Journal of Imaging Science and Technology® 66(1): 010503-1–010503-19, 2022.

Detection of Air-to-Air Flying Targets against Sky–ground Joint Background

Junhua Yan, Kun Zhang, Yin Zhang, Xuyang Cai, Jingchun Qi, and Junjie Fan
Key Laboratory of Space Photoelectric Detection and Perception (Nanjing University of Aeronautics and Astronautics), Ministry of Industry and Information Technology, Nanjing, Jiangsu 211106, China
Nanjing University of Aeronautics and Astronautics, College of Astronautics, Nanjing, Jiangsu 211106, China
E-mail: yjh9758@nuaa.edu.cn

Abstract. *There are three critical problems that need to be tackled in target detection when both the target and the photodetector platform are in flight. First, the background is a sky–ground joint background. Second, the background motion is slow when detecting targets from a long distance, and the targets are small, lacking shape information as well as large in number. Third, when approaching the target, the photodetector platform follows the target in violent movements and the background moves fast. This article is comprised of three parts. The first part is the sky–ground joint background separation algorithm, which extracts the boundary between the sky background and the ground background based on their different characteristics. The second part is the algorithm for the detection of small flying targets against the slow moving background (DSFT-SMB), where the double Gaussian background model is used to extract the target pixel points, then the missed targets are supplemented by correlating target trajectories, and the false alarm targets are filtered out using trajectory features. The third part is the algorithm for the detection of flying targets against the fast moving background (DFT-FMB), where the spectral residual model of target is used to extract the target pixel points for the target feature point optical flow, then the speed of target feature point optical flow is calculated in the sky background and the ground background respectively, thereby targets are detected using the density clustering algorithm. Experimental results show that the proposed algorithms exhibit excellent detection performance, with the recall rate higher than 94%, the precision rate higher than 84%, and the F-measure higher than 89% in the DSFT-SMB, and the recall rate higher than 77%, the precision rate higher than 55%, and the F-measure higher than 65% in the DFT-FMB.*
[DOI: 10.2352/J.ImagingSci.Technol.2022.66.1.010503]

1. INTRODUCTION

This article focuses on target detection under the condition that both the photodetector platform and targets are in flight, such as Unmanned Aerial Vehicle (UAV) in flight, where the sky and the ground can simultaneously appear at the background in the video image. In order to detect targets from a long distance, the photodetector platform needs to detect targets in a large range of area. The background movement in the image is slow. Because the targets are far away from the detector at this time, they occupy a small number of pixels in the image, which makes them small and lacks morphological information. Besides, a large detection area increases the number of interfering targets, such as a large number of birds. Once the photodetector platform detected the target, it will approach and fly close to the target. In doing so, the target maneuver widely and move at a high speed, making the photodetector platform, which is flying close by to follow the target in violent motions, in order to continuously detect the target. Under this condition, the background movement in the image is fast. This article focuses on the detection of air-to-air flying targets against sky–ground joint background. The algorithm includes three parts, including the separation algorithm for the sky–ground joint background, the detection algorithm for small flying targets against slow moving background (DSFT-SMB), and the detection algorithm for flying targets against fast moving background (DFT-FMB).

The most difficult problem for the detection of air–air flying targets against the sky–ground joint background is that the position of the target in the captured image constantly changes, which is caused by the target flying freely in the entire scene and the airborne photodetector platform. The target can be captured in the sky background at a certain time, and at another time, in the ground background, or in transition between the two. To address this problem, researchers have proposed some detection algorithms in recent years.

In 2008, Li et al. [1] proposed an algorithm for detecting and tracking small flying objects against sky background based on image enhancement and motion fitting. This algorithm only uses image threshold segmentation for the target detection, which performs well when the background is only the sky. However, the detection effect is poor when the image contains ground background.

In 2012, Siam et al. [2] extracted Features from Accelerated Segment Test (FAST) corner optical flows for moving targets, calculated global motion parameters to separate out the feature points of the target, and detected moving targets by the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) clustering algorithm. The algorithm has a good detection effect against the sky background but encounters difficulty in extracting the feature corners of small targets against the ground background, rendering it susceptible to missed detection and

Received Feb. 7, 2021; accepted for publication May 30, 2021; published online July 29, 2021. Associate Editor: Steven Simske.
1062-3701/2022/66(1)/010503/19/$25.00

causing poor detection results. Kovács et al. [3] proposed a flying target detection algorithm based on feature corners and edge detection combined with robust background modeling. First, they extracted the corners and edges of the boundary of the flying target, and thereby established a spatiotemporal multimode background model to detect the outline of the target. Subsequently, they used the background and the target appearance models, as well as additional interaction constraints to refine the detection of flying targets in the Markov Random Field (MRF) framework. However, when the target is small or is flying against the ground background, it is difficult for the algorithm to extract the feature edges of the target and the detection effect is poor.

In 2013, Yi et al. [4] proposed a pixel-based piecewise dual-mode single Gaussian background modeling algorithm, which detected moving targets based on compensation for background motion by mixing adjacent models. The algorithm can detect small targets, but the false alarm rate is high for complex backgrounds.

In 2014, Sadeghi-Tehran et al. [5] extracted Binary Robust Invariant Scalable Keypoints (BRISK) [6] feature point optical flows for the moving targets in the images, calculated global motion parameters to separate out feature points of the targets, and used the Evolving Local Means (ELM) algorithm to cluster the feature point optical flows of the target, thereby detecting the moving targets. The algorithm easily misses small targets against ground background, and the detection effect is poor. Xu et al. [7] proposed a bird detection algorithm based on simplified bird skeleton descriptors, as most birds have a similar skeletal structure that are distinctive from other objects. They extracted the interested target areas for effective preprocessing, and then calculated the simplified bird skeleton descriptors to train the linear Support Vector Machine (SVM) classifier, hence realizing the detection of bird targets. This algorithm cannot adapt to the changing posture of freely flying birds, and finds it difficult to detect small targets.

In 2015, Rozantsev et al. [8] used a boosted trees algorithm for motion compensation, which involved using a multi-scale sliding window to extract space-time cubes on multiple successive frames, and then using the AdaBoost classifier to detect the drone targets in the space-time cube. In 2017, Rozantsev et al. [9] improved boosted trees to Convolution Neural Networks (CNNs), which can better adapt to the complex backgrounds. However, it is difficult for the algorithm to obtain accurate target detection results when the flying targets are small. In 2019, on the basis of this algorithm, Wang [10] proposed the Relevant Bounding Box Adjustment Based on Spatiotemporal Cube (ST-RBBA) algorithm to make better use of the spatiotemporal information of the spatiotemporal cube. However, it is also difficult for this algorithm to obtain accurate detection results when the targets are too small.

In 2016, Poiesi et al. [11] estimated the optical flow to compensate for global camera motion, established an adaptively learned background motion model, and eliminated dynamic texture noise in the image, thereby accurately detecting the drone flying into the field of view. This algorithm only estimates optical flow, so it has poor detection limitation on scenes with small targets and insignificant optical flow.

In 2017, Lou et al. [12] proposed a detection algorithm for small targets combining the stability and saliency of the target area, which obtained a stability map from a set of locally stable areas derived from continuous Boolean graphs, obtained a saliency map by comparing the color vector of each pixel with its Gaussian blurred version, and integrated both the stability and the saliency maps in a pixel-wise multiplication manner to detect plane targets in the sky and vehicle targets on the ground. However, when targets are in a complex ground background, the algorithm has many false alarms and undetected targets, and the detection accuracy is poor. Schumann et al. [13] proposed a flying drone target detection algorithm based on video images. Median background subtraction is employed in the case of using a static camera, and Region Proposal Network (RPN) is employed in the case of using a motion camera to obtain candidate target regions, then the UAV and bird targets are detected by a CNN classifier. However, the algorithm has poor detection effects on small targets. Yoshihashi et al. [14] proposed an algorithm for detecting small flying objects based on motion information. This algorithm uses Convolutional Long Short-Term Memory (ConvLSTM) [15] to learn multi-frame target motion features for detection and can effectively detect and track targets in the video by using target motion information. The algorithm extracts candidate target regions based on the 3D Histograms of Oriented Gradients (HOG3D) sliding window detector, which inevitably has certain requirements on the target's scale, so the target detection effect on small targets needs to be further improved.

In 2018, Zhang et al. [16] proposed a flying target detection algorithm based on spatial–temporal context. They extracted temporal context information by calculating forward–backward motion history map from multi-frame video sequences, extracted the spatial context through conditional random fields, and detected the flying targets by fusing temporal and spatial context information. However, the algorithm has poor detection effect for small targets. Chen et al. [17] proposed a bird detection algorithm based on the observation that flying birds fly fast in a wide range, and the background of the image usually moves slow in a small range. In this algorithm, a background model is constructed by processing the 3×3 sub-image blocks, and the bird targets were detected by the pixel difference of the sub-image blocks. However, the algorithm has a poor detection effect in the case of fast background motion. Janousek et al. [18] proposed a flying target detection algorithm. They detected horizontal lines in an image based on filtering and Hough transform, and then extracted the Oriented FAST and Rotated BRIEF (ORB) corner areas in the sky background, and detected targets by testing the features and edges in these areas. The algorithm can only detect targets against the sky background.

