

Article

Moving Object Detection in Traffic Surveillance Video: New MOD-AT Method Based on Adaptive Threshold

Xiaoyue Luo ^{1,2}, Yanhui Wang ^{1,2,*}, Benhe Cai ^{1,2} and Zhanxing Li ^{1,2}

¹ College of Resources Environment and Tourism, Capital Normal University, Beijing 100048, China; xiaoyueluo@cnu.edu.cn (X.L.); benhecai@cnu.edu.cn (B.C.); 2200902158@cnu.edu.cn (Z.L.)

² Key Laboratory of 3-Dimensional Information Acquisition and Application, Ministry of Education, Capital Normal University, Beijing 100048, China

* Correspondence: yanhuiwang@cnu.edu.cn

Abstract: Previous research on moving object detection in traffic surveillance video has mostly adopted a single threshold to eliminate the noise caused by external environmental interference, resulting in low accuracy and low efficiency of moving object detection. Therefore, we propose a moving object detection method that considers the difference of image spatial threshold, i.e., a moving object detection method using adaptive threshold (MOD-AT for short). In particular, based on the homograph method, we first establish the mapping relationship between the geometric-imaging characteristics of moving objects in the image space and the minimum circumscribed rectangle (BLOB) of moving objects in the geographic space to calculate the projected size of moving objects in the image space, by which we can set an adaptive threshold for each moving object to precisely remove the noise interference during moving object detection. Further, we propose a moving object detection algorithm called GMM_BLOB (GMM denotes Gaussian mixture model) to achieve high-precision detection and noise removal of moving objects. The case-study results show the following: (1) Compared with the existing object detection algorithm, the median error (MD) of the MOD-AT algorithm is reduced by 1.2–11.05%, and the mean error (MN) is reduced by 1.5–15.5%, indicating that the accuracy of the MOD-AT algorithm is higher in single-frame detection; (2) in terms of overall accuracy, the performance and time efficiency of the MOD-AT algorithm is improved by 7.9–24.3%, reflecting the higher efficiency of the MOD-AT algorithm; (3) the average accuracy (MP) of the MOD-AT algorithm is improved by 17.13–44.4%, the average recall (MR) by 7.98–24.38%, and the average F1-score (MF) by 10.13–33.97%; in general, the MOD-AT algorithm is more accurate, efficient, and robust.

Keywords: video GIS; moving object detection; threshold differentiation; MOD-AT; BLOB; mapping relationship; adaptive threshold



Citation: Luo, X.; Wang, Y.; Cai, B.; Li, Z. Moving Object Detection in Traffic Surveillance Video: New MOD-AT Method Based on Adaptive Threshold. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 742. <https://doi.org/10.3390/ijgi10110742>

Academic Editor: Wolfgang Kainz

Received: 24 August 2021

Accepted: 23 October 2021

Published: 1 November 2021

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



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the rapid growth of information technology, surveillance cameras have been widely used because of their advantages of real-time performance, low cost, and efficiency. They have become an indispensable technical means of urban management in terms of safety. Massive and real-time video data contain abundant spatiotemporal information and provide essential support for real-time supervision, case investigation, and natural resource monitoring [1,2]. However, video data have the disadvantages of low value, redundancy, and noise. Although significant progress has been made in low-level video understanding, high-precision and high-level video understanding technology is still in the research stage [3]. Moving objects, such as vehicles and pedestrians in traffic videos, have high application requirements for traffic supervision departments. The moving object is the key to video information mining. Setting reasonable thresholds to detect a high-precision moving object from surveillance video is one of the current hot issues in video geographic-information-system (GIS) research [4].

At present, advances have been made in moving object detection, and the application of specific scenarios is close to the practical level in terms of accuracy and efficiency [5]. However, due to the influence of camera-imaging characteristics, the imaging size of the moving object is different, and moving object detection results are easily disturbed by the external environment. As a result, the accuracy of moving object detection is easily affected by noise. Moreover, the previous denoising methods have low efficiency and accuracy in detecting moving objects. The reason for this is that the threshold range impacts the accuracy, efficiency, and noise filtering of moving object detection [6,7]. The traditional method uses the unified threshold to remove noise in the image space; when the threshold value is set too high, an object far away from the camera position will be filtered, and when the threshold value is set too low, noise near the camera position will be retained, so it is challenging to set an appropriate threshold [8]. Moreover, for the same moving object, its size is constant at different locations in geographic space. However, due to the influence of the imaging characteristics of the camera, the imaging size of the same object in the image space is inconsistent, so it is essential to set different threshold intervals at different positions. Therefore, it is necessary to set an adaptive threshold according to the imaging characteristics of the camera in the image space to achieve high-precision detection of moving objects. To summarize, we propose an adaptive threshold-calculation method from the perspective of geographic space, taking into account the projection size, imaging characteristics, and semantic information of the object in geographic space.

The rest of this paper is organized as follows: A literature review of moving object detection and threshold-setting algorithms is provided in Section 2. The general idea of a moving object detection method considering the difference of image spatial threshold is given in Section 3, and an adaptive threshold calculation method and moving object detection algorithm are derived in detail. In Section 3, the adaptive threshold of surveillance video is calculated, and the accuracy and efficiency of moving object detection is verified by single-frame accuracy and overall accuracy. Section 5 summarizes the main conclusions and discusses planned future work.

2. Related Work

To achieve more efficient and high-precision detection of video moving objects, it is necessary to set a reasonable threshold to process the candidate objects in the video imaging space to effectively eliminate the external-environment interference [9–11]. Depending on the location, height, and posture of the surveillance camera, the thresholds of filtering noise (such as leaf shaking, light effects, etc.) are different. The accuracy and efficiency of moving object detection will also vary [12,13]. At present, the moving object detection problem can be divided into the following three categories from the perspective of threshold setting [14,15].

1. Moving object detection methods based on traditional single threshold

These methods mainly include the frame difference [16–18], optical flow [19,20], and background difference methods [21–23], among others. For example, Zuo et al. [24] improved the accuracy of moving object detection based on the background frame difference method. Luo et al. [25] combined the background difference and frame difference methods to detect moving objects and remove external-environment interference. Akhter et al. [26] realized contour detection and feature-point extraction of moving objects through the optical flow method. Li et al. [27] used the background difference algorithm to obtain the foreground and background images, and then extracted the moving object in the surveillance video. The above algorithm uses a unified threshold value in the image space to filter the interference of the external environment, resulting in an unreasonable threshold-value setting, thereby affecting the accuracy of moving object detection.

2. Moving object detection methods based on pixels or regions

Compared with the region-based moving object detection method, the pixel-based moving object detection method is fast and straightforward, which is suitable for the

rapid monitoring of video objects. Its typical methods include the vibe algorithm [28–30], non-parametric model [31], and Gaussian mixture model (GMM) [32], among others. For example, Liu et al. [33] proposed the three-frame difference algorithm of the adaptive GMM to suppress the external environment's interference effectively. Zuo et al. [34] improved the accuracy of the moving object detection algorithm based on the improved GMM (IGMM). The aforementioned algorithm only sets a fixed threshold range according to the geometric-imaging characteristics of moving objects in image space, and the suitability of the detection threshold for moving objects is not considered. According to the perspective-imaging characteristics of the camera, when the object is close to the camera, the area of the object in the image is larger; otherwise, when the object is far away, the area of the object in the image is smaller. In fact, the actual size of the object will not change in the geographical space [35].

3. Moving object detection method based on the segmented threshold

This method attempts to calculate thresholds in different regions of the image space. For example, Chan et al. [36] proposed linear segmentation of video frames to obtain weight maps at different locations and calculated the threshold range based on the position of the object in the video frame. Chang et al. [37] proposed a method of spatial-imaging area normalization. The area weight of the object in the image space is obtained to filter part of the moving object detection interference. Lin et al. [38] established a single mapping relationship between video image space and geographic space and realized interference filtering from the external environment based on the non-linear-perspective correction model algorithm (NPCM). However, the above algorithm only considers the linear or non-linear characteristics of the object on the image and does not consider the projection distortion of the object size caused by the camera-imaging mechanism. It also ignores the difference in imaging the geometric characteristics of moving objects in different positions of the video frame. The threshold setting is not very specific, affecting the accuracy of moving object detection.

In this paper, we propose a moving object detection method considering the difference of image spatial threshold (named MOD-AT). First, based on the perspective characteristics of the camera and the smallest bounding rectangle of the object (dynamic block for short, or BLOB), a detection algorithm for the denoising threshold range of each pixel position is designed. On this basis, a moving object detection algorithm (named GMM_BLOB) was designed, and the corresponding relationship between the BLOB centroid and the pixel-position threshold range was constructed. Finally, the high-precision detection of moving objects based on the adaptive threshold was realized. The purpose of this study is to provide a new perspective for object detection, improve the accuracy and efficiency of object detection, and provide technical support for the integration of video and GIS.

3. Methodology

3.1. General Idea and Technical Process

A new dynamic object detection method, i.e., MOD-AT (Figure 1), is proposed in this paper. The method mainly focuses on adaptive threshold calculation based on camera-imaging characteristics and moving object detection based on the adaptive threshold.

