274, and 322, the overlap between the superimposed targets of superimposed 2 frames and the real targets is still very good; but in the case of superimposed 8 frames, the real target shifts slightly to the upper right of the superimposed target at frame 214, the real target distinctly moves to the upper right edge of the

superimposed target at frame 274, the real target completely moves beyond the upper right edge of the superimposed target at frame 322, and the detection performance becomes more and more unstable after frame 322. So, it can be seen that the value of the superimposed frame number  $m$  is not the larger the better, which should be determined according

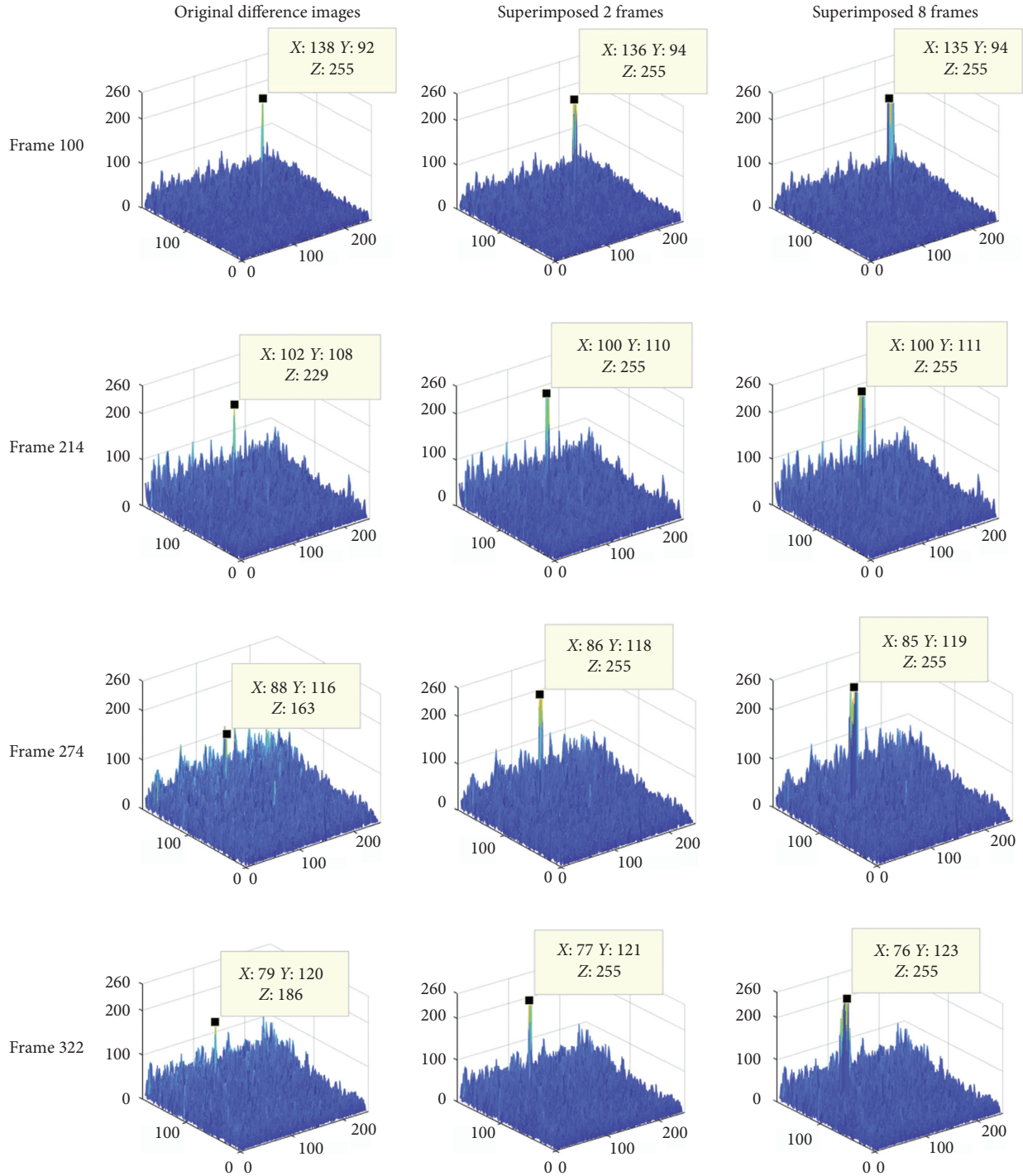


FIGURE 6: Three-dimensional figures of energy before and after time-domain energy enhancement of targets of frames 100, 214, 274, and 322. The first to third columns correspond to the three-dimensional figures of energy of original difference images, images of superimposed 2 frames, and images of superimposed 8 frames, respectively.