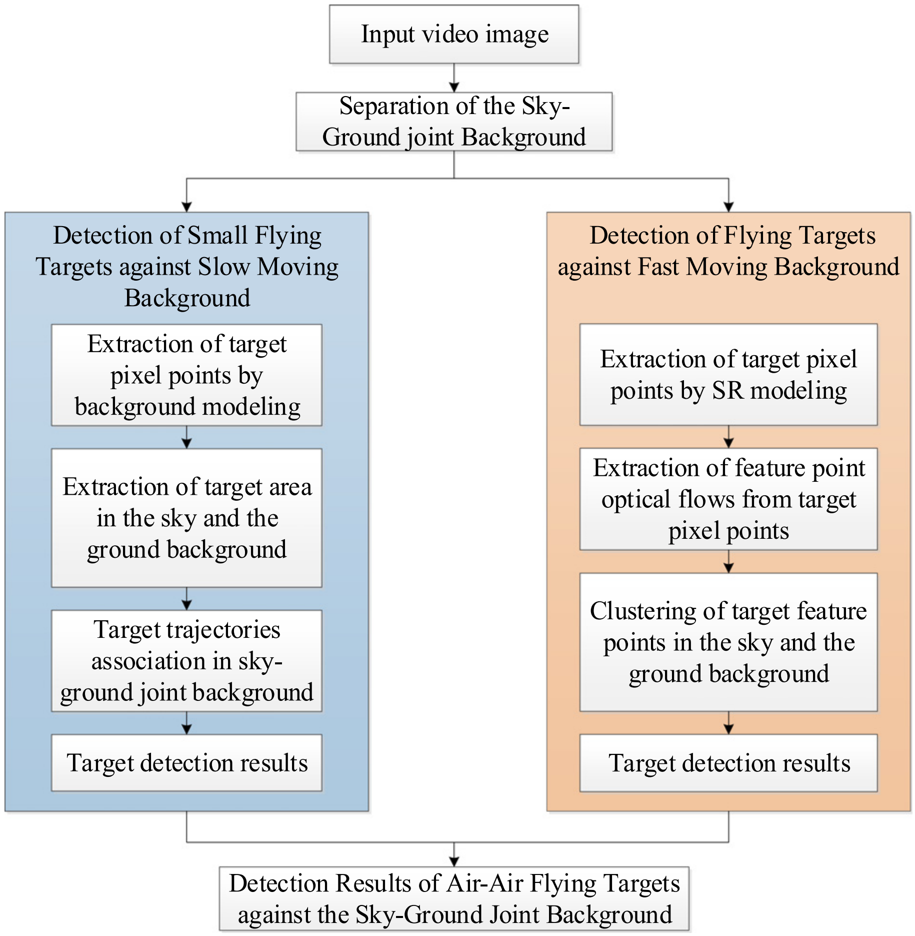


Figure 1. Block diagram of the proposed detection algorithm for air–air flying targets against sky–ground joint background.

When the targets are small or in ground backgrounds, this algorithm finds it difficult to detect the targets.

Deeming the above-mentioned literature as the foundation, this article focuses on the detection of air-to-air flying targets against the sky–ground joint background. The block diagram of the proposed detection algorithm is shown in Figure 1, including the separation algorithm for the sky–ground joint background, the detection algorithm for small flying targets against the slow moving background, and the detection algorithm for flying targets against the fast moving background.

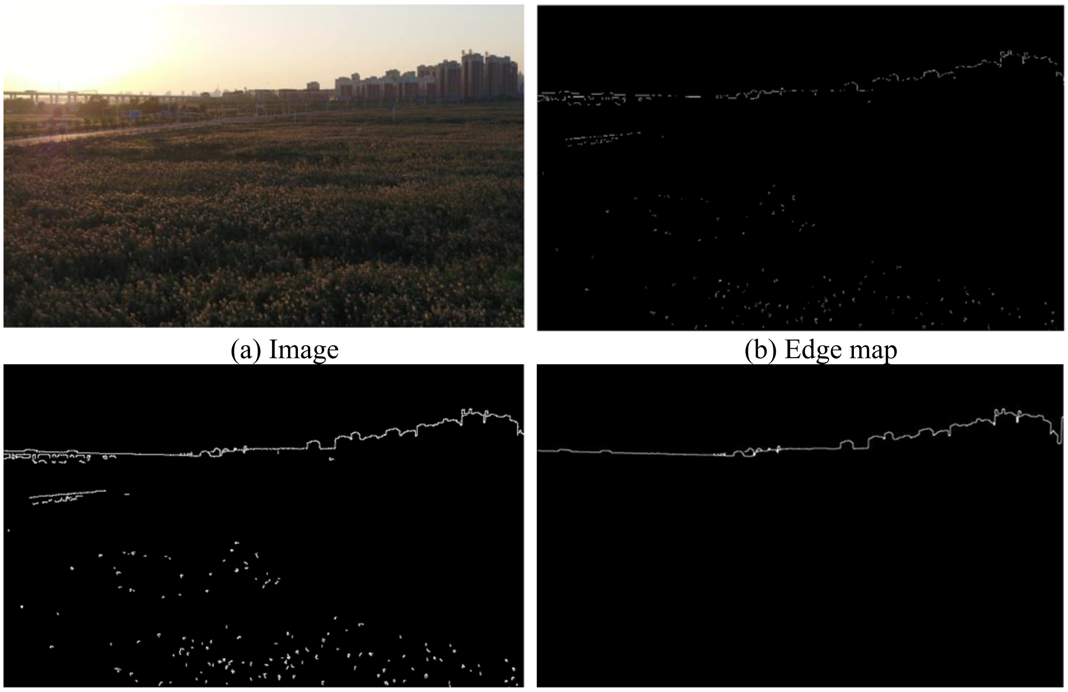
First, the boundary between the sky background and the ground background is extracted, in order to separate the two.

Subsequently, different detection algorithms are proposed for two different situations, which are the detection algorithm for small flying targets against the slow moving background and the detection algorithm for flying targets against the fast moving background. For the detection of small flying targets against the slow moving background, background modeling is first established to extract the target pixel points. Subsequently, the target pixel points are detected in the sky and the ground background, respectively, and the target area is extracted. Finally, trajectory association is carried out on the targets in the sky–ground joint background, and missed targets are supplemented based on the target trajectory, and false alarm targets are filtered out based on the trajectory feature, thereby the target detection results are obtained. For the detection of flying targets against the fast moving background, first, the Spectral Residual (SR) model is established to extract the target pixel points. Subsequently, the target feature points in the target pixel points are extracted to calculate the optical flow of the target feature points. Finally, the target feature point optical flow speed is calculated in the sky and the ground backgrounds, respectively, and the target feature points are clustered according to their optical flow speeds, that is, those with similar optical flow speeds are clustered into a target, thereby the target detection results are obtained.

Lastly, the detection results of air–air flying targets against the sky–ground joint background are obtained by combining the detection results of small flying targets against the slow moving background and that of flying targets against the fast moving background.

2. SEPARATION OF THE SKY–GROUND JOINT BACKGROUND

The sky–ground joint background has both the sky background and the ground background, each with distinctive



(a) Image (b) Edge map

(c) Edge line connection (d) Boundary line between the sky and the ground background

Figure 2. Separation of the sky–ground joint background.

features. The distinguishability between target features and background features is different for these two types of background. Therefore, it is necessary to separate the two backgrounds and develop different target detection algorithms according to the location of the target in the background. Because the features of the sky background and the ground background are different, there is a clear dividing line between the two, and by extracting this line, the two backgrounds can be separated.

The surface features where the sky meets the ground in the image are usually undulating, and the sky–ground boundary is not a straight line. The gray value of the pixels at the edge of the boundary line changes rapidly, that is, the gradient is the largest at these edges. Therefore, edge detection is first performed using the Canny operator [19], because it has excellent anti-noise performance and high edge detection accuracy. Then the extracted edges are morphologically processed to connect the edge lines to obtain the complete boundary between the sky and the ground background. The detailed steps are listed as follows:

1. Multiple edge lines are obtained by Canny edge detection, and multiple interference edges in the sky and the ground background are also extracted, as shown in Figure 2(b).
2. The morphological expansion operator is used to process each edge line in the edge map to connect the neighboring edge lines together, as shown in Fig. 2(c).
3. Calculate the length of each edge line in Fig. 2(c), in which the longest edge line is the boundary between the sky background and the ground background. In order to filter the interference edge lines in the sky background and the ground background, the length of the boundary line is set as the threshold, and the edge lines whose length is less than the threshold are removed, as shown in Fig. 2(d).

3. DETECTION OF SMALL FLYING TARGETS AGAINST SLOW MOVING BACKGROUND

The flow chart of the detection algorithm for small flying targets against slow moving background (DSFT-SMB) is shown in Figure 3.

First, target pixel points are extracted from the sky–ground joint background. Global background motion estimation is performed on the image to calculate the background motion parameters. The image grid is regionalized, and a double Gaussian background model is established for each grid area. Global background motion parameter compensation is performed on the background model before the background model parameters are compensated, and finally, the target pixel points in the sky–ground joint background are extracted based on the background model. Meanwhile, the sky–ground joint background is separated into the sky background and the ground background.

Subsequently, the target area map in the sky background and that in the ground background are combined to obtain the target area map in the sky–ground joint background. In detail, in the sky background, the morphological expansion operator is used to enhance the target pixel points to obtain

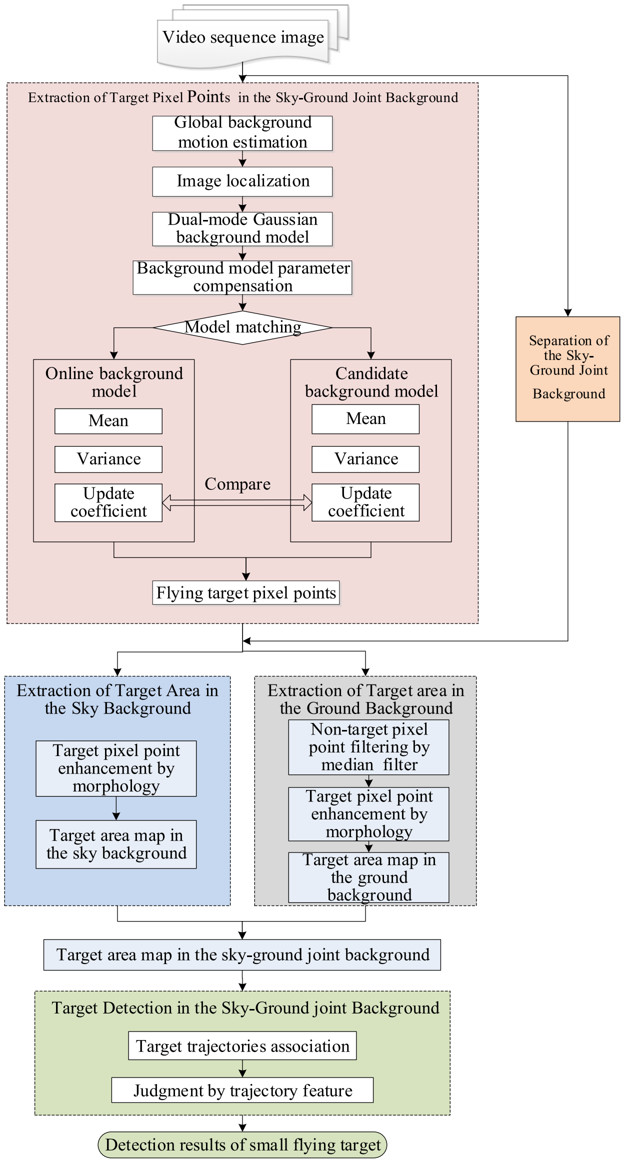


Figure 3. Flow chart for the detection of small flying targets against the slow moving background.

the target area map, while in the ground background, the median filtering algorithm is used to filter out the non-target pixel points, and the morphological expansion operator is used to enhance the target pixel points to obtain the target area map in the ground background. Combining the target area map in the sky background and that in the ground background, the target area map in the sky–ground joint background is obtained.