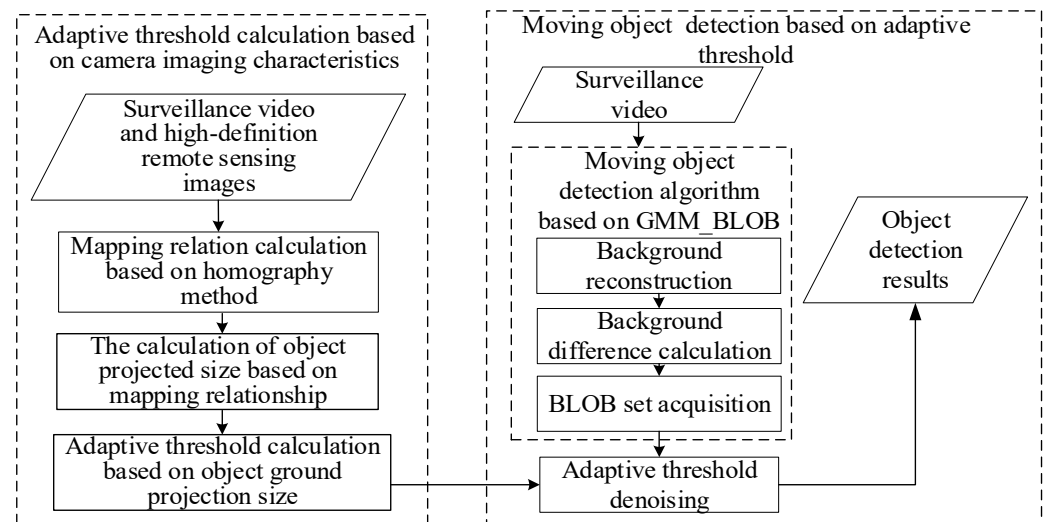


Figure 1. Overall process.

1. Adaptive threshold calculation based on camera-imaging characteristics

Previous research mostly adopted a single threshold to eliminate the noise caused by external-environment interference, resulting in low accuracy and low efficiency of moving object detection. We consider the orientation of moving objects in geographical space to obtain the adaptive threshold for a moving object. First, we calculate the homography matrix by selecting the homonymous points between the surveillance video and the online remote sensing image in the actual scene. Then, based on the homography, we establish the mapping relationship between the geometric BLOB in the image space and the smallest bounding rectangle in the geographic space. Finally, we calculate the actual size of the object based on the mapping relationship, and then obtain both the imaging area of the BLOB in image space and the range of adaptive threshold according to the projection size of the object.

2. Moving object detection based on adaptive threshold

In moving object detection, the existing object detection algorithms do not fully consider the perspective-imaging characteristics of the moving objects and the interference of external environmental noise. We abstract the irregular moving objects in image space as a set of minimum circumscribed rectangles (BLOBs). Using the above adaptive threshold calculation method, we design a moving object detection algorithm based on GMM_BLOB by background reconstruction, background-difference calculation, and moving-block (BLOB) set acquisition. We achieve high accuracy detection and noise removal of moving objects by adaptive threshold.

3.2. Adaptive Threshold Calculation Based on Camera Perspective Characteristics

3.2.1. Mapping Relation Calculation Based on Homography Method

For the convenience of threshold calculation, it is necessary to establish the mapping relationship between the actual orientation of the object in geographic space and the geometric-imaging characteristics in image space. Compared with the traditional mapping method based on the camera model, the homography method is simpler and does not require camera-internal and external parameters [39]. Therefore, we use the homography method to construct the mapping relationship between image space and geographical space.

First, four or more control points $q\{q_1(x_1, y_1), q_2(x_2, y_2), \dots, q_n(x_n, y_n)\}$ in the video are selected, and their pixel coordinates are obtained. Then, in the high-precision remote sensing image, the corresponding points $Q\{Q_1(X_1, Y_1), Q_2(X_2, Y_2), \dots, Q_n(X_n, Y_n)\}$ are selected to obtain the geographic coordinates. Finally, based on these control points, the

mapping matrix, H , of the camera video to geospatial mapping is obtained, according to Equation (1). The inverse matrix of H^{-1} is the mapping matrix from geographic space to image space.

$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = H^{-1} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2)$$

where (x, y) is the image coordinate of a point in the image space, (X, Y) is the geographic coordinates of the corresponding point (x, y) in geographic space, and H^{-1} is the inverse 3×3 matrix solved by the homography matrix, H .

3.2.2. Calculation of Object Projected Size Based on Mapping Relationship

To calculate the projection size of the object in geographical space, it is necessary to know the external parameters of the camera, i.e., the geographic location $C(X_{cam}, Y_{cam}, H_{cam})$ and the homography matrix, H . We can calculate the homography matrix, H , as in Section 3.2.1. However, for moving objects of different heights not in a plane or three-dimensional space, it is necessary to obtain the camera's internal parameters further, i.e., internal parameter matrix, K , rotation matrix, R , and translation matrix, T , as well as high-precision digital-elevation-model (DTM) and digital-surface-model (DSM) data. Considering that our existing data cannot obtain the camera's internal parameters and there are no high-precision DTM and DSM data of the camera area, we choose the flat video data and focus on the moving object on the flat surface. We assume that the video resolution is $i \times j$ and that the image coordinates of the corresponding pixel in the row, u ($0 \leq u \leq i - 1$), and column, v ($0 \leq v \leq j - 1$), are $C_{uv}(x, y, 0)$. According to the homography matrix, H , we transform the image-coordinate point, $C_{uv}(x, y, 0)$, into the geographic coordinate $R_{uv}(X, Y, 0)$.

Because of the different distances and orientations of the object from the camera during the process of the object moving, as a result, the geometric-imaging characteristics of the object at different pixel points in the image space are constantly changing. Therefore, it is necessary to obtain the projection length and width of the object's outer contour in geographic space based on the object-mapping relationship and then set different threshold ranges for each pixel position based on the camera-imaging characteristics. As shown in Figure 2, $C(X_{cam}, Y_{cam}, H_{cam})$ is the center of the camera in geographical space; the object dimensions are height, H_{uv} , width W_{uv}^l , and length, T_{uv} . The upper midpoint of the BLOB is the top point, and its geographic coordinates are $B_{uv}(X, Y, H_{uv})$. The lower midpoint of the BLOB is the touch point, and its geographical coordinates are $P_{uv}(X, Y, 0)$. Rays LT and LJ from the camera position, $C(X_{cam}, Y_{cam}, H_{cam})$, point to $B_{uv}(X, Y, H_{uv})$ and $P_{uv}(X, Y, 0)$, respectively. The angles between the ray and the ground are α and β . The object-ground projection area is a trapezoid, and its height, upper-side and lower-side lengths are calculated, which correspond to the object-ground projection length, h_{uv} , and ground projection width w_{uv}^u (one for near and one in the distance).

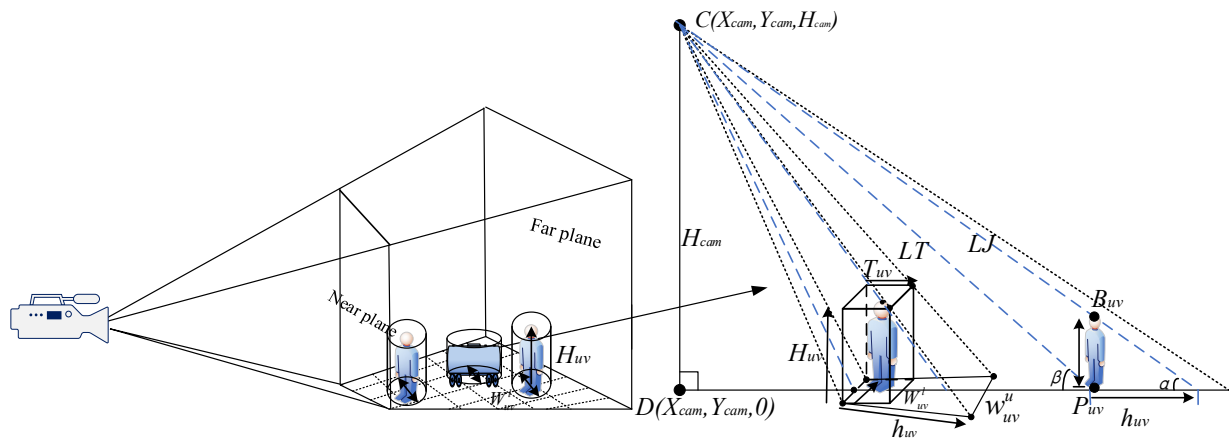


Figure 2. Illustration of object projection.

1. Calculation of the object's projected length, h_{uv} , in ground

According to the principle that the corresponding sides of similar triangles are proportional, we calculate the coordinates of the intersection point between the ray, LT , and the ground plane, $z = 0$, according to Equations (3) and (4). Then, the projection length, h_{uv} , of the object on the ground is calculated, as in Equation (5).

$$x = \frac{H_{cam}(X - X_{cam})}{H_{cam} - H_{uv}} \quad (3)$$

$$y = \frac{H_{cam}(Y - Y_{cam})}{H_{cam} - H_{uv}} + Y_{cam} \quad (4)$$

$$h_{uv} = \sqrt{(x - X)^2 + (y - Y)^2} \quad (5)$$

2. Calculation of the object's projected length, w_{uv}^u , in ground

During the object movement, the orientation of the moving object relative to the camera will change. On the horizon, however, the different directions of the object relative to the camera can be approximated as a cylinder. In geographic space, if the height and width of the cylinder do not change at a certain position, the projected width, w_{uv}^u , in geographic space will not change. As shown in Figure 3, assuming the geographic coordinates of the object (Obj) are $R_{uv}(X, Y, 0)$, the symbol of W_{uv}^l represents the width of the object projected to the ground. Two rays are drawn from the camera position, $C(X_{cam}, Y_{cam}, H_{cam})$, to points $J_L(X - \frac{W_{uv}^l}{2}, Y, H_{uv})$ and $J_R(X + \frac{W_{uv}^l}{2}, Y, H_{uv})$, respectively. Then, they intersect the ground plane at Geo_{uv}^3 and Geo_{uv}^4 , respectively. The linear formulas are CJ_L and CJ_R , as show in (6) and (7), and their intersection points with the ground plane are $Geo_{uv}^3(X^3, Y^3, 0)$ and $Geo_{uv}^4(X^4, Y^4, 0)$, respectively, i.e., Equations (8)–(11), and the values of Geo_{uv}^3 , Geo_{uv}^4 , and w_{uv}^u can be derived.