to different scenes. Experimental data show that for scene B, in the case of superimposed 2 frames, the overlap between superimposed targets and real targets remains good through all the frames of scene B, which is a total of 532 frames. Therefore, for scene B,  $m=2$  can achieve better effect of energy enhancement and target detection. Similarly, better effect of energy enhancement and target detection can be

achieved when  $m=2$  for scene A,  $m=4$  for scene C,  $m=2$  for scene D, and  $m=2$  for scene E.

**3.2.2. Estimation and Detection Experiment for Targets under Occlusion.** In a complex scene, targets may be submerged by strong noises or temporarily occluded by obstacles. In these

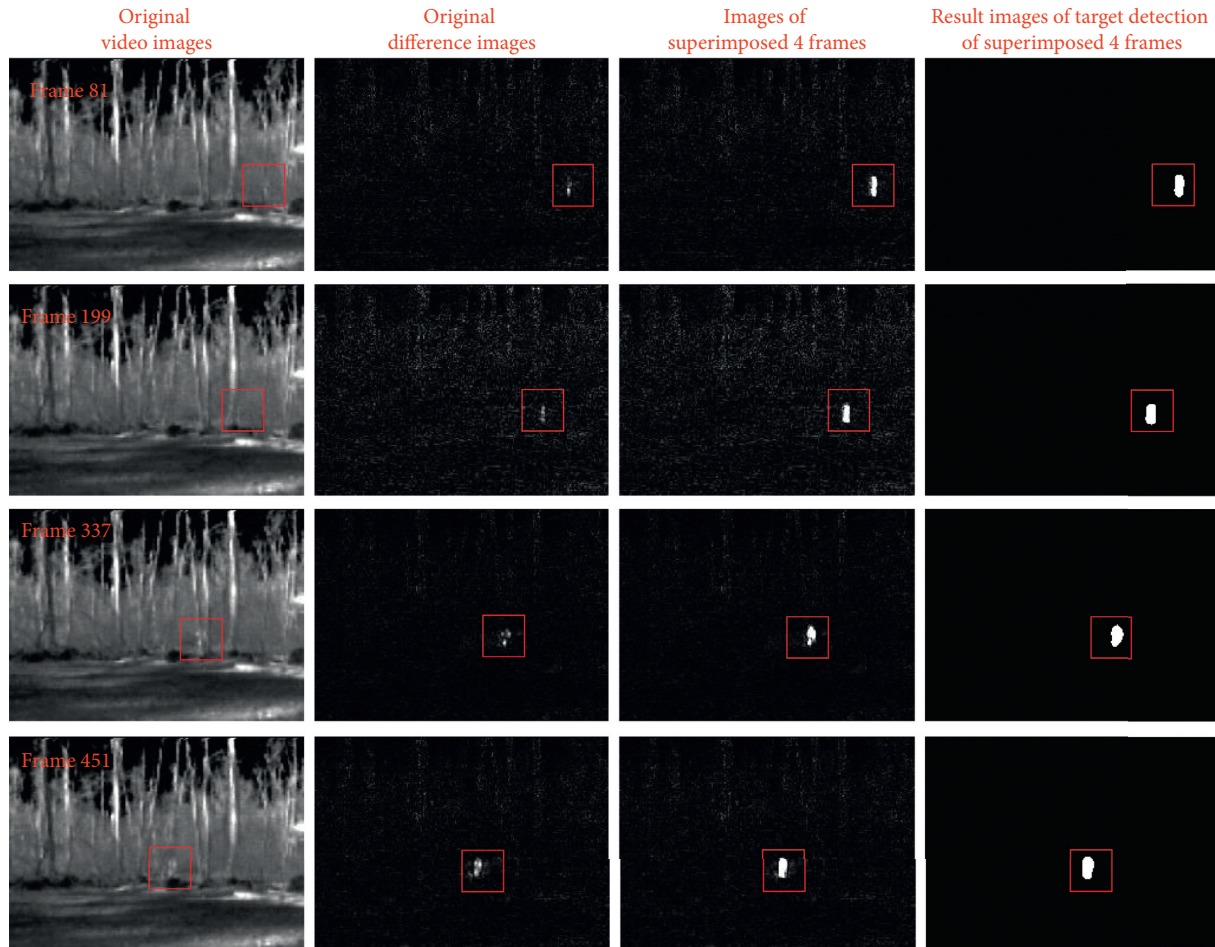


FIGURE 7: The target is partially obscured by trees. The first to fourth rows correspond to the frames 81, 199, 337, and 451, respectively. The first to fourth columns correspond to the original video images, original difference images, images of superimposed 4 frames, and the result images of target detection of superimposed 4 frames, respectively.

cases, the detection performance of TPFA would decrease sharply, and result in instability and serious missing detection. It can be seen from Figures 7–9 that because APFA adopts centroid prediction and energy enhancement compensation of targets, it can effectively solve the problem that TPFA had encountered. The target in scene C walks by the side of the woods. From frames 80 to 89, 190 to 200, 224 to 230, 272 to 356, and 448 to 471, targets in these five different time periods are partially occluded by trees. Figure 7 shows that targets can be with good recovery through prediction and superposition enhancement when targets are obviously partially obscured by trees, so targets can be accurately detected. In scene C, from frames 201 to 223, there are 23 frames in total, during which targets are completely obscured by trees. In scene D, from frames 103 to 123, the target is completely submerged in the highlighted background. As shown in Figure 8, in this case, targets still can be estimated and detected through prediction and superposition enhancement.

In scene C, from frames 224 to 226, the targets are just beginning to come out from the trees. During the process from the time when the target was just completely covered by trees to the time when the target gradually came out

from the trees, the accuracy of target estimation is mainly determined by the prediction accuracy of the target centroid. Before the target started to come out from the trees, there are 23 consecutive frames without the target, which can cause the prediction error of the target centroid. It can be seen from the third column in Figure 9 that the prediction position of the target is lagging behind the real position when the target is slightly emerging from the trees at the beginning, so that the estimated target is located on the right side of the partially exposed target (the target is walking from the right to the left in scene C), and at this time, the expansion of the target is relatively serious. As the real target continuously emerges from behind the trees, the prediction of the target position is more and more accurate, and the target detection result is more and more accurate too. Experimental data show that APFA can detect the target well, whether the target is partially or completely obscured.

**3.3. Target Detection Experiments.** Target detection rate is used to evaluate the detection effect. The target detection rate is defined as follows:



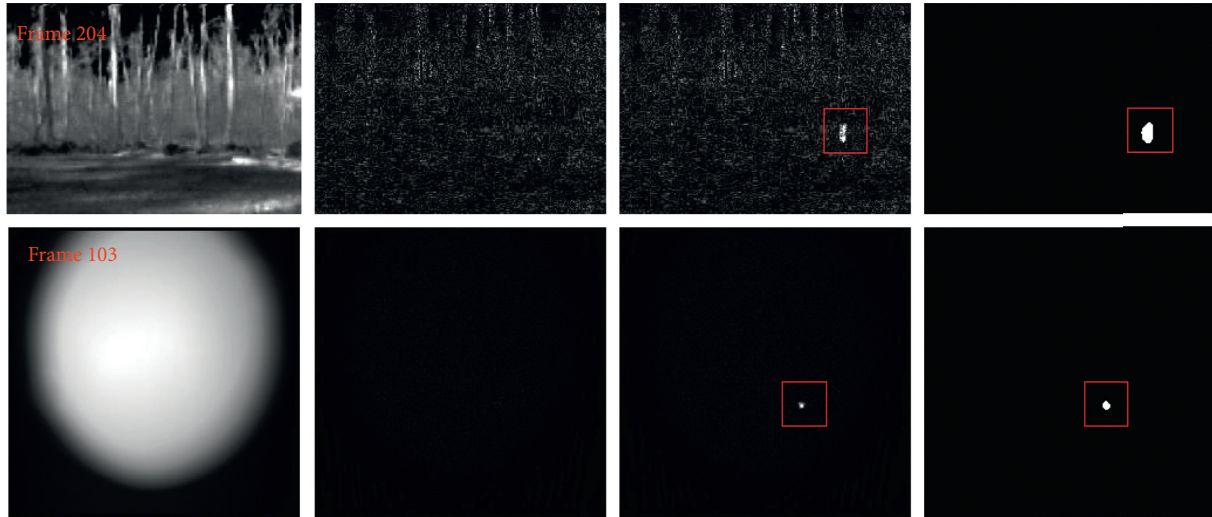


FIGURE 8: Targets are completely obscured by trees or is submerged in the highlight background. The first row corresponds to scene C, frame 204, and from left to right, the four images correspond to the original video image, original difference image, image of superimposed 4 frames, and the result image of target detection of superimposed 4 frames. The second row corresponds to scene D, frame 103, and from left to right, the four images correspond to the original video image, original difference image, image of superimposed 2 frames, and the result image of target detection of superimposed 2 frames.

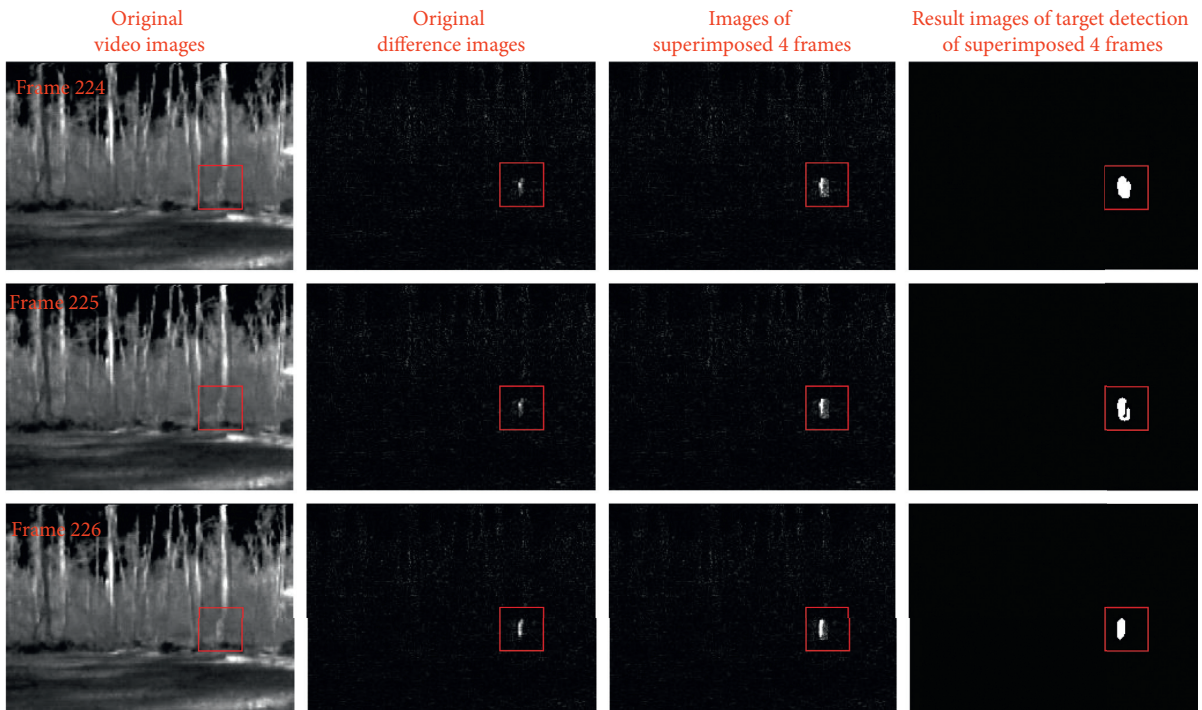


FIGURE 9: The target is gradually emerging from the behind of trees. The first to third rows correspond to the frames 224, 225, and 226, respectively. The first to fourth columns correspond to the original video images, original difference images, images of superimposed 4 frames, and the result images of target detection of superimposed 4 frames, respectively.