Finally, targets are detected in the sky–ground joint background. The targets in the target area map in the sky–ground joint background are associated to obtain the target trajectory, based on which the targets missed by detection are supplemented to improve the target recall rate, and judging by the trajectory features, false alarm targets are filtered out to improve the target precision rate, thereby realizing the detection of the small flying targets in the sky–ground joint background.

3.1 *Extraction of Target Pixel Points against the Sky–Ground Joint Background*

When the photodetector platform detects targets at a long distance and a large range, the background in the video image includes both the sky background and the ground background. The background movement of the image is slow, and the UAV and birds in flight occupy a small number of pixels in the image, which becomes a small target in the image and lacks topographical information. Therefore, pixel points of small flying targets in the sky–ground joint background are extracted based on the pixel-level background modeling algorithm.

3.1.1 *Global Background Motion Estimation*

In this article, regional random points and the Lucas–Kanade (LK) optical flow method [20] are used to obtain random optical flow tracking points. Regional random points can well represent regional characteristics and do not require much gradient calculations. Regional random points are evenly extracted from the image $I(t-1)$, and corresponding random optical flow tracking points are obtained from the image $I(t)$ using the LK optical flow method. The Random Sample Consensus (RANSAC) algorithm is used to fit the 8-parameter homography matrix H_{t-1}^{t}, which is the global background motion parameter, as shown in Eq. (1).

$$\begin{bmatrix} x_i^t \\ y_i^t \\ 1 \end{bmatrix} = H_{t-1}^{t} \begin{bmatrix} x_i^{t-1} \\ y_i^{t-1} \\ 1 \end{bmatrix}, \tag{1}$$

where (x_i^{t-1}, y_i^{t-1}) is the coordinate of the regional random point in the frame $I(t-1)$, and (x_i^t, y_i^t) is the coordinate of the corresponding optical flow tracking point in the frame $I(t)$.

3.1.2 *Background Modeling to Extract Target Pixel Points*

This article uses the Dual-mode Gaussian Model (DGM) [4] for background modeling. Global background motion parameter compensation is performed on the background model, and background model parameter compensation is performed on the motion-compensated background model, based on which background model parameters are updated. Whether the pixel points in the image are the target pixel points or the background pixel points is determined according to the background model, thereby the target pixel points are extracted.

First, the image $I(t)$ is evenly divided into grid regions of size $L \times L$ and the ith region at time t is represented by $R_{i,t}$. A DGM is applied to each grid region, as shown in Eq. (2).

$$\begin{aligned} \mathrm{DGM}(R_{i,t}) = (1-c) \cdot G^o(R_{i,t}, \mu_{i,t}^o, \sigma_{i,t}^o, a_{i,t}^o) \\ + c \cdot G^c(R_{i,t}, \mu_{i,t}^c, \sigma_{i,t}^c, a_{i,t}^c), \end{aligned} \tag{2}$$

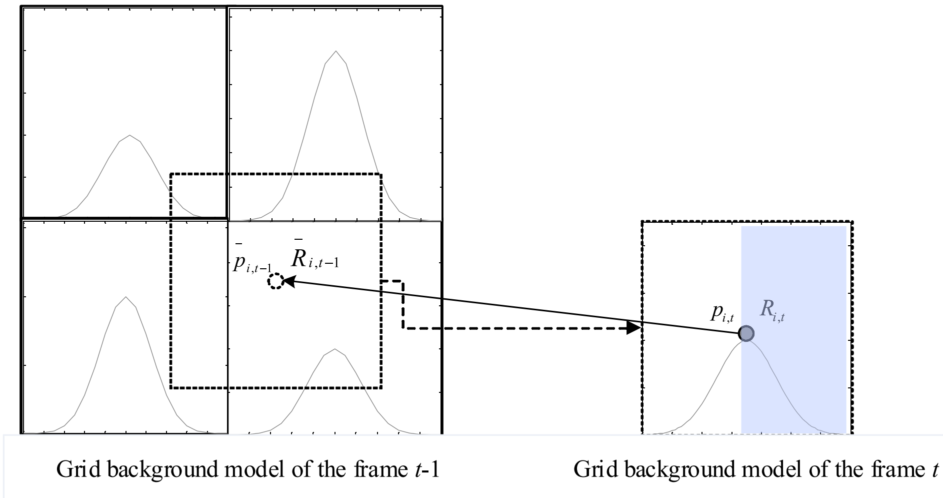


Figure 4. Background model parameter compensation.

where G^o represents an online Gaussian background model and G^c represents a candidate Gaussian background model. $\mu^o_{i,t}$, $\sigma^o_{i,t}$, and $a^o_{i,t}$ are the mean, variance, and update coefficient of the online Gaussian background model G^o, and $\mu^c_{i,t}$, $\sigma^c_{i,t}$, and $a^c_{i,t}$ are the mean, variance, and update coefficient of the candidate Gaussian background model G^c. c is the candidate Gaussian background model coefficient. When the candidate model matches the region, $c = 1$, otherwise $c = 0$, which ensures the most suitable model for region matching.

Next, motion compensation is performed on the background model. The center position of the grid area $R_{i,t}$ is set to $p_{i,t}$, and the grid $R_{i,t}$ in the frame $t-1$ is $\bar{R}_{i,t-1}$, whose center position is $\bar{p}_{i,t-1}$, which is calculated based on the global background motion parameter H^t_{t-1}:

$$\bar{p}_{i,t-1} = (H^t_{t-1})^{-1} p_{i,t}. \tag{3}$$

Background model parameter compensation is performed on the motion-compensated background model $\tilde{R}_{i,t-1}$ to obtain $R_{i,t}$, as shown in Figure 4. $\bar{R}_{i,t-1}$ overlaps with the K grids $R_{k,t-1}$ in the frame $t-1$, and the overlap ratio is ω_k. The online background model parameters $\mu^{o,\mathrm{comp}}_{i,t}$, $\sigma^{o,\mathrm{comp}}_{i,t}$, $a^{o,\mathrm{comp}}_{i,t}$ of the background model $R_{i,t}$ are shown in Eq. (4).

$$\begin{cases} \mu^{o,\mathrm{comp}}_{i,t} = \sum_{k=1}^{K} \omega_k \mu^o_{k,t-1} \\ \sigma^{o,\mathrm{comp}}_{i,t} = \sum_{k=1}^{K} \omega_k (\sigma^o_{k,t-1} + (\mu^o_{k,t-1} - \mu^{o,\mathrm{comp}}_{i,t})^2) \\ a^{o,\mathrm{comp}}_{i,t} = \sum_{k=1}^{K} \omega_k a^o_{k,t-1}. \end{cases} \tag{4}$$

Using the same method as in Eq. (4), candidate background model parameters $\mu^{c,\mathrm{comp}}_{i,t}$, $\sigma^{c,\mathrm{comp}}_{i,t}$, $a^{c,\mathrm{comp}}_{i,t}$ of the background model $R_{i,t}$ are obtained. If the variance of the compensated online background model or the variance of the candidate background model is greater than the threshold θ_σ, whose value is determined to be 2500 by experiment, it is necessary to reduce the update coefficient of the corresponding background model so that the background model converges, as shown in Eq. (5).

$$\begin{cases} a^{o,\mathrm{comp}}_{i,t} = a^{o,\mathrm{comp}}_{i,t} \exp(-\theta_d(\sigma^{o,\mathrm{comp}}_{i,t} - \theta_\sigma)), & \text{if } \sigma^{o,\mathrm{comp}}_{i,t} > \theta_\sigma \\ a^{c,\mathrm{comp}}_{i,t} = a^{c,\mathrm{comp}}_{i,t} \exp(-\theta_d(\sigma^{c,\mathrm{comp}}_{i,t} - \theta_\sigma)), & \text{if } \sigma^{o,\mathrm{comp}}_{i,t} > \theta_\sigma. \end{cases} \tag{5}$$

In Eq. (5), θ_d is the attenuation parameter, whose value is determined to be 0.001 by experiment, which can effectively prevent the variance of background model from being too large.

Subsequently, the background model is updated with the parameters compensated by the background model. According to the matching states of the average value $M_{i,t}$ of the pixels in the grid area $R_{i,t}$ with the online background model G^o or with the candidate background model G^c, corresponding update methods are adopted.

If $M_{i,t}$ matches G^o after compensation, that is $(M_{i,t} - \mu^{o,\mathrm{comp}}_{i,t})^2 < \theta_s \sigma^{o,\mathrm{comp}}_{i,t}$, where θ_s is the threshold parameter, whose value is determined to be 2 by experiment, then the parameters $\mu^o_{i,t}$, $\sigma^o_{i,t}$, $a^o_{i,t}$ of the online background model G^o are updated by the method shown in Eq. (6).

$$\begin{cases} \mu^o_{i,t} = \left(1 - \frac{1}{a^{o,\mathrm{comp}}_{i,t} + 1}\right) \mu^{o,\mathrm{comp}}_{i,t} + \frac{1}{a^{o,\mathrm{comp}}_{i,t} + 1} M_{i,t} \\ \sigma^o_{i,t} = \left(1 - \frac{1}{a^{o,\mathrm{comp}}_{i,t} + 1}\right) \sigma^{o,\mathrm{comp}}_{i,t} + \frac{1}{a^{o,\mathrm{comp}}_{i,t} + 1} V_{i,t} \\ a^o_{i,t} = a^{o,\mathrm{comp}}_{i,t} + 1, \end{cases} \tag{6}$$

where $V_{i,t}$ is the variance of the pixels in the grid area $R_{i,t}$.