$$(x - X_{cam}) / (X - \frac{W_{uv}^l}{2} + T_{uv} - X_{cam}) = (y - Y_{cam}) / (Y - Y_{cam}) = (z - H_{cam}) / (H_{uv} - H_{cam}) \quad (6)$$

$$(x - X_{cam}) / (X + \frac{W_{uv}^l}{2} + T_{uv} - X_{cam}) = (y - Y_{cam}) / (Y - Y_{cam}) = (z - H_{cam}) / (H_{uv} - H_{cam}) \quad (7)$$

$$X^3 = \frac{H_{cam}(X - \frac{W_{uv}^l}{2} + T_{uv} - X_{cam})}{H_{cam} - H_{uv}} + X_{cam} \quad (8)$$

$$X^4 = \frac{H_{cam}(X + \frac{W_{uv}^l}{2} + T_{uv} - X_{cam})}{H_{cam} - H_{uv}} + X_{cam} \quad (9)$$

$$Geo_{uv}^2(X^2, Y^2) = Geo(X + \frac{w_{uv}^u}{2} * \sin \beta, Y + \frac{w_{uv}^u}{2} * \cos \beta) \quad (15)$$

$$Geo_{uv}^3(X^3, Y^3) = Geo(O_x + \frac{w_{uv}^u}{2} * \sin \beta, O_y + \frac{w_{uv}^u}{2} * \cos \beta) \quad (16)$$

$$Geo_{uv}^4(X^4, Y^4) = Geo(O_x - \frac{w_{uv}^u}{2} * \sin \beta, O_y - \frac{w_{uv}^u}{2} * \cos \beta) \quad (17)$$

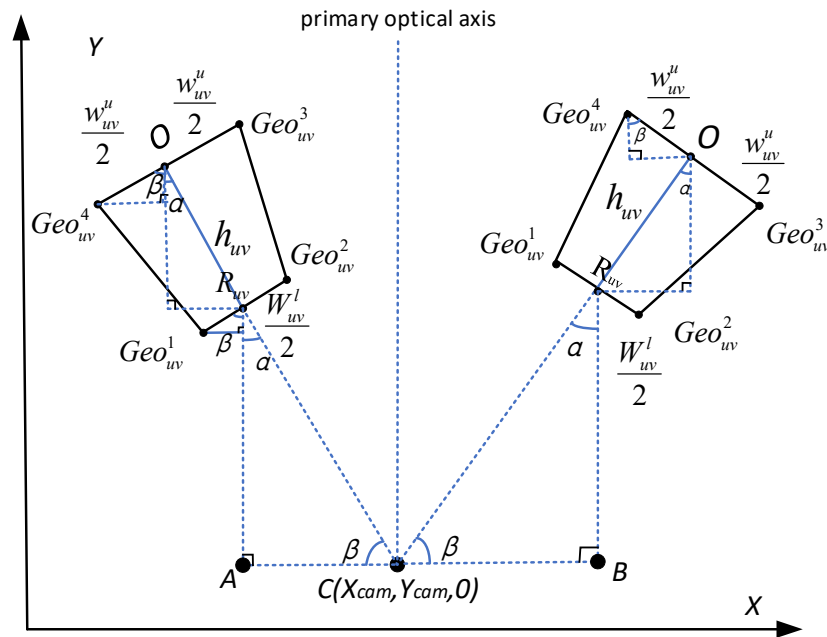


Figure 4. Illustration of Calculation of Object Projection Size.

2. Calculation of the area range of the object in the image plane

According to the algorithm presented in Section 3.2.1, the inverse matrix, H^{-1} , is obtained, and the image coordinates $Pic_{uv}^1(x^1, y^1, 0)$, $Pic_{uv}^2(x^2, y^2, 0)$, $Pic_{uv}^3(x^3, y^3, 0)$, and $Pic_{uv}^4(x^4, y^4, 0)$ on the video frame corresponding to $Geo_{uv}^1(X^1, Y^1, 0)$, $Geo_{uv}^2(X^2, Y^2, 0)$, $Geo_{uv}^3(X^3, Y^3, 0)$, and $Geo_{uv}^4(X^4, Y^4, 0)$, respectively, are solved. According to the coordinates of the four points on the video frame, the minimum area, S'_{uv} , of the external rectangle of the object on the image, that is, the minimum value, Min_{uv} , of the threshold, can be calculated, as shown in Equation (18). According to Section 3.2, the object size remains unchanged at the same position, but the size changes at different positions. In the experimental setting, the empirical threshold, Max_{uv} , is obtained after many experiments, as shown in Equation (19). the image coordinates of the object are $mPic_{uv}^1(mx^1, my^1, 0)$, $mPic_{uv}^2(mx^2, my^2, 0)$, $mPic_{uv}^3(mx^3, my^3, 0)$ and $mPic_{uv}^4(mx^4, my^4, 0)$.

As shown in Figure 5, when traversing each point within the field of view, the coordinates of the midpoint, the coordinates of the pixel, and the threshold range compose a set that can be expressed as Formula (20). This set contains the image coordinates, $C_{uv}(x, y, 0)$, the corresponding geographic coordinates, $R_{uv}(X, Y, 0)$, and the maximum, Max_{uv} , and minimum, Min_{uv} , threshold values of the object at the pixel Obj .

$$Min_{uv} = S'_{uv} = 1/2 = |x^1y^2 + x^2y^3 + x^3y^4 + x^4y^1 - x^2y^1 - x^3y^2 - x^1y^3 - x^1y^4| \quad (18)$$

$$Max_{uv} = \frac{1}{2|m x^1 m y^2 + m x^2 m y^3 + m x^3 m y^4 + m x^4 m y^1 - m x^2 m y^1 - m x^3 m y^2 - m x^1 m y^3 - m x^1 m y^4|} \quad (19)$$

$$R_{uv} = \{(x, y), (X, Y), Min_{uv}, Max_{uv}\} \quad (20)$$

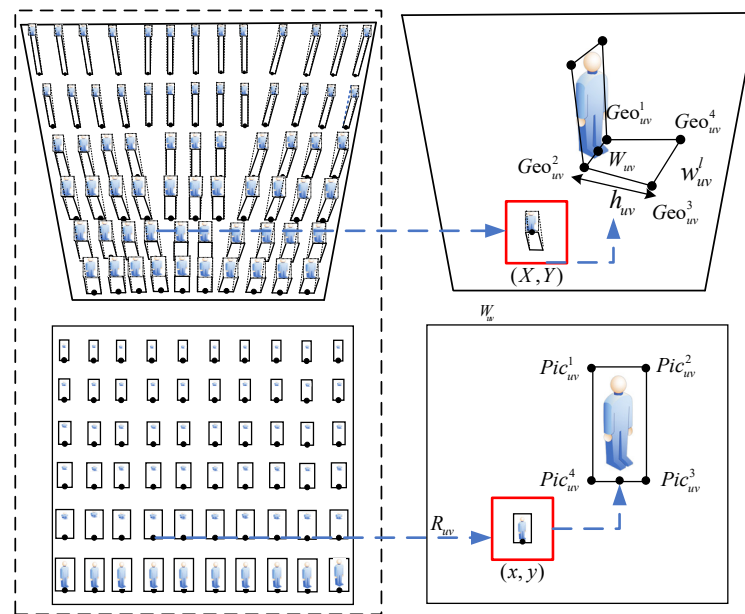


Figure 5. Threshold R_{uv} calculation.

3.3. Moving Object Detection Based on Adaptive Threshold

3.3.1. Adaptive Threshold Calculation and Application

The application of adaptive threshold in moving object detection mainly includes three processes: First, each pixel in the video frame is traversed to obtain its corresponding BLOB threshold range. Then, based on the center of the moving object BLOB, the relationship between its area and the pixel threshold range is judged. Finally, in the process of Obj_i movement, different threshold ranges are automatically used to filter interference from the external environment. As shown in Figure 6a, moving objects A, B, C, etc. are distributed in different locations in geographic space. The quadrilateral in Figure 6b is the region of these objects on the image plane. When the centroid of the moving object, A, is located at pixel R_5 , the relationship between the areas of object A and the range of the corresponding threshold at R_5 is judged. If the area of object A is less than the minimum threshold or greater than the maximum threshold, it is regarded as noise.