$$\text{det\_rate} = \frac{\text{target\_det}}{\text{target\_all}}, \quad (16)$$

where  $\text{det\_rate}$  is the target detection rate,  $\text{target\_det}$  is the total number of frames from which the real target has been detected, and  $\text{target\_all}$  is the total number of frames that have the real target in them. Table 4 gives the target detection

rate data of scenes A, B, C, D, and E obtained by TPFA and APFA. Figure 10 shows the targets' trajectory detected by the two algorithms.

Scene A has a total of 110 frames, and each frame has two targets. In each frame, target 1 is at the top right of the image and target 2 is below target 1. For target 1, the TPFA failed to detect it in frames 15 to 23, 92 to 97, and 102 to 109. So, the



TABLE 4: Comparison of target detection rates.

Scenes	Detection rates of TPFA	Detection rates of APFA
<i>A</i>	79.09 (target 1) 92.73 (target 2)	100% (target 1) 100% (target 2)
<i>B</i>	78.89%	98.77%
<i>C</i>	94.98%	99.60%
<i>D</i>	97.60%	100%
<i>E</i>	61.74%	100%

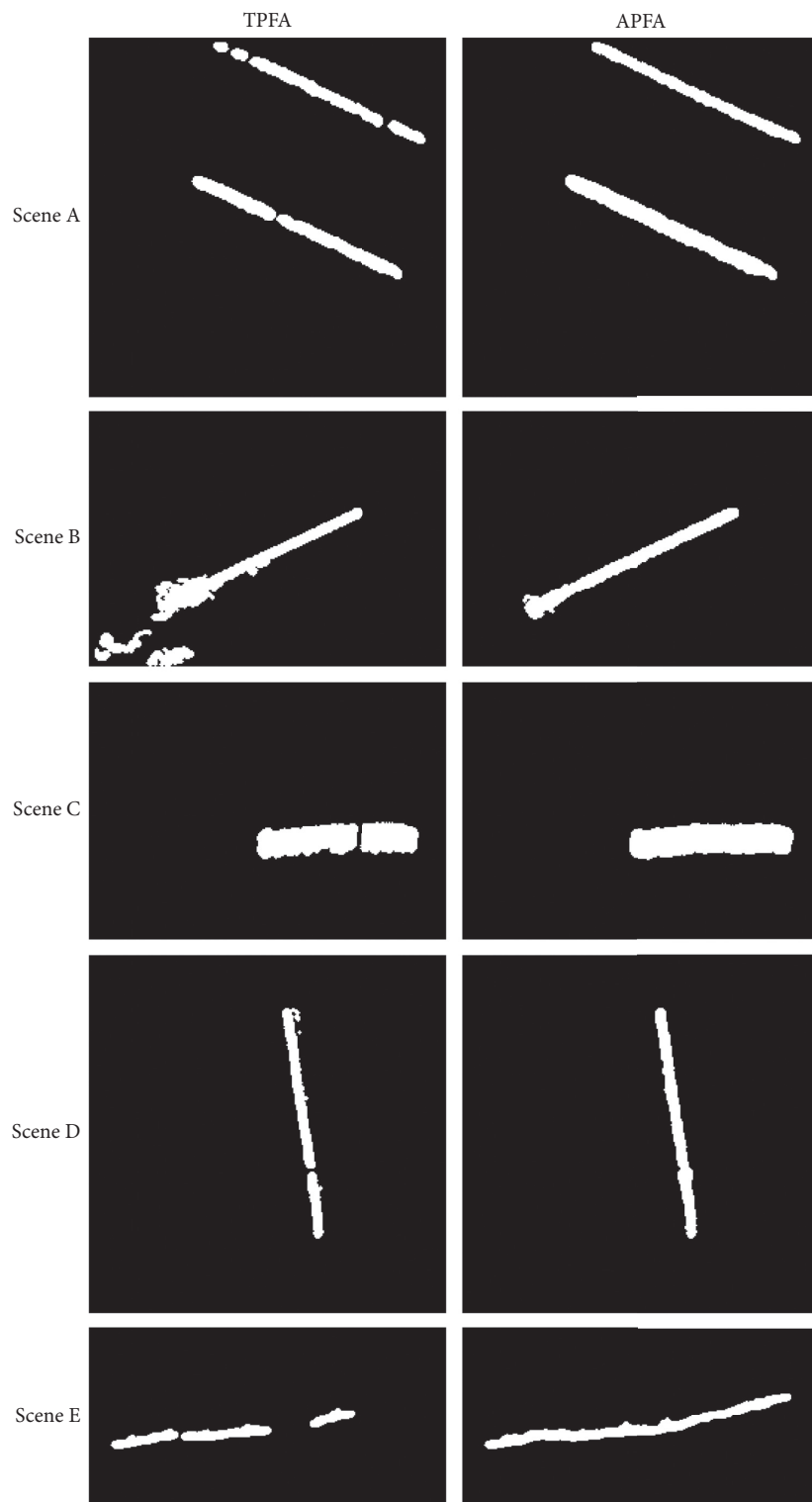


FIGURE 10: Target trajectories. The first to fifth rows correspond to scenes *A*, *B*, *C*, *D*, and *E*, respectively. The target trajectories in the first column were obtained by TPFA, and the target trajectories in the second column were obtained by APFA.

TABLE 5: The time consumptions of APFA for scenes A, B, C, D, and E.

Scenes	Total frames	Image size	Total time	Average time per frame
A	110	250 * 250	23.0560	0.2096
B	532	250 * 180	201.2024	0.3782
C	521	250 * 180	160.9369	0.3089
D	399	256 * 256	108.9669	0.2731
E	115	300 * 150	33.5340	0.2916

The time unit is second.

target detection rate of TPFA for target 1 is 79.09%. For target 2, the TPFA failed to detect it in frames 64 to 71. So, the target detection rate of TPFA for target 2 is 92.73%. As a result, for TPFA, in Figure 10, it can be clearly seen that there are 3 breakpoints on the detected trajectory of target 1, and 1 breakpoint on the detection trajectory of target 2. For the two targets in scene A, the detection rate of APFA is 100%.