If $M_{i,t}$ does not match G^o after compensation, but $M_{i,t}$ matches G^c after compensation, that is, $(M_{i,t} - \mu^{c,\mathrm{comp}}_{i,t})^2 < \theta_s \sigma^{c,\mathrm{comp}}_{i,t}$, where θ_s is the threshold parameter, whose value is set to be 2, same as in the case above, then the parameters $\mu^c_{i,t}$, $\sigma^c_{i,t}$, $a^c_{i,t}$ of the candidate background model G^c are updated in the same way as in Eq. (6).

If $M_{i,t}$ does not match G^o or G^c after compensation, then $M_{i,t}$ and $V_{i,t}$ are used to initialize the candidate background model, that is, $\mu^c_{i,t} = M_{i,t}$, $\sigma^c_{i,t} = V_{i,t}$, $a^c_{i,t} = 1$.

If $a^c_{i,t} > a^o_{i,t}$ occurs after the background model parameters are updated, which means that the update coefficient of the candidate background model G^c of the current area $R_{i,t}$ is larger than that of the online background model G^o, then the parameters of the online background model G^o are replaced by that of the candidate background model G^c, and initialize the parameters of G^c.

Finally, whether the pixel points in the image are the target pixel points or the background pixel points is determined according to the updated background model, as shown in Eq. (7).

$$PI^j_{i,t}(x, y) = \begin{cases} 1 & \text{if } (I^j_{i,t} - \mu^o_{i,t})^2 > \theta_f \sigma^o_{i,t} \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

where θ_f is the threshold parameter, whose value is determined to be 3 with better detection effect by experiment. $I^j_{i,t}$ represents the pixel value of the jth pixel in the current grid area $R_{i,t}$. If $PI^j_{i,t}(x, y)$ is 1, $I^j_{i,t}$ is the pixel value of the target pixel point; if $PI^j_{i,t}(x, y)$ is 0, $I^j_{i,t}$ is the pixel value of the background pixel point. The target pixel point map $PI(t)$ is obtained by traversing all the grid regions of the image $I(t)$.

3.2 *Extraction of Target Area against the Sky–Ground Joint Background*

Because targets in the video image are small and lack topographical information, there are non-target pixel points in the target pixel point map $PI(t)$. Because features of the sky background and the ground background are distinctive, the distinguishability between the target pixel point and the two types of background is different. For the target pixel point map $PI(t)$, the target pixel points are detected in the sky background and the ground background, respectively, in order to improve the target detection accuracy.

For the target pixel point map $PI(t)$, in the sky background, the morphological expansion operator is used to enhance the target pixel points to obtain the target area map $M^S_{\mathrm{REG}}(t)$, whereas in the ground background, the median filtering algorithm is used to filter out the non-target pixel points, and the morphological expansion operator is used to enhance the target pixel points to obtain the target area map $M^G_{\mathrm{REG}}(t)$. The target area map $M^S_{\mathrm{REG}}(t)$ in the sky background and the target area map $M^G_{\mathrm{REG}}(t)$ in the ground background are combined to obtain the target area map $M_{\mathrm{REG}}(t)$ in the sky–ground joint background, which improves the target detection accuracy, as shown in Figure 5.

3.3 *Target Detection against the Sky–Ground Joint Background*

In the sky background, detection of targets in a wide range can lead to a larger number of targets appearing in the field of view, which may appear occluded or adjacent. In the ground background, the contrast between small targets and the background is not obvious. Therefore, there are often some missed detections and false alarms in the target area map M_{REG} in the sky–ground joint background. The Kalman predictor is used to predict every target's position. Based on the Euclidean distance between the detected target position and the predicted target position, the Hungarian matching algorithm is used to correlate targets to obtain multiple target trajectories. If no detected target has the position that matches the predicted target position, which indicates missed detection, then the missed target is supplemented at the predicted target position to improve the recall rate. Because the false alarm target trajectory features are different from the target trajectory features, then the target trajectory features are used to filter out false alarm target trajectories to improve precision rate.

3.3.1 *Target Trajectories Association*

Within the moving target area map $M_{RGE}(t)$, the target's location in the next frame $M_{RGE}(t+1)$ can be predicted. Based on the Euclidean distance between the detected target position and the predicted target position, corresponding targets are associated to form multiple target trajectories.

(1) Predict targets' positions. The eight-connected domain algorithm is used to detect targets and their positions in the $M_{RGE}(t)$, then the Kalman predictor is used to predict every target's position in the next frame $M_{RGE}(t+1)$.
(2) Detect targets. The eight-connected domain algorithm is used to detect targets and their positions in the $M_{RGE}(t+1)$.
(3) Associate targets. In the $M_{RGE}(t{+}1)$, the detected position and the predicted position of every target is matched. If the detected position of the target matches the predicted position, they are associated as the same target. If the predicted position of the target matches none of the detected positions, which indicates missed detection, the missed target is supplemented at the predicted position in the $M_{RGE}(t+1)$. If the detected position of the target matches none of the predicted positions, then this indicates that a new target appears.
(4) Determine target trajectory. Target trajectories are determined from all the associated targets. If a target in one trajectory is supplemented at the predicted position for five consecutive frames, the target is considered to have disappeared, and the trajectory will be deleted.

3.3.2 *Extraction of Target Trajectory Features*

Trajectory features of the real target are different from those of the false alarm target. Hence, the trajectory features can be used to filter out the false alarm target trajectories.

The target trajectory features include target area feature A, target position feature V, and target movement direction feature D. The mean square error of the target area in the target trajectory is calculated to obtain the target area feature

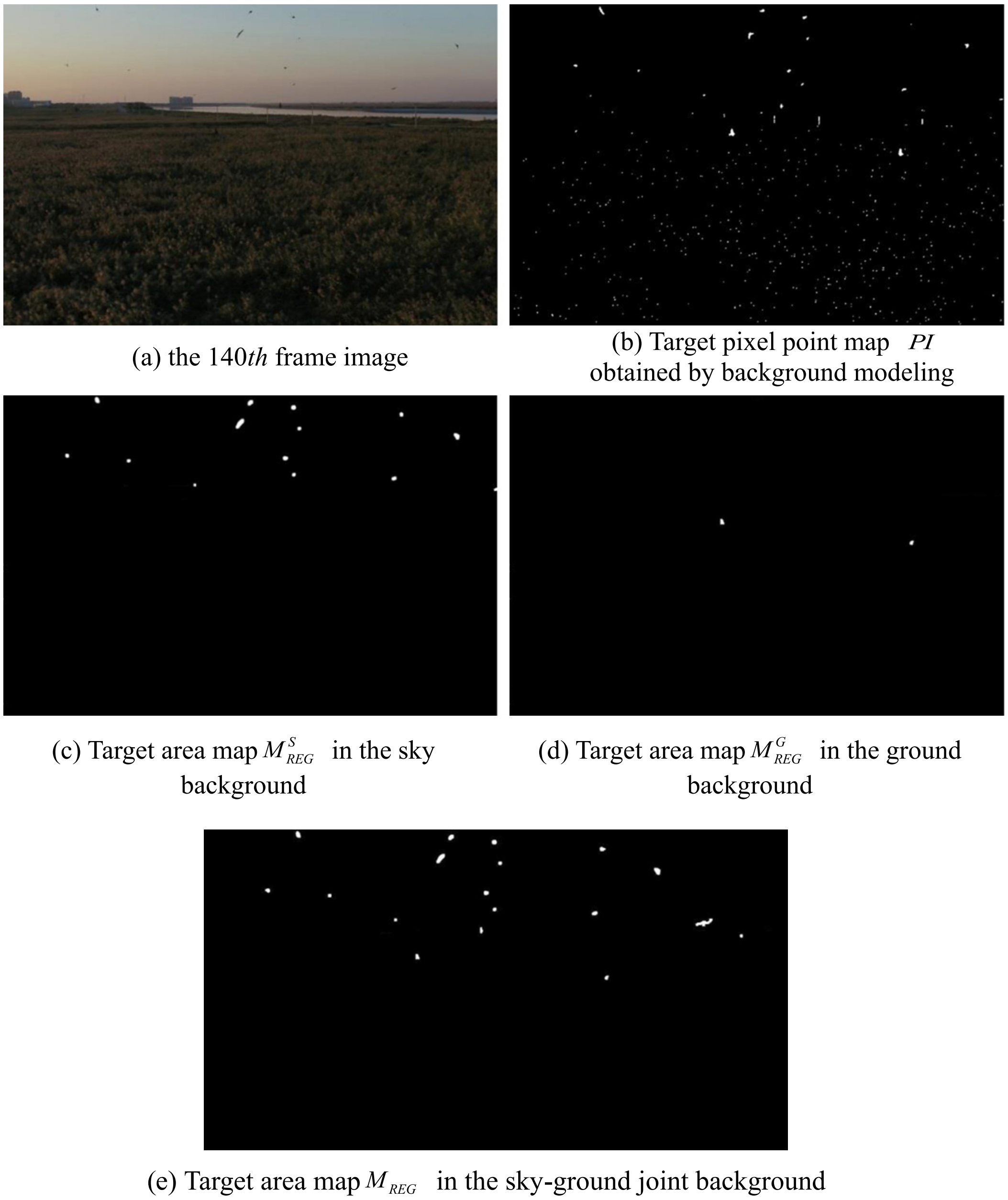


Figure 5. Target area map in the sky–ground joint background.

A. The target's area in the real target trajectory changes little, so the value of A is small, while the target's area in the false alarm target trajectory changes greatly, giving a large A value. Calculate the mean square error of the targets' moving pixels between adjacent frames in the target trajectories to obtain the target position feature V. The target's position in the real target trajectory changes little, so the value of V is small, while the target's position in the false alarm target trajectory changes greatly, giving a large V value. The mean square error of the angle between the target movement direction and the vertical direction in the target trajectory is calculated to obtain the target movement direction feature D. The target's movement direction is regular in the real target trajectory, so the value of D is small, while the target moving direction is aberrant in the false alarm target trajectory, giving a large D value.