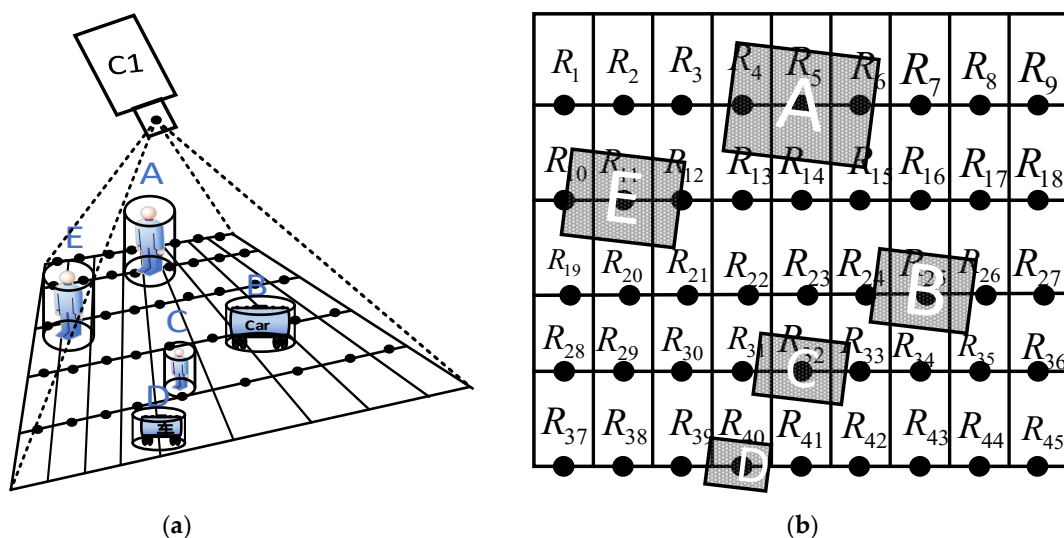


Figure 6. Design of adaptive threshold target-detection method (taking A, B, etc. as examples): Object in (a) geographical space and (b) in image space.

3.3.2. Moving Object Detection Based on GMM_BLOB Algorithm

The previous moving object detection algorithm mainly sets the dynamic threshold based on the depth map or the normalized pixel value without considering the adaptability of the threshold value in the process of the moving object. The key to object detection is to build a robust background image. The current background modeling methods are the GMM [40,41] and the Vibe algorithm [42–46]. Owing to parameter-setting and background-template updating problems, the vibe algorithm will lead to missed detection, residual shadow, and ghost phenomena in object detection. However, the traditional GMM method is slow and significantly affected by illumination. Adding a balance coefficient and merging redundant Gaussian distribution into the traditional GMM algorithm can improve the real-time performance and accuracy of the algorithm [47–50]. However, the traditional GMM algorithm ignores the influence of the external environment, which leads to an increase in the number of moving object detections. Meanwhile, the traditional object detection and tracking algorithms are slow, and it is difficult for them to meet the requirements of real-time surveillance video processing. As mentioned above, the improved object detection algorithm (GMM_BLOB) is designed. Based on the GMM algorithm, this algorithm abstracts the dynamic block BLOB of irregular moving objects in image space. It adds BLOB-threshold-filter conditions to improve the accuracy of moving object detection.

As shown in Figure 7, the foreground image $F(x, y, i)$, and background image, $B(x, y, i)$, of the video are extracted based on the background-mixture method [48]. On this basis, the background-subtraction method is used to extract the difference image, $R(x, y, i)$, and the candidate BLOB sets are further obtained. Owing to the BLOB set containing real moving objects and noise, it is necessary to further filter the noise in the BLOB set to improve the accuracy of object detection. According to the method detailed in Section 3.3.1, the imaging area and threshold range of each BLOB are also changing in the process of moving in geographic space. The relationship between the area of each BLOB in the candidate object set and the threshold range of its centroid pixel is determined. When the BLOB area is less than the minimum value of the threshold and greater than the maximum value, it is used as noise. Finally, the accurate moving object set, $BLOB_i (i = 1, 2, \dots, n)$, is obtained.

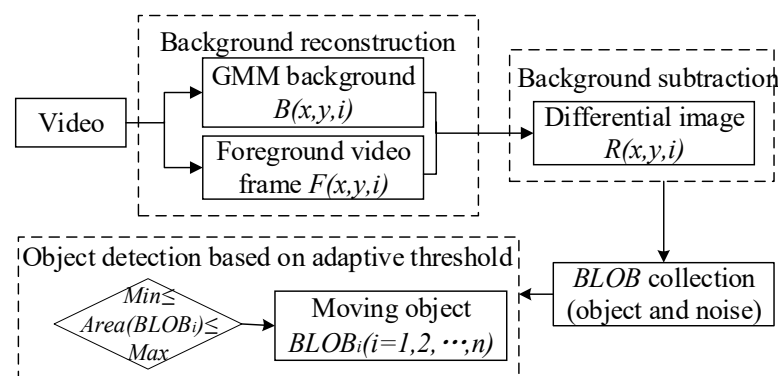


Figure 7. Dynamic target detection and BLOB moving object acquisition.

4. Experimental Test

4.1. Experimental Design

The software environment used in the experiments in this work is VS, C#, Emgu CV, and Arcengine, and the hardware environment is a GTX-1660Ti GPU and an i7-10750H CPU with 16 GB of memory. The experimental sites are a playground video (designated video #1) and a traffic scene video (video #2), as shown in Table 1. The videos are outdoor scenes, and the image resolution is 1280×720 . The differences between videos #1 and #2 are the following: (1) the moving object in video #1 contains only people, and video #2 contains people and cars; (2) the camera height of video #1 is higher than that of the video #2 corresponding camera, and the horizon is wider than that of video #2; and (3) the frame rate for video #1 is 25 frames per second (fps) and that for video #2 is 30 fps.

Precision, recall, and F1 score were used as the evaluation indexes to verify the accuracy of MOD-AT, and their calculation formulas are (21), (22), and (23), respectively. The test data for four moving object detection algorithms are recorded every 30 frames. The results of moving object detection were compared from two scales, namely single-frame accuracy and overall accuracy. At the same time, the mean precision (MP), mean recall (MR), mean F1-score (MF), variance precision (VP), variance recall (VR), variance F1 score (VF), and mean error number (MN) of the three indexes were used to evaluate the robustness of the algorithm. The calculation formulas are the following:

$$\text{precision} = \frac{TP}{TP + FP} \quad (21)$$

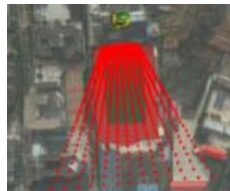
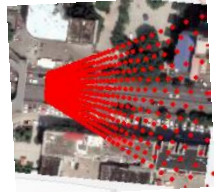
$$\text{recall} = \frac{TP}{TP + FN} \quad (22)$$

$$\text{F1 - score} = \frac{2 * \text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad (23)$$

$$\text{MN} = \sum_{i=1}^n \frac{|TN - TP|}{N} (n = 1, \dots, N) \quad (24)$$

where TP indicates that the foreground object is correctly identified, FP indicates the number of video backgrounds recognized as foreground objects, FN represents the number of foreground objects recognized as background, TN is the actual number of objects in the experimental scene, and N is the total number of detection frames.

Table 1. The experimental data.

Camera Name	X (m)	Y (m)	Altitude (m)	Video Length (min)	Video Resolution (pix)	Location and FOV
video #1	3,781,008.991	12,694,007.489	30 m	20	1280*720	
video #2	3,779,585.151	12,698,259.888	6 m	20	1280*720	

4.2. Adaptive Threshold Calculation

Using the algorithm presented in Section 3.2, the true threshold of a moving object at any position can be calculated. A moving object with a height of 1.75 m and a width of 0.8 m in the experimental scene was taken as an example. The threshold ranges of moving objects in videos #1 and #2 were calculated separately. To obtain the maximum threshold value in Section 3.2.3, we analyzed the size-change range of the object in videos #1 and #2 at different positions. As in Figure 8a,b, the width and height ranges of the object were set to $[W_{uv}^l, 2W_{uv}^l]$ and $[H_{uv}, 2H_{uv}]$, respectively, to obtain the corresponding threshold change ranges. After many experiments, results show that the size of the moving object changes within the corresponding thresholds of $[W_{uv}^l, 3/2W_{uv}^l]$ and $[H_{uv}, 3/2H_{uv}]$. Setting the threshold range too large or too small will affect the accuracy of moving object detection. Therefore, we chose 3/2 as the scale factor determined by the maximum threshold.