Scene B has a total of 532 frames. In six periods of time, a total of 488 frames have the real target, and they are frames 1 to 410, 414 to 436, 441 to 469, 498 to 505, 512 to 521, and 524 to 532. The contrast between targets and their neighborhood is relatively high for frames 1 to 331, so APFA can detect all targets of these frames. However, TPFA failed to detect targets in frames 161, 165, and 190 because the target in frame 161 is blocked by the fixed vertical bright line and targets in frames 165 and 190 are interfered by fixed strong noises. After frame 332, the target flies deeper and closer to the atmosphere above the cloud, and the atmospheric turbulence noises become more and more complex. The target is almost drowned by the turbulence noises, and the target is sometimes vanished. Therefore, after frame 332, the detection performance of TPFA is extremely unstable, and only a total of 57 frames can effectively detected out of the total target. So, for the total 488 frames those containing the target in scene B, 385 frames can be detected by TPFA. As a result, for scene B, the detection rate of TPFA is 78.89%. It can also be seen from Figure 10 that the target trajectory was obtained by TPFA before frame 331 is smooth except that there are two protrusions at the lower edge due to strong noise interference in frames 165 and 190. However, the trajectory after frame 332 becomes divergent, coarse, and fracture, which directly reflects that the detection performance of TPFA is extremely unstable. In sharp contrast, even in the case of strong noise interference and the absence of the target, the APFA can still achieve a good detection result after frame 332. For APFA, in those periods that there is a target, only 6 frames, i.e., frames 371, 377, 378, 384, 513, and 514, cannot detect the target. So, for the total 488 frames those containing the target in scene B, 482 frames can be detected by APFA. As a result, for scene B, the detection rate of APFA is 98.77%. It can be clearly seen from Figure 10 that the overall detection performance of APFA is much stabler than that of the TPFA, that is, the target trajectory obtained by APFA is more continuous and smooth than that of the TPFA.