Due to the movement of the target, the target trajectory features change greatly over time. Therefore, the target trajectories need to be segmented before processing. After segmentation, target trajectory features in every segment change little, which can improve the accuracy of false alarm target trajectories filtration. In this article, target trajectory features are extracted from each segment, which is set to be 10 consecutive frames. As shown in Eqs. (8)–(10), "*area*" represents the size of the target area, "*speed*" represents the moving pixels between adjacent frames, and "*direction*" represents the angle between the target movement direction and the vertical direction.

$$A = \mathrm{Var}(\mathrm{area}_t, \mathrm{area}_{t+1}, \ldots, \mathrm{area}_{t+9}) \tag{8}$$

$$V = \mathrm{Var}(\mathrm{speed}_t, \mathrm{speed}_{t+1}, \ldots, \mathrm{speed}_{t+9}). \tag{9}$$

$$D = \mathrm{Var}(\mathrm{direction}_t, \mathrm{direction}_{t+1}, \ldots, \mathrm{direction}_{t+9}). \tag{10}$$

Combining the target area feature A, target position feature V, and target movement direction feature D, the target trajectory feature vector $\mathrm{S} = [A, V, D]$ is obtained.

3.3.3 *Filtration of False Alarm Target Trajectories*

In the trajectory feature vector S, when A, V, and D are less than the corresponding threshold, the trajectory is determined to be a real target trajectory; otherwise, it is determined to be a false alarm target trajectory, as shown in Eq. (11).

$$\mathrm{flag} = \begin{cases} 1 & \text{if } A < m\_\mathrm{area},\ V < m\_\mathrm{speed},\ D < m\_\mathrm{direction} \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Here, *m_area*, *m_speed* and *m_direction* are the corresponding thresholds of A, V, and D, which are related to the target motion in the video. The mean square errors of A, V, and D values of multiple trajectories are normalized to get the minimum values of A, V and D. The corresponding thresholds of A, V, and D are determined by experiments to be double the minimum values. If *flag* is 1, the trajectory is determined to be a real target trajectory, otherwise, the trajectory is determined to be a false alarm target trajectory [21–23].

4. DETECTION OF FLYING TARGETS AGAINST FAST MOVING BACKGROUND

The flow chart of the detection algorithm for flying targets against fast moving background (DFT-FMB) is shown in Figure 6.

First, the target feature point optical flows are extracted from the sky–ground joint background. The saliency SR model [24] is used to extract the target pixel points, and target feature points are extracted from target pixel points. The Pyramidal implementation of the Lucas–Kanade feature tracker (PyrLK) algorithm [25] is used to calculate the target feature point optical flow.

Subsequently, the sky and the ground background are separated. The optical flow velocity of target feature point is directly calculated in the sky background, and that in the ground background is calculated after non-target feature point optical flows are filtered out. The optical flow velocities of target feature points are clustered based on the DBSCAN algorithm. So, the target feature points with similar optical flow velocity are clustered to detect the target.

Finally, the target detection results in the sky background and the ground background are combined to obtain the flying target detection results against the fast moving background.

4.1 *Extraction of Target Feature Point Optical Flow in the Sky–Ground Joint Background*

Once the photodetector platform finds the target, it will approach and fly close to the target. In doing so, the target can maneuver widely in the sky or above the ground and move at a high speed, making the photodetector platform, which is flying close by, to follow the target in vigorous motions in order to continuously detect the target. Under this condition, both the sky background and the ground background appear in the image and the background movement in the image is fast. Therefore, in order to process fast moving background and improve detection accuracy, the target saliency model algorithm and the pyramid optical flow algorithm are used to extract the feature point optical flows of the flying target in the sky–ground joint background.

4.1.1 *Extraction of Target Pixel Points by the SR Model*

In this article, the saliency SR model is used to establish the target saliency model of the image. Based on the target saliency model, whether the pixel points in the image are target pixel points or background pixel points is determined, in order to extract the target pixel points.

Firstly, the image $I(t)$ is transformed into the frequency domain by Fourier transform $F(\cdot)$. The amplitude spectrum curve $A(f)$ of the image is calculated by the modulus operation $Z(\cdot)$ in the frequency domain, the phase spectrum curve $P(f)$ of the image is calculated by the phase operation $S(\cdot)$, and the logarithmic spectrum curve $L(f)$ of the image is calculated according to the amplitude spectrum curve $A(f)$, as shown in Eq. (12), Eq. (13), and Eq. (14).

$$A(f) = Z(F[I(t)]) \tag{12}$$

$$P(f) = S(F[I(t)]) \tag{13}$$

$$L(f) = \log(A(f)). \tag{14}$$

Next, the logarithmic spectral curve $L(f)$ of the image is convolved using the mean operator $h_n(f)$ to obtain the background logarithmic spectral curve $h_n(f) * L(f)$, because the logarithmic spectral curve of the pure background area in the image is smooth. The target logarithmic curve $R(f)$ is obtained by subtracting $h_n(f) * L(f)$ from $L(f)$, because the singularity of $L(f)$is caused by the target. The calculation is shown in Eq. (15).

$$R(f) = L(f) - h_n(f) * L(f). \tag{15}$$

Subsequently, inverse Fourier transform $F^{-1}(\cdot)$ is used to inversely transform the target logarithmic spectral curve $R(f)$ in the frequency domain into the target saliency model $SR(t)$ in the spatial domain, as shown in Eq. (16).

$$SR(t) = g * F^{-1}[\exp(R(f) + P(f))]^2. \tag{16}$$

where g is the Gaussian filtering template.

Finally, it is determined whether the pixel points in the image are target pixel points or background pixel points according to the target saliency model $SR(t)$. If the value $SR_t(x, y)$ of a pixel point in the model is bigger than the

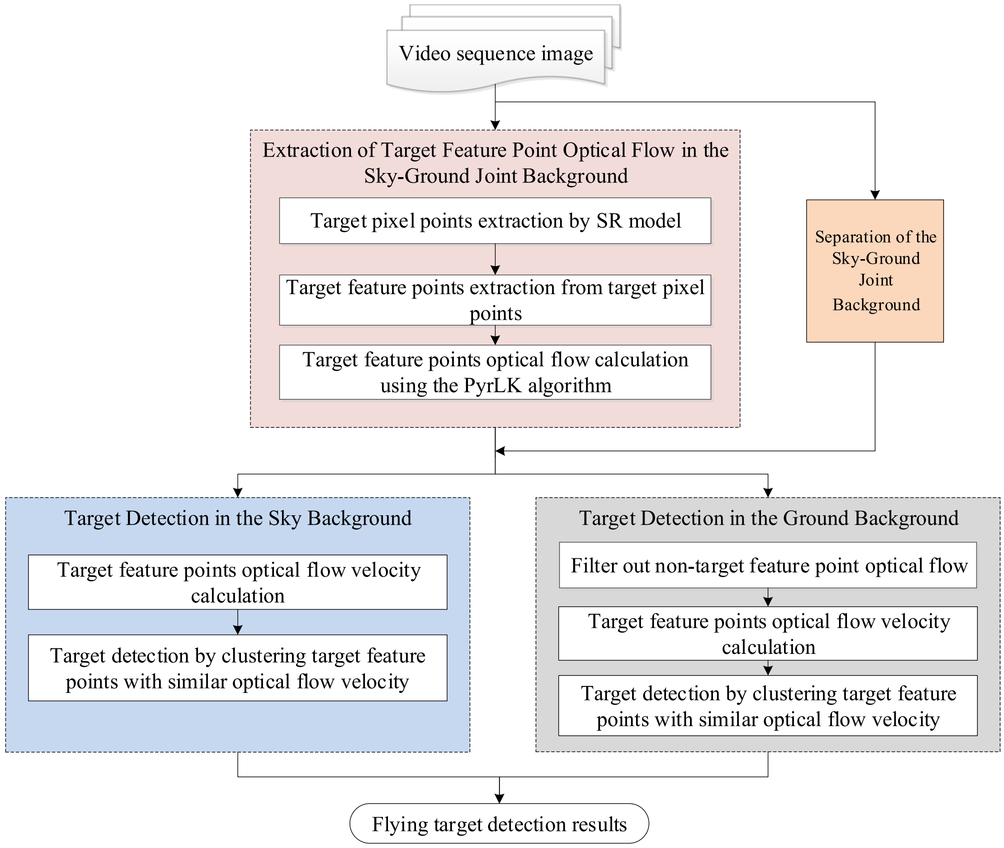


Figure 6. Flow chart of detection of flying targets against the fast moving background.

threshold γ, the pixel point $PS_t(x, y)$ at the corresponding position in the image is a target pixel point, otherwise it is a background pixel point. Thereby the target pixel point map $PS(t)$ is obtained, as shown in Eq. (17).

$$PS_t(x, y) = \begin{cases} 1 & \text{if } SR_t(x, y) > \gamma \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

where γ is the adaptive threshold. $\gamma = E(SR(t))$, where $E(SR(t))$ is the average value of all pixel points in the target saliency model.

4.1.2 *Extraction of Feature Point Optical Flow in Target Pixel Points*

The target pixel points in the image $I(t)$ are obtained according to the target pixel point map $PS(t)$. The target feature points are extracted from the target pixel points, and the target feature point optical flows are calculated based on the PyrLK optical flow method. For two adjacent frames of video images $I(t)$ and $I(t+1)$, the steps for calculating the target feature point optical flows are as follows:

(1) In the image $I(t)$, the ORB feature points are extracted from the target pixel points to obtain the target feature point set SoP_t.
(2) In image $I(t+1)$, the PyrLK optical flow method is used to calculate the matching optical flow tracking point of every target feature point in image $I(t)$, then obtain its position $(x'_{i,t}, y'_{i,t})$ after motion compensation based on the global background motion parameter. Thereby the target feature point optical flow set $SoF_t = \{v_{i,t}, \boldsymbol{v}_{i,t} = (x'_{i,t} - x_{i,t}, y'_{i,t} - y_{i,t}), i = 1, \ldots, N\}$ in image $I(t)$ is obtained, where $(x_{i,t}, y_{i,t})$ is the target feature point and N is the number of target feature points.