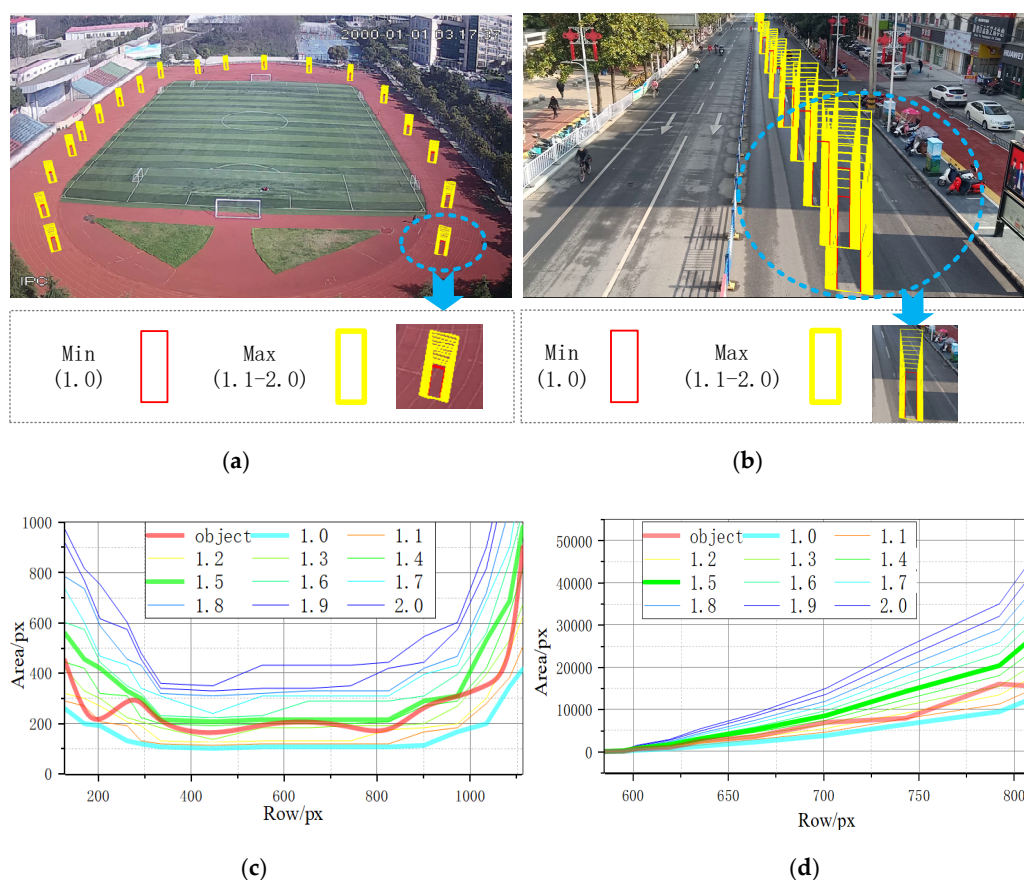


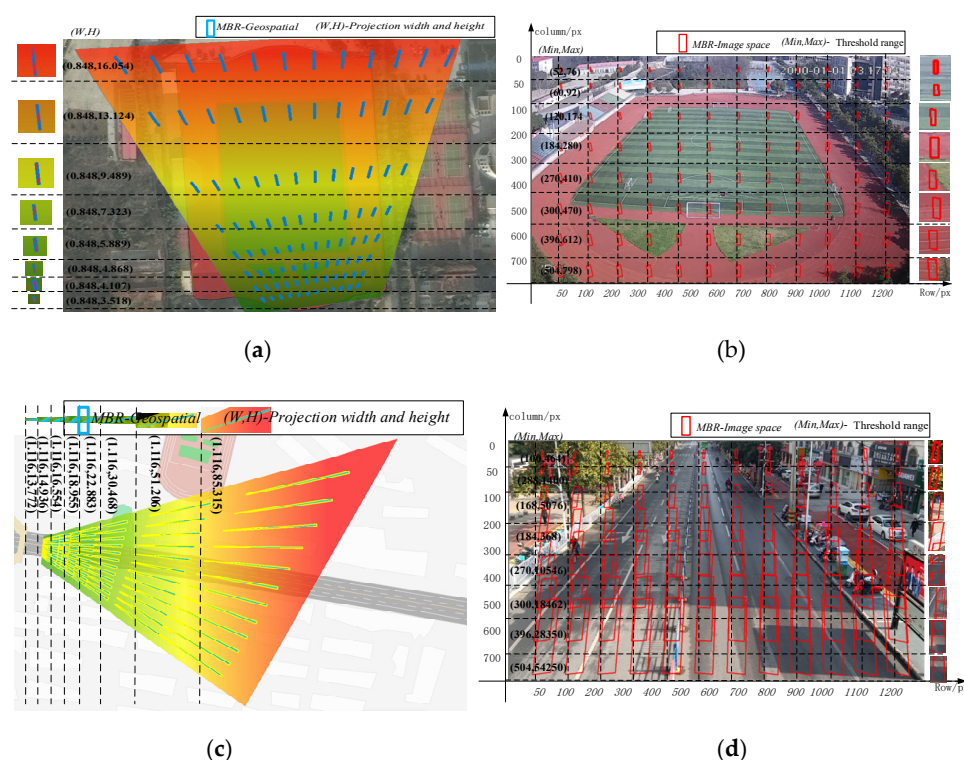
Figure 8. Maximum threshold range of videos #1 and #2: Object location and threshold range of (a) video #1 and (b) video #2. (c) Statistics of the width and height in video #1 are 1.0–2.0 times the threshold range and object-size change, respectively. (d) Statistics of the width and height in video #2 are 1.0–2.0 times the threshold range and the object-size change, respectively.

The algorithm detailed in Section 3.2 was used to obtain the threshold maximum and minimum values of the object (width, w_{uv}^u , and height, h_{uv}) at different pixel points. As shown in Table 2, (1) the projected height of the object is constantly changing in the process of moving in geographic space, and it is different from the actual height of the object; and (2) the threshold range of the object in different positions is constantly changing. Therefore, an adaptive threshold range should be used in the process of object detection.

The minimum peripheral contour of the object corresponding to each pixel of videos #1 and #2 is mapped to the corresponding position in the geographic space and image space, respectively. As shown in Figure 9a,b, the projection height and width of the object in the geographic space are used to realize the mapping from the geographic space to the image space. As shown in Figure 9c,d, the threshold range of each pixel-position object in the image space can realize moving object detection based on the adaptive threshold.

Table 2. Threshold information of videos #1 and #2.

Camera Name	Row (j)	Column (j)	Object Projection Width (m)	Object Projection Height (m)	Image Space Object Area (Number of Pixels)	Threshold Minimum (Number of Pixels)	Threshold Maximum (Number of Pixels)
video #1	527	21	1.13	5.17	60	60	299
	451	111	1.13	3.81	410	410	2013
	372	288	1.13	1.73	2185	2185	10,425
	248	422	1.13	1.25	4455	4455	20,544
	154	476	1.13	1.14	5624	5624	25,680
	338	56	0.85	1.46	56	56	80
video #2	276	148	0.85	1.06	108	108	156
	158	278	0.85	0.76	184	184	272
	54	370	0.85	0.64	208	208	312
	16	680	0.85	0.38	490	490	784

**Figure 9.** Threshold calculation of videos #1 and #2 in image space: (a) object size projected on the geographical space of video #1 and (b) image space threshold of video #1. (c) Object size projected on the geographical space of video #2 and (d) image space threshold of video #2.

4.3. Single-Frame Accuracy Verification

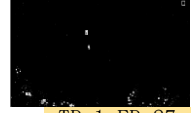
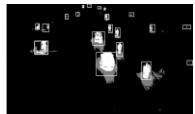
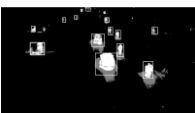
The object detection results were recorded by videos #1 and #2 every 30 frames, and 2400 records were obtained. At fixed intervals, 40 samples were randomly selected to verify the accuracy of a single frame. Table 3 shows the object detection results of video #1 at 9:20 and 9:22 and of video #2 at 9:40 and 9:42. This indicates that the TP value of the MOD-AT algorithm is closer to TN, and that the FP value is smaller than the existing object detection algorithm. It is proven that the MOD-AT algorithm eliminates most of the noise generated by the external environment, compared with the current algorithm.

The number of correctly detected objects (TP) is compared with real objects (TN). As shown in Figure 10a,c, the TP and TN values of the four detection algorithms at different time points are compared. The result shows that the moving object detection result, TP, of the MOD-AT algorithm is closer to TN. The box diagram in Figure 10b,d, shows the

average variability of the four algorithms, GVIBE [39], IGMM [34], NPCM [38], and MOD-AT, compared with the number of TN detections. These indicators show that

1. The mean value of object detection error (MD) is reduced by 1.2–6.8% for video #1 and by 1.65–11.05% for video #2.
2. The MN of MOD-AT object detection results for videos #1 and #2 decreases by 1.5–10.5% and 1–15.5%, respectively, on the whole; by 3.5–5% and 3–6.5%, respectively, compared with the IGMM algorithm; by 7–10.5% and 8–15.5%, respectively, compared with the GVIBE algorithm; and by 1.5–2% and 1–1.5%, respectively, compared with the NPCM algorithm. As shown in the randomly selected time points in Table 4, it can be seen that the precision, recall, and F1 score of the MOD-AT algorithm for single-frame detection are all higher than those of the GVIBE, IGMM, NPCM, and MOD-AT, indicating that the MOD-AT algorithm has high precision.

Table 3. Object detection results of videos #1 and #2.

Video Time	Video #1		Video #2	
	9:21	9:22	9:41	9:42
Video frame	 TN=2	 TN=2	 TN=15	 TN=17
Background				
GVIBE	 TP=1, FP=27	 TP=2, FP=5	 TP=10, FP=26	 TP=16, FP=11
IGMM	 TP=2, FP=9	 TP=2, FP=6	 TP=11, FP=29	 TP=16, FP=9
NPCM	 TP=2, FP=3	 TP=2, FP=2	 TP=13, FP=8	 TP=15, FP=3
MOD-AT	 TP=2, FP=0	 TP=2, FP=1	 TP=13, FP=1	 TP=16, FP=0