Scene C has a total of 521 frames. Except the target in frames 201 to 223 is completely obscured by trees, the rest of the 498 frames all have the target in them. In all frames where there is the target in them, a total of 25 frames, i.e.,

frames 56, 200, 224 to 227, 306 to 310, 338 to 343, 487, 488, 496 to 498, and 519 to 521, failed to detect the target by TPFA. As a result, for scene C, the target detection rate of TPFA is 94.98%. For TPFA, it can be seen from Figure 10 that there is an obvious fracture in the target trajectory because of failure in detecting the target in frames 201 to 227, and the smoothness of the target trajectory is relatively poor due to failure in detecting the complete target at each period when the target is partly obscured by trees. In sharp contrast, only frames 224 and 225 cannot be accurately detected by APFA because the target is severely extended, and all the rest of the frames can be detected. So, the target detection rate of APFA is 99.60%. Because APFA can well estimate the target when the target is partially or completely obscured by trees, it can be seen from Figure 10 that the target trajectory that was obtained by APFA is much smoother than that was obtained by the TPFA.

Scene D has a total of 399 frames. Except the target in frames 103 to 123 which is completely submerged in the highlight background, the rest of the 375 frames all have the target in them. In all frames where there is the target in them, a total of 9 frames, i.e., frames 92, 256, 300 to 303, 349, 377, and 391, failed to detect the target by TPFA. As a result, for scene D, the target detection rate of TPFA is 97.60%. For TPFA, it can be seen from Figure 10 that there is an obvious fracture in the target trajectory because of failure in detecting the target in frames 103 to 123. In sharp contrast, not only APFA can detect all the targets in Scene D, but also can estimate the target when it is completely submerged in the highlight background. So, the target detection rate of APFA is 100%, and it can be seen from Figure 10 that the target trajectory obtained by APFA is much smoother than that was obtained by the TPFA.

Scene E has a total of 115 frames, and all have the target in them. Because of the influence of the noises, a total of 44 frames that are frames 27 to 33, 67 to 83, and 96 to 115 failed to detect the target by TPFA. Thus, for scene E, the target detection rate of TPFA is 61.74%, and it can be seen from Figure 10 that there are obvious fractures in the target trajectory that was obtained by TPFA. In sharp contrast, because APFA has the robust performance to detect the target, the target detection rate of APFA is 100% and it can be seen from Figure 10 that the target trajectory that was obtained by APFA is much smoother than that was obtained by the TPFA.

All these experiments were carried out on a PC with the Windows7 operating system, Intel Core i5 CPU, 3G memory, and Matlab R2017b. Then, the time consumptions of APFA for scenes A, B, C, D, and E are given in Table 5.

## 4. Conclusion

Through effectively enhancing the energy and compensating the information of the target and adaptive updating of the center and radius of the pipeline, the APFA can well improve the robustness of the pipeline filter against strong noises and temporary partial or complete occlusion of the target. As a result, the target detection rate of APFA is significantly higher than that of the TPFA. The experimental data of time-domain energy enhancement of targets of scene B show that the signal-to-noise ratio gains of targets are increased more than 2 dB after superimposed 2 and 8 frames, and the average gray values of the targets are greatly promoted. In terms of robust performance, experiments also show that although targets in many frames are overwhelmed by turbulence strong noises such as those after frame 332 in scene B, APFA can still steadily and effectively detect the target until the last frame. Even, in scene C, there are a total of 23 frames where the target is completely obscured by trees and a total of 137 frames where the target is partially obscured by trees, and in scene D, there are a total of 21 frames where the target is completely submerged in the highlight background; however, in these cases, APFA can still steadily and effectively estimate the target. For the five scenes A, B, C, D, and E, the target detection rates of APFA are 100%, 98.77%, 99.60%, 100%, and 100%, respectively, which are significantly higher than those of TPFA.