4.2 *Target Detection against the Sky Background and the Ground Background*

When the target engages in large-range maneuvering, the background in the video image moves rapidly. As a result, there are non-target feature point optical flows in the target feature point optical flow set SoF_t. Because the features of the sky background and the ground background are distinctive, the distinguishability between the target feature point optical flows and the two types of background is different. Therefore, for the target feature point optical flow set SoF_t, the optical flow velocity of target feature points in sky background and the ground background is calculated separately, and then the target feature points are clustered according to their optical flow velocity to detect the target. This method improves the target detection accuracy.

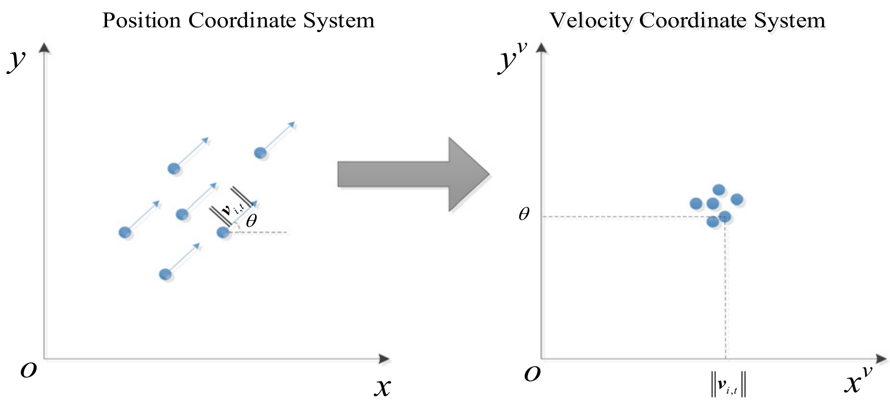


Figure 7. Position coordinate system of feature point and optical flow velocity coordinate system of feature point.

4.2.1 *Calculation of the Target Feature Point Optical Flow Velocity*

In the sky background, the optical flow velocity of a target feature point is directly calculated as $\boldsymbol{v}_{i,t}^{S} = (x'_{i,t}, y'_{i,t}) - (x_{i,t}, y_{i,t})$, where $(x'_{i,t}, y'_{i,t})$ is the position of the matching optical flow tracking point in image $I(t+1)$ after motion compensation, which corresponds to the target feature point $(x_{i,t}, y_{i,t})$ in image $I(t)$.$(x'_{i,t}, y'_{i,t})$ is calculated based on the global background motion parameter H_t^{t+1}, as shown in Eq. (18).

$$\begin{pmatrix} x'_{i,t} \\ y'_{i,t} \end{pmatrix} = H_t^{t+1} \begin{pmatrix} x_{i,t} \\ y_{i,t} \end{pmatrix}. \tag{18}$$

In the ground background, there are non-target feature point optical flows in the target feature point optical flow set due to the complexity of it. According to the difference between the target feature point optical flows and the background motion optical flows, the non-target feature point optical flows in ground background are filtered out. The optical flow velocity of a target feature point is calculated as $\boldsymbol{v}_{i,t}^{G} = (x'_{i,t}, y'_{i,t}) - (x_{i,t}, y_{i,t})$. If the optical flow velocity of the target feature point satisfies Eq. (19), which means that it shows a big difference from the background motion, then it is retained. Otherwise, it is determined to be non-target feature point optical flow and is filtered out.

$$\left\| \boldsymbol{v}_{i,t}^{G} \right\| > v_{\min}, \tag{19}$$

where $v_{\min}$ is the threshold value of the target feature point optical flow velocity, the value of which depends on the speed of the background and the target, and is determined to be 1 by experiments.

4.2.2 *Target detection by Clustering Target Feature Points with Similar Optical Flow Velocity*

According to the calculated target feature point optical flow velocity $\boldsymbol{v}_{i,t}$ ($\boldsymbol{v}_{i,t}^{S}$ and $\boldsymbol{v}_{i,t}^{G}$), the target feature points with similar optical flow velocity are clustered based on the DBSCAN algorithm to detect the target.

The traditional DBSCAN algorithm clusters feature points in the image that are close to each other to detect the target. Because some feature points of the same target are relatively close to each other while some are far away, the target detection accuracy is not high if the feature points are clustered according to their relative position distance. In contrast, the speed of a specific target is relatively consistent, thus the optical flow speed of the feature points belonging to the same target is also consistent. Therefore, target feature points with similar optical flow velocity are clustered into one target, in order to improve the target detection accuracy. The traditional feature point position coordinate system xoy is improved into the feature point optical flow velocity coordinate system $x^v oy^v$ in this article, on which the feature points belonging to the same target are close to each other, as shown in Figure 7.

The abscissa of the feature point optical flow velocity coordinate system $x^v oy^v$ is the magnitude $\left\| \boldsymbol{v}_{i,t} \right\|$ of the feature point optical flow velocity $\boldsymbol{v}_{i,t}$, and the ordinate is the angle θ of $\boldsymbol{v}_{i,t}$, as shown in Eq. (20) and Eq. (21).

$$x^v : \left\| \boldsymbol{v}_{i,t} \right\| = \sqrt{(\boldsymbol{v}_{i,t}^{x})^2 + (\boldsymbol{v}_{i,t}^{y})^2} \tag{20}$$

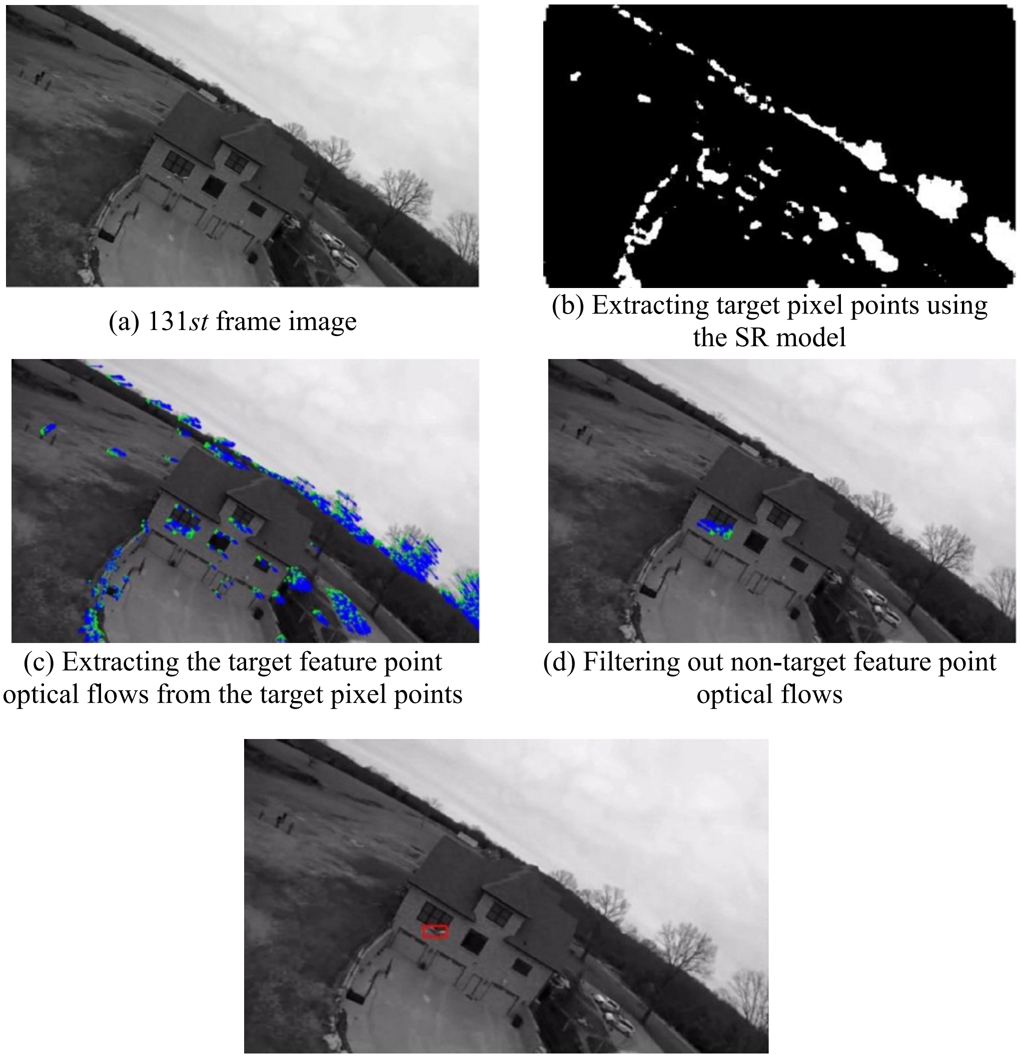
$$y^v : \theta = \arctan \frac{\boldsymbol{v}_{i,t}^{y}}{\boldsymbol{v}_{i,t}^{x}}, \tag{21}$$

where $\boldsymbol{v}_{i,t}^{y}$ represents the component of the optical flow velocity of the target feature point in the vertical direction, and $\boldsymbol{v}_{i,t}^{x}$ represents the component of the optical flow velocity of the target feature point in the horizontal direction.

In the feature point optical flow velocity coordinate system, according to the amplitude and the angle of the target feature point optical flow velocity, the target feature points with similar optical flow velocity are clustered into one target based on the DBSCAN algorithm to obtain the target detection result, as shown in Figure 8.

5. EXPERIMENTAL RESULTS AND ANALYSIS

In order to verify the performance of the proposed algorithm for detecting air-to-air flying targets in the sky–ground joint background, the detection algorithm for small flying targets against slow moving background (DSFT-SMB) and detection



(a) 131*st* frame image

(b) Extracting target pixel points using the SR model

(c) Extracting the target feature point optical flows from the target pixel points

(d) Filtering out non-target feature point optical flows

(e) Target detection by clustering target feature points with similar optical flow velocity

Figure 8. Detection of flying targets against fast moving background.

algorithm for flying targets against fast moving background (DFT-FMB) proposed in this article are compared with three other superior algorithms for multiple moving target detection, which are the detection algorithm based on dual-mode Gaussian background modeling (DGM), the detection algorithm based on clustering algorithm density based spatial clustering of applications with noise(DBSCAN), and the detection algorithm based on the clustering algorithm evolutionary local mean (ELM).