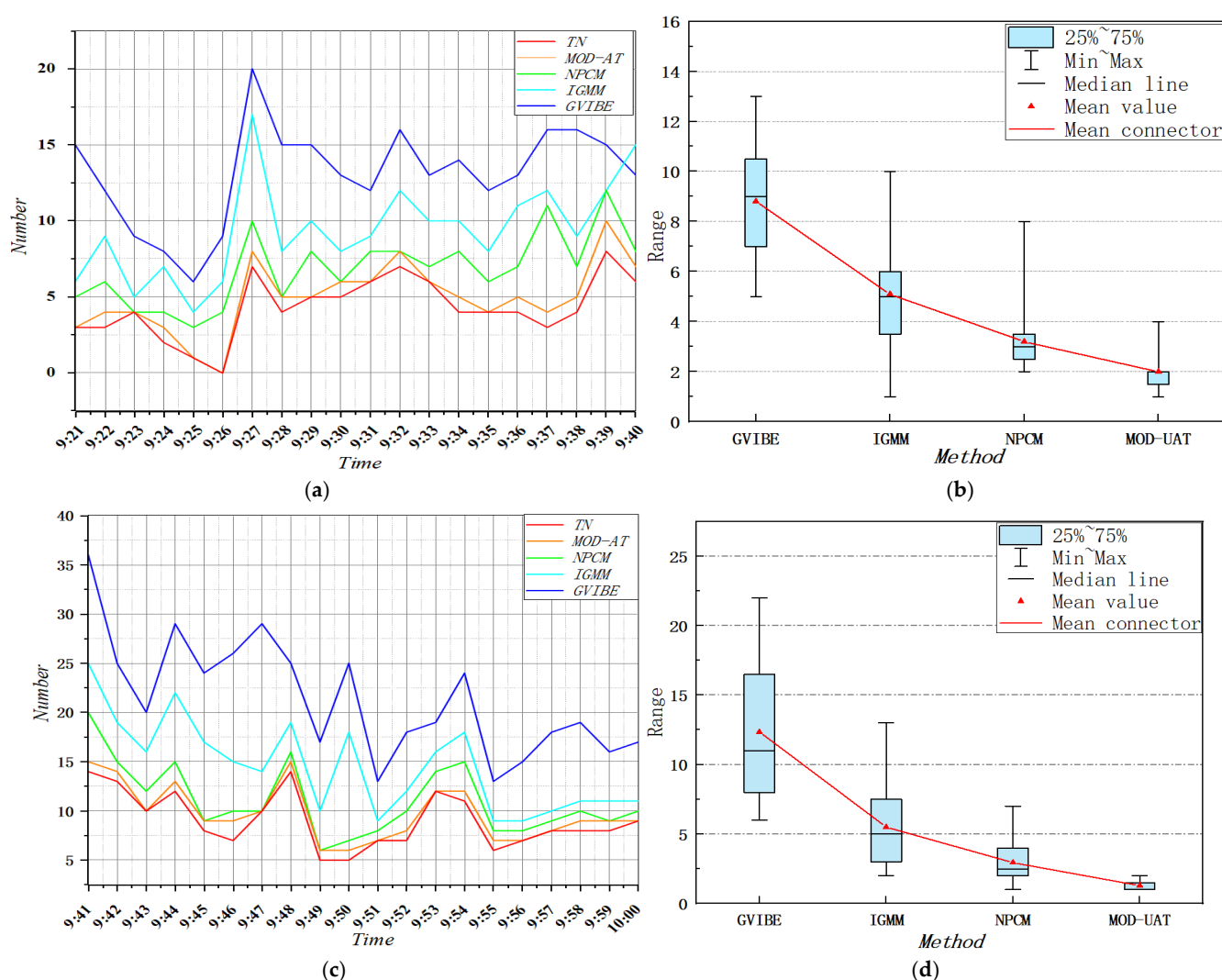


Figure 10. Experimental results of videos #1 and #2 for algorithm comparison: (a) comparison of number of correct detections for video #1; (b) TP and TN error box charts of video #1; (c) comparison of number of correct detections for video #2; (d) TP and TN error box charts of video #2.

Table 4. Comparison of single-frame indicators between videos #1 and #2.

Method Time	Video #1			Video #2		
	Precision	Recall	F1_Score	Precision	Recall	F1_Score
	9:20/9:30			9:40/9:50		
GVIBE	46.15/50	55.56/60	58.82/60	47.12/45	66.67/64.29	56/54.55
IGMM	40/57.89	77.78/71.43	68.75/53.85	58.33/60	75/77.78	63.23/63.16
NPCM	70/77.78	84.62/85.71	80/82.53	66.67/71.43	80/83.33	76.04/72.3
MOD-AT	90/89.9	96/87.5	87.50/85.71	90.91/89.66	90.23/90.89	88.89/94.74

4.4. Verification of Overall Accuracy

The overall accuracy of 2400 random records was verified, and then the precision, re-call, and F1-score were calculated for each frame. The results can be seen from the radar map distribution in Figure 11. The precision, recall, and F1-score of the MOD-AT algorithm are greater than those of the GVIBE, IGMM, NPCM, and MOD-AT algorithms. At the same time, the average value (MP) and variance (VR) of the calculation results of multiple frames of various indicators were obtained, and the results are shown in Table 5, as follows. (1) For different video data, the F1-score of the MOD-AT algorithm is

maintained above 90%, and the results show that MOD-AT algorithm can maintain good object detection performance for videos with different view and height. (2) Compared with other object detection algorithms, the MOD-AT algorithm improves MP by 17.13–44.4%, MR by 7.98–24.38%, and MF by 10.13–33.97%. (3) The MOD-AT algorithm VP, VR, and VF are lower than those of other object detection algorithms, showing that the precision, recall, and F1-score of the MOD-AT algorithm are stable and maintain high accuracy. Mean-while, the MOD-AT algorithm reduced the time consumption per frame by 7.9–24.3%, as shown in Table 6, indicating an optimized efficiency and performance. To verify the impact of videos with different frame rates on the algorithm's performance, the frame rates of videos #1 and #2 were separately converted to 10, 20, 30, 40, and 50 fps, respectively, and time efficiency and CPU memory consumption were compared. As can be seen in Figure 12, for videos with different frame rates, videos #1 and #2 show the following change patterns: (1) For every 10-fps increase in frame rate, CPU usage increases by 15.4–47.5% and time efficiency increases by 18.7–43.7%; and (2) for the same frame rate, the more targets each frame contains, the longer the processing time and the higher the CPU consumption.

Table 5. Comparison of indicators of MOD-AT algorithm.

Method	Video #1			Video #2		
	MP/VP	MR/VR	MF/VF	MP/VP	MR/VR	MF/VF
GVIIE	45.64/4.63	68.14/11.87	54.49/6.86	47.91/8.99	66.19/11.67	54.19/8.12
IGMM	42.24/10.1	73.11/6.41	53.14/9.22	54.98/8.92	73.78/9.10	62.75/6.86
NPCM	72.91/8.25	81.77/5.74	76.98/6.81	71.61/6.15	81.17/7.06	76.04/4.03
MOD-AT	90.04/4.29	89.75/4.06	87.11/4.03	91.13/4.32	90.57/3.73	87.52/3.42

Table 6. Comparison of MOD-AT efficiency and performance.

Method	Time Efficiency (ms/frame ^{−1})	Threshold Setting			Adaptive Threshold for Object Detection
		Geospatial	Dynamic Threshold	Object Type	
GVIIE [39]	60.653	No	No	No	No
IGMM [34]	45.951	No	No	No	No
NPCM [38]	40.653	Yes	Yes	No	Yes
MOD-AT	37.439	Yes	Yes	Yes	Yes

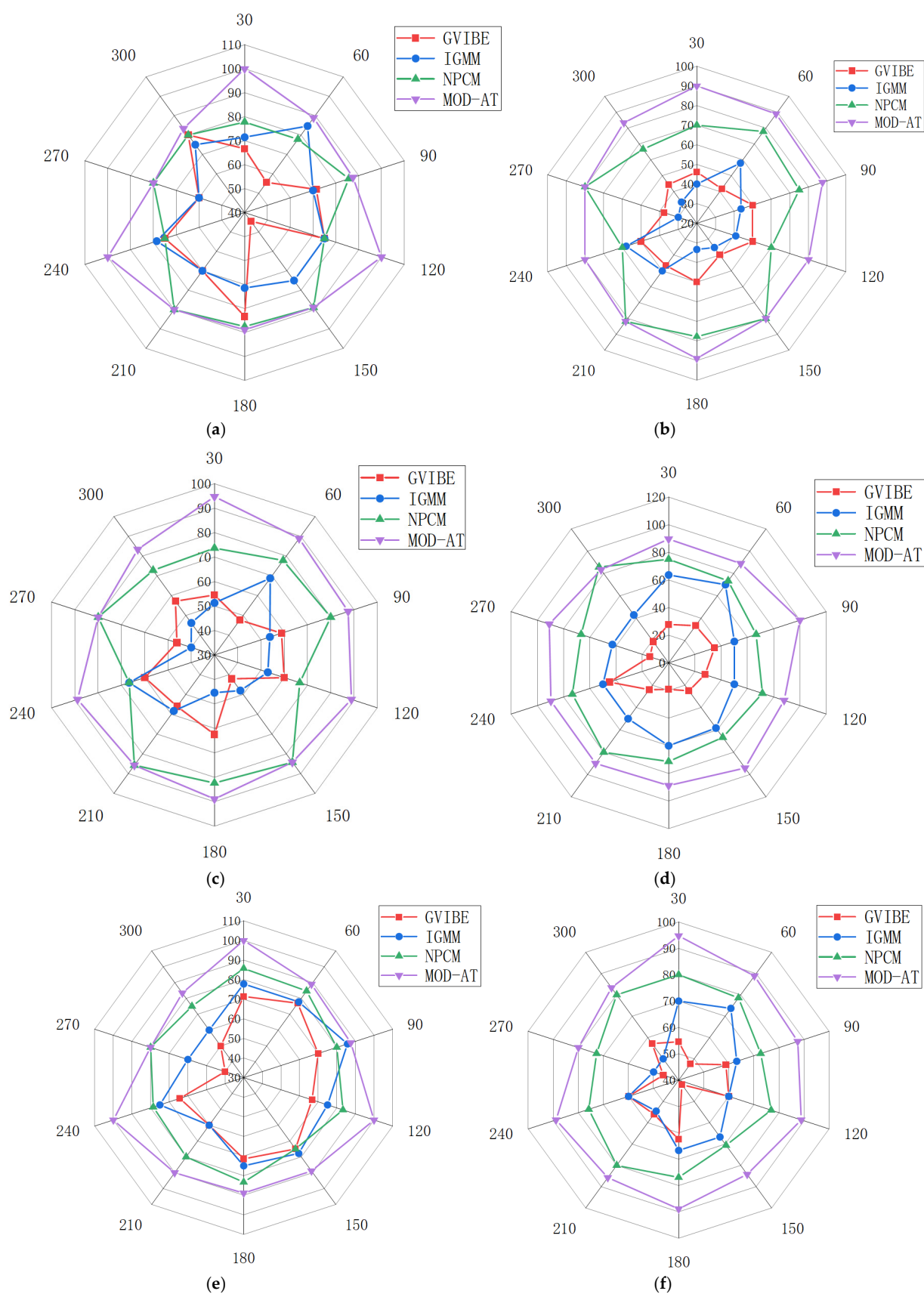


Figure 11. Comparison of MOD-AT and other algorithms: (a) the precision comparison of video #1; (b) the recall comparison of video #1; (c) the F1 score comparison of video #1; (d) the precision comparison of video #2; (e) the recall comparison of video #2; (f) the F1 score comparison of video #2.