The main shortcomings of APFA which are inevitable are the prediction error and target expansion, which will affect the stability of the algorithm in serious cases.

## Data Availability

The data used to support the findings of this study are included within the article.

## Conflicts of Interest

The authors declare that they have no conflicts of interest related to this work.

## Acknowledgments

This work was partly supported by the West Light Foundation of the Chinese Academy of Sciences (ya18k001), the Guangxi Science and Technology Base and Talent Project (acceptance no. 2019AC20147), and the doctoral fund of the Guangxi University of Science and Technology (no. 19Z31).

## References

- [1] D. Korchev, H. Kwon, and Y. Owechko, "Low-contrast moving targets in infrared video produced by inconsistent sensor with bad pixels," *Optical Engineering*, vol. 54, no. 11, 2015.
- [2] X. Fan, Z. Xu, J. Zhang, Y. Huang, and Z. Peng, "Infrared dim and small targets detection method based on local energy center of sequential image," *Mathematical Problems in Engineering*, vol. 2017, Article ID 4572147, 16 pages, 2017.
- [3] G. Wang, R. M. Inigo, and E. S. Mcvey, "Pipeline algorithm for detection and tracking of pixel-sized target trajectories," *Signal and Data Processing of Small Targets*, vol. 1305, no. 1, pp. 167–178, 1990.
- [4] L. Zhao and H. Yang, "Small-target leak detection for a closed vessel via infrared image sequences," *Infrared Physics & Technology*, vol. 81, pp. 109–116, 2017.
- [5] C. Wang, Y. Shen, D. Zhang, and Y. Cai, "A dynamic queue based pipeline filter for infrared dim small target detection," in *Proceedings of the 34th Chinese Control Conference*, pp. 3770–3775, Hangzhou, China, July 2015.
- [6] H. Jiang, W. Yi, G. Cui, L. Kong, and X. Yang, "Track-before-detect strategies for range distributed target detection in compound-Gaussian clutter," *Signal Processing*, vol. 120, pp. 462–467, 2016.
- [7] S. Wu, Q. Xue, S. Zhu, Q. Yan, and X. Sun, "Track-before-detect algorithm for weak extended target based on particle filter under clutter environment," *Journal of Radars*, vol. 6, no. 3, pp. 252–258, 2017.
- [8] B. Yan, L. P. Xu, Y. Yang, and C. Li, "Improved plot fusion method for dynamic programming based track before detect algorithm," *AEU-International Journal of Electronics and Communications*, vol. 74, pp. 31–43, 2017.
- [9] J. Liu and H. Ji, "Infrared small targets detection based on mobile weighted pipeline filter," *Journal of Xidian University*, vol. 34, no. 5, pp. 743–747, 2007.
- [10] G. Liu and X. Liang, "Detection of aerial small target in infrared image based on wavelet transform and pipeline filter," *Computer Engineering and Applications*, vol. 47, no. 30, pp. 198–201, 2011.
- [11] W. Dong, J. Zhang, D. Liu, and X. Wang, "Pipeline filter algorithm based on movement direction estimation," *Acta Photonica Sinica*, vol. 42, no. 4, pp. 471–474, 2013.
- [12] X. Zhao, S. Yuan, X. Ma, and L. Qi, "Research on infrared small target detection technique based on moving pipeline filtering," *Infrared Technology*, vol. 31, no. 5, pp. 295–297, 2009.
- [13] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, pp. 1–37, 2011.
- [14] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.
- [15] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM Journal on Optimization*, vol. 21, no. 2, pp. 572–596, 2011.
- [16] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," <http://arxiv.org/abs/1009.5055>.
- [17] X. Yuan and J. Yang, "Sparse and low-rank matrix decomposition via alternating direction method," *Pacific Journal Of Optimization*, vol. 9, no. 1, pp. 167–180, 2013.
- [18] X. Huang and Y. Wang, *Principle and Application of Kalman Filter—Matlab Simulation*, Publishing House of Electronics Industry, Beijing, China, 2017.