All experimental results were obtained with the same data and initialization conditions. The experiment environment is: VS2010, Matlab2016. The experiment platform is: 3.60 Ghz-Intel i7 processor, 64-bit win7 system, and 8 GB memory.

5.1 *Evaluation Indexes*

The recall rate (R), precision rate (P), and F-measure (F) are used to quantitatively evaluate the performances of multiple target detection algorithms. R represents the proportion of targets that are correctly detected out of all the targets. P represents the proportion of correctly detected targets out of all the detection results. F is the harmonic weighted average of the two, which is calculated by Eq. (22), Eq. (23), and Eq. (24).

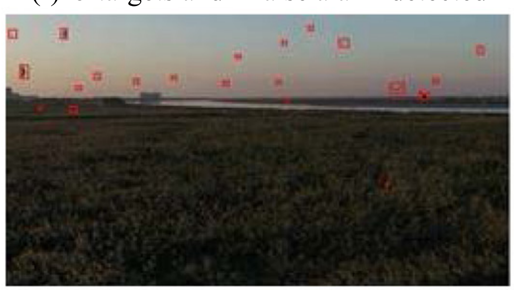
$$R = TP/(TP + FN), \tag{22}$$

$$P = TP/(TP + FP), \tag{23}$$

$$F = \frac{(1 + \beta^2) \times P \times R}{\beta^2 \times P + R}. \tag{24}$$

where TP represents the number of targets that are correctly detected as targets, FN represents the number of targets that are incorrectly detected as background, and FP represents the number of backgrounds that are incorrectly detected as targets. β is the weight which represents the relative significance of the recall and precision rates. $\beta = 1$ means that the recall and precision rates are equally significant. $\beta > 1$ means that the recall rate is more significant. $\beta < 1$ means that the precision rate is more significant. In this article, β is set to 1 to take both the recall rate and the precision rate into consideration.

Table I. Experimental results of multiple small flying target detection in the direct sunlight environment.

Algorithm	150^{th} frame (21 flying targets, 1 UAV, 20 flying birds)	170^{th} frame (21 flying targets, 1 UAV, 20 flying birds)
DGM		
	(a)17 targets and 9 false alarms detected	(b)18 targets and 4 false alarms detected
DBSCAN		
	(c)5 targets and 0 false alarm detected	(d)12 targets and 0 false alarm detected
ELM		
	(e)6 targets and 0 false alarm detected	(f)15 targets and 1 false alarm detected
DSFT-SMB		
	(g)21 targets and 0 false alarm detected	(h)21 targets and 1 false alarm detected

5.2 *Experimental Data*

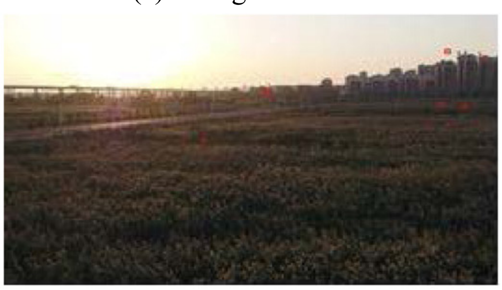
The experimental videos in this article are divided into two categories: small flying targets against the slow moving background and flying targets against the fast moving background.

5.2.1 *Experimental Videos for Small Flying Target Detection Against Slow Motion Background*

The experimental videos for small flying target detection against slow moving background need to meet certain conditions, such as containing slow moving background, the sky–ground joint background, small flying targets, etc. After thorough investigation, no experimental video meeting all the conditions was found in the open data set. Therefore, the two experimental videos are got through using method of field shooting by our team.

The first video contains 4400 frames with an image size of 1920×1056 pixels. The background of the image is comprised of half sky and half grassland separated by a river. There are a large number of targets in the video,

Table II. Experimental results of multiple small flying target detection in the backlighting environment.

Algorithm	56^{th} frame (4 flying targets, 1 UAV, 3 flying birds)	180^{th} frame (4 flying targets, 1 UAV, 3 flying birds)
DGM	 (a)3 targets and 5 false alarms detected	 (b)1 target and 3 false alarms detected
DBSCAN	 (c)no target detected	 (d)no target detected
ELM	 (e)no target detected	 (f)no target detected
DSFT-SMB	 (g)4 targets and 2 false alarms detected	 (h)4 targets and 0 false alarm detected

including a small UAV and multiple flying birds. The UAV flies from the near to the far and is captured in the video in the ground background first before moving into the sky background. Videos were shot in the direct sunlight environment. The movement of the shooting platform caused the background to move slowly during shooting. From the whole video, 200 consecutive images were selected to obtain the experimental video VD-1.

The second video contains 2200 frames with the image size of 1920×1064 pixels. The background of the image is comprised of half sky and half grassland separated by a building. There are some targets in the video, including a small UAV and some flying birds. The UAV flies from the far to the near and is captured in the video in the sky background first before moving into the ground background. Videos were shot in the backlighting environment. The movement of the shooting platform caused the background to move slowly during shooting. From the whole video, 200 consecutive images were selected to obtain the experimental video VD-2.

Table III. Performance comparison of detection algorithms on the small flying target detection against the slow moving background.

Experiment video	Algorithm	TP	FP	FN	R(%)	P(%)	F(%)
	DGM	2430	1966	858	73.9	55.3	63.3
	DBSCAN	1196	**63**	2089	36.4	95.0	52.6
VD-1	ELM	1751	122	1534	53.3	93.5	67.9
	DSFT-SMB	**3203**	122	**85**	**97.4**	**96.3**	**96.8**
	DGM	238	834	670	26.2	22.3	24.1
	DBSCAN	0	—	—	0	0	0
VD-2	ELM	0	—	—	0	0	0
	DSFT-SMB	**858**	**161**	**50**	**94.5**	**84.2**	**89.1**

5.2.2 *Experimental Videos for Flying Target Detection against Fast Motion Background*

The experimental videos for flying target detection against fast moving background need to meet certain conditions, such as containing fast moving background, the sky–ground joint background, flying targets, etc. After thorough investigation, some videos in the planes' data set taken by Artem Rozantsev's computer vision laboratory can meet these conditions. Therefore, Video_13 and Video_16 were selected from the planes' data set as the experimental videos.

Video_13 contains 375 frames with the image size of 720 × 540 pixels. The target flies from the near to the far and is captured in the video in the ground background first before moving into the sky background. 280 consecutive images meeting the conditions were selected and processed to obtain the first experimental video video-1, which has the image size of 720 × 430 pixels.

Video_16 contains 333 frames with the image size of 720 × 540 pixels. The target also flies from the near to the far and is captured in the video in the ground background first before moving into the sky background. 260 consecutive images meeting the conditions were selected and processed to obtain the second experimental video video-2, which has the image size of 720 × 430 pixels.

Since the part of photodetector platform is captured in the original video images, it is detected as a target. In order to avoid this false detection, the part of the images including the photodetector platform is removed. Hence after processing, the image size of experimental videos is changed from the original 720 × 540 pixels to 720 × 430 pixels, as shown in Figure 9.

5.3 *Detection Experiment of Small Flying Targets against Slow Moving Background*

(1) The performance of the proposed algorithm DSFT-SMB for detecting multiple small flying targets in the direct sunlight environment and the sky–grassland background is verified on VD-1. Comparative experimental results of the proposed algorithm and three other superior target detection algorithms are shown in Table I.

Table I shows that the proposed algorithm DSFT-SMB detects the most targets in this case. Although there are more false alarms reported by the proposed algorithm than by the algorithm DBSCAN, the number of true targets detected by the proposed algorithm greatly exceeds that by the algorithm DBSCAN. Therefore, the performance of the proposed algorithm DSFT-SMB is better than that of the three other algorithms.

(2) The performance of the proposed algorithm DSFT-SMB for detecting multiple small flying targets in the backlighting environment and the sky–grassland background is verified on VD-2. Comparative experimental results of the proposed algorithm and three other superior target detection algorithms are shown in Table II.

Table II shows that the algorithm DSFT-SMB detects the most targets and the least false alarms in this case. Therefore, the performance of the proposed algorithm DSFT-FMB is better than that of the three other algorithms.

(3) The performance of the proposed detection algorithm DSFT-SMB for small flying targets against the slow moving background is compared with three other superior target detection algorithms, and their performance is evaluated by *TP*, *FP*, *R*, *P*, and *F* indicators. Experimental results are shown in Table III.

Table III shows that the proposed algorithm DSFT-SMB has the highest recall ratio *R*, precision ratio *P*, and *F* measures, indicating that the performance of DSFT-SMB algorithm is better than that of the three other comparison algorithms. Although in the experimental video VD-1, the DBSCAN algorithm has the least *FP* and the ELM algorithm has the same *FP* as the proposed algorithm, the recall rates of the DBSCAN algorithm and the ELM algorithm are 36.4% and 53.3%, respectively, significantly lower than that of the proposed algorithm, which is 97.4%. In the experimental video VD-2, the DBSCAN algorithm and the ELM algorithm are unable to detect any target so *TP* is 0, and recall ratio *R*, precision ratio *P*, and *F* measure are all 0, rendering the FP and the FN meaningless. Together, these experimental results show that DSFT-SMB performs the best for small flying target detection against the slow moving background.

5.4 *Detection Experiment of Flying Targets against Fast Moving Background*

(1) The performance of the proposed algorithm DFT-FMB for detecting flying targets in fast rotating background is verified on video-1. Comparative experimental results of the proposed algorithm and three other superior target detection algorithms are shown in Table IV.

Table IV shows that the proposed algorithm DFT-FMB has the highest *TP* and the lowest *FP*, which demonstrates that the proposed algorithm correctly detects the most number of true targets and the least number of false alarms. Therefore, the performance of the proposed algorithm DFT-FMB is better than that of three other algorithms.

(2) The performance of the proposed algorithm DFT-FMB for detecting flying targets in fast moving background is verified on video-2. Comparative experimental results of the

Table IV. Experimental results of flying target detection in fast rotating background.