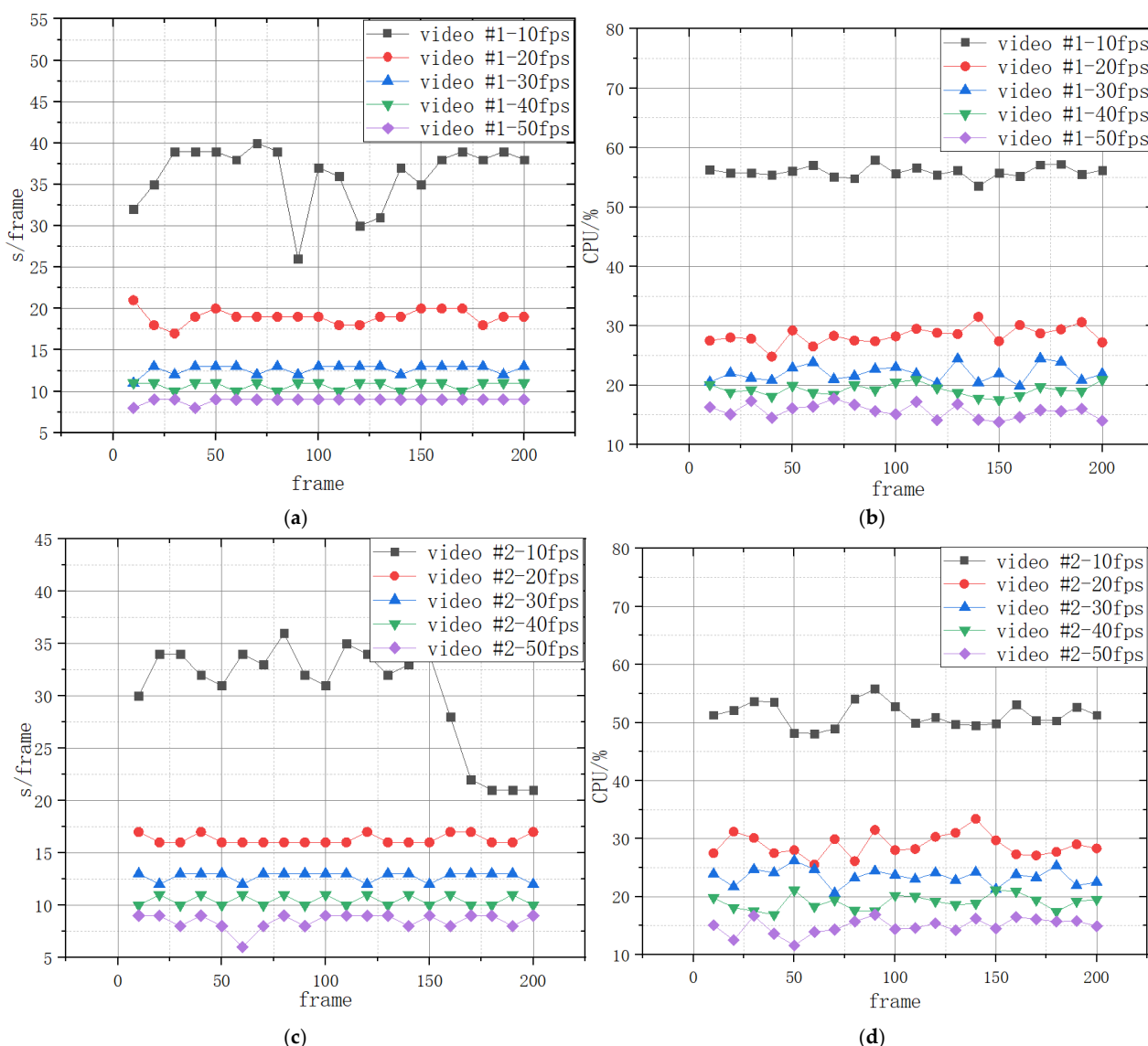


Figure 12. Algorithm performance of videos #1 and #2 at different frame rates: (a) the time efficiency of video #1; (b) the CPU occupancy percentage of video #1; (c) the time efficiency of video #2; (d) the CPU occupancy percentage of video #2.

5. Conclusions and Discussion

Aiming at the problem that the previous moving object detection algorithm does not consider the influence of camera-imaging characteristics, resulting in low target-detection accuracy, a moving object detection method called MOD-AT that considers the difference in image spatial threshold was designed in this work. MOD-AT realizes the high-precision detection of moving objects at different positions on the horizon according to different thresholds.

Experimental results show that the MOD-AT algorithm has higher accuracy in both single-frame and overall accuracy evaluation. In the aspect of single-frame accuracy, we report the following.

1. Compared with the existing object detection algorithm, the median error (MD) of the MOD-AT algorithm is reduced by 1.2–11.05%.
2. The mean error (MN) of the MOD-AT object detection results is reduced by 1–15.5%, which shows that the MOD-AT algorithm has high accuracy in single-frame detection. In terms of overall accuracy, (a) the results show that the F1 score of the MOD-AT algorithm is above 90% for different experimental scenarios, demonstrating

the stability of the MOD-AT algorithm; and (b) compared with the existing object detection algorithms, the MOD-AT algorithm improves MP by 17.13–44.4%, MR by 7.98–24.38%, and MF by 10.13–33.97%, which shows that the MOD-AT algorithm has high precision.

3. The MOD-AT algorithm performance was improved by 7.9–24.3% compared to other algorithms, reflecting its efficiency.

Of course, this algorithm has several shortcomings. For example, due to the limitation of video data and the difficulty of obtaining high-precision DTM and DSM data, the threshold calculation is limited to moving objects on a plane. At the same time, multiple experiments were carried out on the two videos to determine the maximum threshold range. More experimental data are needed to verify the universality of the maximum threshold range. In addition, it is important to note that for more complex monitoring scenarios, such as group objects, current algorithms need to consider object detection, semantic segmentation, deep learning, and other methods. In conclusion, how to further coordinate the efficiency of the current method with the high accuracy of a deep-learning method is also a problem that needs further study. These problems remain the focus of planned follow up research.

Author Contributions: Xiaoyue Luo established the unique idea and methodology for the current study; Yanhui Wang contributed guidance on paper presentation and logical organization; Benhe Cai and Zhanxing Li provided review and editing of manuscript articles. All authors have read and agreed to the published version of the manuscript.

Funding: National Natural Science Foundation of China (42171224, 41771157), National Key R&D Program of China (2018YFB0505400), the Great Wall Scholars Program (CIT&TCD20190328), Key Research Projects of National Statistical Science of China (2021LZ23), Young Yanjing Scholar Project of Capital Normal University, and Academy for Multidisciplinary Studies, Capital Normal University.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The author appreciates the editors and reviewers for their comments, suggestions, and valuable time and efforts in reviewing this manuscript. At the same time, thanks to Zhigang Yang of the Department of Surveying and Mapping of the Department of Land and Resources of Guangdong Province for modifying the English language of the paper.

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

References

1. Yong, W.W. Video Specialization Method and Its Application. *J. Nanjing Norm. Univ.* **2014**, *37*, 119–125.
2. Du, B.; Zhang, L. Target detection based on a dynamic subspace. *Pattern Recognit.* **2014**, *47*, 344–358. [[CrossRef](#)]
3. Cheng, B.; Zhao, C. A new anomaly target detection algorithm for hyperspectral imagery based on optimal band subspaces. *J. Appl. Sci. Eng.* **2020**, *23*, 213–224.
4. Sun, W.; Du, H.; Ma, G.; Shi, S.S.; Zhang, X.R.; Wu, Y. Moving vehicle video detection combining Vibe and inter-frame difference. *Int. J. Embed. Syst.* **2020**, *12*, 371–379. [[CrossRef](#)]
5. Bouwmans, T. Traditional and recent approaches in background modeling for foreground detection: An overview. *J. Comput. Sci. Rev.* **2014**, *11*, 31–66. [[CrossRef](#)]
6. Zhang, X.G.; Liu, X.J.; Song, H.Q. Video surveillance GIS: A novel application. In Proceedings of the 21st International Conference on Geoinformatics, Kaifeng, China, 20–22 June 2013.
7. Li, X.P.; Lei, S.Z.; Zhang, B.; Yanhong, W.; Feng, X. Fast aerial ova detection using improved inter-frame difference and svm. *J. Phys. Conf. Ser.* **2019**, *1187*, 032082.
8. Song, H.Q.; Liu, X.J.; Lv, G.N.; Zhang, X.G.; Feng, W. Video scene invariant crowd density estimation using geographic information systems. *J. China Commun.* **2014**, *11*, 80–89.
9. Shao, Z.; Li, C.; Li, D.; Orhan, A.; Lei, Z.; Lin, D. An accurate matching method for projecting vector data into surveillance video to monitor and protect cultivated land. *ISPRS Int. J. Geo. Inf.* **2020**, *9*, 448. [[CrossRef](#)]