Algorithm	74th frame (1 flying target)	Detected target area	264th frame (1 flying target)	Detected target area	Full video
DGM	 (a)	 (b) 1 target and 3 false alarms detected	 (c)	 (d) no target detected	*TP*=165 *FP*=255
DBSCAN	 (e)	 (f) no target detected	 (g)	 (h) no target detected	*TP*=215 *FP*=329
ELM	 (i)	 (j) 1 target and 1 false alarm detected	 (k)	 (l) no target detected	*TP*=210 *FP*=245
DFT-FMB	 (m)	 (n) 1 target and 0 false alarm detected	 (o)	 (p) 1 target and 0 false alarm detected	*TP*=221 *FP*=96

proposed algorithm and three other superior target detection algorithms are shown in Table V.

Table V shows that the proposed algorithm DFT-FMB has the highest *TP* and the lowest *FP*, which demonstrates that the proposed algorithm correctly detects the most number of true targets and the least number of false alarms. Therefore, the performance of the proposed algorithm DFT-FMB is better than that of three other algorithms.

(3) The performance of the proposed detection algorithm DFT-FMB for flying targets against the fast moving background is compared with three other superior target detection algorithms, and their performance is evaluated by *TP*, *FP*, *R*, *P*, and *F* indicators. Experimental results are shown in Table VI.

Table VI shows that the proposed algorithm DFT-FMB has the highest *TP*, *R*, *P*, *F*, and the lowest *FP*, *FN*, indicating that the performance of the DFT-FMB algorithm is better than that of the three other comparison algorithms. In the experimental video-1 and video-2, all the three other algorithms failed to cope with the changes in the sky–ground joint background caused by rapid background movements, resulting in high false alarm rates and low

(a) Original video (b) Experimental video

Figure 9. Video image processing.

Table V. Experimental results of flying target detection in fast moving background.

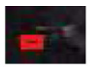
Algorithm	36th frame (1 flying target)	Detected target area	136th frame (1 flying target)	Detected target area	Full video
DGM	 (a)	 (b) 1 target and 3 false alarms detected	 (c)	 (d) 1 target and 1 false alarm detected	TP=182 FP=790
DBSCAN	 (e)	 (f) no target detected	 (g)	 (h) 1 target and 0 false alarm detected	TP=201 FP=479
ELM	 (i)	 (j) no target detected	 (k)	 (l) no target detected	TP=150 FP=389
DFT-FMB	 (m)	 (n) 1 target and 0 false alarm detected	 (o)	 (p) 1 target and 0 false alarms detected	TP=218 FP=172

Table VI. Performance comparison of detection algorithms on flying target detection against the fast moving background.

Experiment video	Algorithm	TP	FP	FN	R(%)	P(%)	F(%)
video-1	DGM	165	255	85	66.0	39.3	49.3
	DBSCAN	215	329	54	79.9	39.5	52.9
	ELM	210	245	40	84.0	46.2	59.6
	DFT-FMB	**221**	**96**	**29**	**88.4**	**69.7**	**77.9**
video-2	DGM	182	790	98	65.0	18.7	29.0
	DBSCAN	201	479	79	71.8	29.6	41.9
	ELM	150	389	130	53.6	27.8	36.6
	DFT-FMB	**218**	**172**	**62**	**77.9**	**55.9**	**65.1**

precision ratio, none of which exceed 50%. The algorithm DFT-FMB proposed in this article calculates the optical flow velocity of target feature points in the sky background and the ground background separately, and clusters the target feature points with similar optical flow velocity into one target, which improves the target detection accuracy. Experimental results show that the DFT-FMB algorithm shows the best performance for flying targets detection against the fast moving background.

6. CONCLUSION

To tackle the technical challenge that the sky–ground joint background contains both the sky and the ground backgrounds, this article proposes a separation algorithm for the sky–ground joint background. Based on the separation algorithm, a detection algorithm for air-to-air flying targets against the sky–ground joint background is further proposed. The proposed algorithm is divided into the detection algorithm for small flying targets against the slow moving background and the detection algorithm for flying targets against the fast moving background, which respectively solves the difficult problems of detecting small flying targets against the slow moving background and flying targets against the fast moving background. Experimental results show that the proposed algorithm yield high recall rate, precision rate, and F-measure in the detection of air-to-air flying targets against the sky–ground joint background. Future research will take other background types into consideration such as the sea and winding coastlines and focus on multiple moving target detection against the air–sea background. The current research results will be improved to develop an excellent multiple moving target detection algorithm against the air–sea background.

ACKNOWLEDGMENT

This work was supported by the Fundamental Research Funds for the Central Universities (NJ2020021 and NT2020022), the National Natural Science Foundation of China (61471194 and 61705104),and the Natural Science Foundation of Jiangsu Province (BK20170804).

REFERENCES

[1] Z. Li, L. Wang, and J. Cui, "The design of the detecting and tracking system for flying weak-target under complex environment," *Proc. Sixth National Conf. on Information Acquisition and Processing (2)* China (Journal of Instrumentation, Chinese Institute of Instrumentation), 2008, pp. 644–648.

[2] M. Siam, R. Elsayed, and M. Elhelw, "On-board multiple target detection and tracking on camera-equipped aerial vehicles," *2012 IEEE Int'l. Conf. on Robotics and Biomimetics (ROBIO)*, (IEEE, Piscataway, NJ, 2012), pp. 2399–2405.

[3] L. Kovács, A. Kovács, Á. Utasi, and T. Szirányi, "Flying target detection and recognition by feature fusion," Opt. Engng. **51**, 117002 (2012).

[4] K. M. Yi, K. Yun, S. W. Kim, H. J. Chang, H. Jeong, and J. Y. Choi, "Detection of moving objects with non-stationary cameras in 5.8 ms: bringing motion detection to your mobile device," *2013 IEEE Conf. on Computer Vision and Pattern Recognition Workshops* (IEEE, Piscataway, NJ, 2013), pp. 27–34.

[5] P. Sadeghi-Tehran, C. Clarke, and P. Angelov, "A real-time approach for autonomous detection and tracking of moving objects from UAV," *2014 IEEE Symposium on Evolving and Autonomous Learning Systems (EALS)* (IEEE, Piscataway, NJ, 2014), pp. 43–49.

[6] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," *2011 Int'l. Conf. on Computer Vision* (IEEE, Piscataway, NJ, 2011), pp. 2548–2555.

[7] Q. Xu and X. Shi, "A simplified bird skeleton based flying bird detection," *Proc. 11th World Congress on Intelligent Control and Automation* (IEEE, Piscataway, NJ, 2014), pp. 1075–1078.

[8] A Rozantsev, V Lepetit, and P. Fua, "Flying objects detection from a single moving camera," *2015 IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2015), pp. 4128–4136.

[9] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting flying objects using a single moving camera," IEEE Trans. on Pattern Anal. Machine Intelligence **39**, 879–892 (2016).

[10] P. Wang, "Study on air target detection based on videos," *University of Electronic Science and Technology of China* (2019), pp. 1–67.

[11] F. Poiesi and A. Cavallaro, "Detection of fast incoming objects with a moving camera," *British Machine Vision Conference (BMVC)* (York, UK, 2016), pp. 146.1–146.11.

[12] J. Lou, W. Zhu, H. Wang, and M. Ren, "Small target detection combining regional stability and saliency in a color image," Multimedia Tools Appl. **76**, 14781–14798 (2017).

[13] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer, "Deep cross-domain flying object classification for robust UAV detection," *2017 14th IEEE Int'l. Conf. on Advanced Video and Signal Based Surveillance (AVSS)* (IEEE, Piscataway, NJ, 2017), pp. 1–6.

[14] R. Yoshihashi, T. T. Trinh, R. Kawakami, S. You, M. lida, and T. Naemura, "Differentiating objects by motion: joint detection and tracking of small flying objects," Computer Vision and Pattern Recognition. 2017, pp. 1–10.

[15] X. Shi, Z. Chen, H. Wang, D. Yeung, W. Wong, and W. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *29th Annual Conf. on Neural Information Processing Systems (NIPS)* (ACM, New York, NY, 2015), Advances in neural information processing systems. 2015(28): pp. 802–810.

[16] Z. Zhang, Y. Cao, M. Ding, L. Zhuang, and Z. Wang, "Spatial and temporal context information fusion based flying objects detection for autonomous sense and avoid," *2018 Int'l. Conf. on Unmanned Aircraft Systems (ICUAS)* (IEEE, Piscataway, NJ, 2018), pp. 569–578.

[17] Y. K. Chen and Y. H. Chen, "Flying bird detection under shaking background," *2018 IEEE Int'l. Conf. on Information and Automation (ICIA)* (IEEE, Piscataway, NJ, 2018), pp. 439–443.

[18] J. Janousek, J. Novotny, P. Marcon, A. Siruckova, and R. Kadlec, "Algorithms for flying object detection," *2018 Progress in Electromagnetics Research Symposium (PIERS-Toyama)* (IEEE, Piscataway, NJ, 2018), pp. 782–786.

[19] J. Canny, "A computational approach to edge detection," IEEE Trans. on Pattern Anal. Mach. Intelligence **1986**, 679–698.

[20] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," *Int'l. Joint Conf. on Artificial Intelligence* (Morgan Kaufmann Publishers Inc, USA, 1981), pp. 674–679.
[21] X. Liu and X. Zhang, "NOMA-based resource allocation for cluster-based cognitive industrial internet of things," IEEE Trans. on Industrial Informatics **16**, 5379–5388 (2020).
[22] X. Liu, X. Zhai, W. Lu, and C. Wu, "QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA," IEEE Trans. on Industrial Informatics **17**, 2052–2061 (2021).
[23] X Liu and X. Zhang, "Rate and energy efficiency improvements for 5G-based iot with simultaneous transfer," IEEE Internet of Things J. **6**, 5971–5980 (2019).
[24] X. Hou and L. Zhang, "Saliency detection: A spectral residual approach," *2007 IEEE Conf. on Computer Vision and Pattern Recognition* (IEEE, Piscataway, NJ, 2007), pp. 1–8.
[25] B. Jean-Yves, "Pyramidal implementation of the Lucas Kanade feature tracker description of the algorithm," *Intel Corporation, OpenCV Documents* (1999), Vol. 22, pp. 363–381.