10. Amrutha, C.V.; Jyotsna, C.; Amudha, J. Deep learning approach for suspicious activity detection from surveillance video. In Proceedings of the 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 5–7 March 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 335–339.
11. Elharrouss, O.; Almaadeed, N.; Al-Maadeed, S.; Bouridane, A.; Beghdadi, A. A combined multiple action recognition and summarization for surveillance video sequences. *Appl. Intell.* **2021**, *51*, 690–712. [\[CrossRef\]](#)
12. French, G.; Mackiewicz, M.; Fisher, M.; Holah, H.; Kilburn, R.; Campbell, N.; Needle, C. Deep neural networks for analysis of fisheries surveillance video and automated monitoring of fish discards. *ICES J. Mar. Sci.* **2020**, *77*, 1340–1353. [\[CrossRef\]](#)
13. Balasundaram, A.; Chellappan, C. An intelligent video analytics model for abnormal event detection in online surveillance video. *J. Real Time Image Process.* **2020**, *17*, 915–930. [\[CrossRef\]](#)
14. Gracewell, J.; John, M. Dynamic background modeling using deep learning auto encoder network. *Multimed. Tools Appl.* **2020**, *79*, 4639–4659. [\[CrossRef\]](#)
15. Qin, W.; Zhu, X.L.; Xiao, Y.P.; Yan, J. An Improved Non-Parametric Method for Multiple Moving Objects Detection in the Markov Random Field. *Comput. Model. Eng. Sci.* **2020**, *124*, 129–149.
16. Singla, N. Motion detection based on frame difference method. *Int. J. Inf. Comput. Technol.* **2014**, *4*, 1559–1565.
17. Colque, R.V.H.M.; Caetano, C.; Andrade, M.T.L.; Schwartz, W.R. Histograms of optical flow orientation and magnitude and entropy to detect anomalous events in videos. *IEEE Trans. Circuits Syst. Video Technol.* **2016**, *27*, 673–682. [\[CrossRef\]](#)
18. Gannakeris, P.; Kaltsa, V.; Avgerinakis, K.; Briassoulis, A.; Vrochidis, S.; Kompatsiaris, I. Speed estimation and abnormality detection from surveillance cameras. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 93–99.
19. Liu, H.B.; Chang, F.L. Moving object detection by optical flow method based on adaptive weight coefficient. *Opt. Precis. Eng.* **2016**, *24*, 460–468.
20. Fan, L.; Zhang, T.; Du, W. Optical-flow-based framework to boost video object detection performance with object enhancement. *Expert Syst. Appl.* **2020**, *170*, 114544. [\[CrossRef\]](#)
21. Xing, X.; Yong, J.Y.; Huang, X. Real-Time Object Tracking Based on Optical Flow. In Proceedings of the 2021 International Conference on Computer, Control and Robotics (ICCCR), Shanghai, China, 8–10 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 315–318.
22. Wu, B.; Wang, C.; Huang, W.; Huang, P. Recognition of Student Classroom Behaviors Based on Moving Target Detection. *J. Treat. Signal* **2021**, *38*, 215–220.
23. Li, X.; Zheng, H. Target detection algorithm for dance moving images based on sensor and motion capture data. *Microprocess. Microsyst.* **2021**, *81*, 103743. [\[CrossRef\]](#)
24. Zuo, J.H.; Jia, Z.H.; Yang, J.; Nikola, K. Moving object detection in video image based on improved background subtraction. *Comput. Eng. Des.* **2020**, *41*, 1367–1372.
25. Luo, M.; Liu, D.B.; Wen, H.X.; Chen, X.T.; Song, D. A New Vehicle Moving Detection Method Based on Background Difference and Frame Difference. *J. Human Inst. Eng.* **2019**, *29*, 58–61.
26. Akhter, I.; Jalal, A.; Kim, K. Pose estimation and detection for event recognition using sense-aware features and Adaboost classifier. In Proceedings of the 2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST), Islamabad, Pakistan, 12–16 January 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 500–505.
27. Li, Y.; Wu, J.; Bai, X.; Yang, X.; Tan, X.; Li, G.; Zhang, H.; Ding, E. Multi-granularity tracking with modularized components for unsupervised vehicles anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 586–587.
28. Liu, Z.L.; Wu, Y.Q.; Zou, Y. Multiscale infrared superpixel-image model for small-target detection. *J. Image Graph.* **2019**, *24*, 2159–2173.
29. Xin, Y.X.; Shi, P.F.; Xue, Y.Y. Moving Object Detection Based on Region Extraction and Improved LBP Features. *Comput. Sci.* **2021**, *48*, 233–237.
30. Heikkilä, M.; Pietikäinen, M.A. Texture-Based Method for Modeling the Background and Detecting Moving Objects. *IEEE Trans Pattern Anal. Mach. Intell.* **2006**, *28*, 415–423. [\[CrossRef\]](#)
31. Kong, G. Moving Object Detection Using a Cross Correlation between a Short Accumulated Histogram and a Long Accumulated Histogram. In Proceedings of the IEEE 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; pp. 896–899.
32. Luo, D. Texture Analysis for Shadow Removing and Tracking of Vehicle in Traffic Monitoring System. In Proceedings of the 2008 International Symposium on Intelligent Information Technology Application Workshops, Shanghai, China, 21–22 December 2008; pp. 863–866.
33. Liu, W.; Hao, X.L.; Lv, J.L. Efficient moving targets detection based on adaptive Gaussian mixture modelling. *J. Image Graph.* **2020**, *25*, 113–125.
34. Zuo, J.; Jia, Z.; Yang, J.; Kasabov, N. Moving Target Detection Based on Improved Gaussian Mixture Background Subtraction in Video Images. *IEEE Access* **2019**, *29*, 152612–152623. [\[CrossRef\]](#)
35. Zhang, X.G.; Liu, X.J.; Wang, S.N.; Liu, Y. Mutual Mapping between Surveillance Video and 2D Geospatial Data. *Geomat. Inf. Sci. Wuhan Univ.* **2015**, *40*, 1130–1136.

36. Chan, A.B.; Liang, Z.-S.J.; Vasconcelos, N. Privacy preserving crowd monitoring: Counting people without people models or tracking. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–7.
37. Chang, Q.L.; Xia, H.S.; Li, N. A Method for People Counting in Complex Scenes Based on Normalized Foreground and Corner Information. *J. Electron. Inf. Tech.* **2014**, *36*, 312–317.
38. Lin, B.X.; Xu, C.H.; Lan, X.; Zhou, L. A method of perspective normalization for video images based on map data. *Ann. GIS* **2020**, *26*, 35–47. [[CrossRef](#)]
39. Xie, Y.; Wang, M.; Liu, X.; Mao, B.; Wang, F. Integration of Multi-Camera Video Moving Objects and GIS. *ISPRS Int. J. Geo. Inf.* **2020**, *8*, 561. [[CrossRef](#)]
40. Sun, W.; Chen, S.; Shi, L.; Li, Y.; Lin, Z. Vehicle Following in Intelligent Multi-Vehicle Systems Based on SSD-MobileNet. In Proceedings of the Chinese Automation Congress (CAC), Hangzhou, China, 22–24 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 5004–5009.
41. Chen, L.; Zhang, R.G.; Hu, J.; Liu, K. Improved Gaussian Mixture Model and Shadow Elimination Method. *J. Comput. Appl.* **2013**, *33*, 1394–1400. [[CrossRef](#)]
42. Zhang, L.; Shi, L.; Zhang, L. Adaptive threshold moving target detection algorithm based on Vibe. *Comput. Inf. Technol.* **2021**, *29*, 12–15.
43. Zhu, S.F.; Yang, F.; Ma, W. Difference Method Target Detection Algorithm Based on Vibe Model and Five-Frame. *Comput. Digit. Eng.* **2020**, *48*, 667–670.
44. Liu, Y.Z.; Zhang, H.B.; Yang, J.; Tang, T.G.; Tan, H.Y. A new algorithm of moving target tracking and detection based on optical flow techniques with frame difference methods. *Electron. Des. Eng.* **2021**, *29*, 139–144.
45. Xiao, Y.Q.; Yang, H.M. Research on Application of Object Detection Algorithm in Traffic Scene. *Comput. Eng. Appl.* **2021**, *57*, 30–41.
46. Barata, C.; Nascimento, J.C.; Marques, J.S. Multiple agent's representation using motion fields. In Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 2007–2011.
47. Li, X.; Yang, Y.; Xu, Y.M. Detection of moving targets by four-frame difference in modified Gaussian mixture Model. *Sci. Technol. Eng.* **2020**, *20*, 6141–6150.
48. Xie, Y.J.; Wang, M.Z.; Liu, X.J.; Wang, Z.; Mao, B.; Wang, F.; Wang, X. Spatiotemporal retrieval of dynamic video object trajectories in geographical scenes. *Trans. GIS* **2020**, *25*, 450–467. [[CrossRef](#)]
49. Yang, X.; Li, F.; Lu, M. Moving Object Detection Method of Video Satellite Based on Tracking Correction Detection. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2020**, *3*, 701–707. [[CrossRef](#)]
50. Yao, P.; Zhao, Z. Improved Glasius bio-inspired neural network for target search by multi-agents. *Inf. Sci.* **2021**, *568*, 40–53. [[CrossRef](#)]