Journal of Computers Vol. 32 No. 1, 2021, pp. 53-64
doi:10.3966/199115992021023201005

Ghost Suppression Algorithm Based on TFDT Modeling

Guo-Liang Xu, Hang Zhou*, Liang-You Yuan

School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
{18120027, hangzhou, 19125065}@bjtu.edu.cn

Received 4 November 2019; Revised 7 April 2020; Accepted 29 May 2020

Abstract. In the background modeling, moving objects would often enter the scene. "Ghost" emerges as the pixels of interfering objects damage the background model. This paper presents a "ghost" suppression algorithm based on two - frame difference method and topology (TFDT). First in TFDT algorithm is to obtain the change mask with two-frame-difference method detecting the difference between two adjacent frames. Then according to the change mask, the complete object region and its bounding box are estimated with topo-logical structure. Finally, the pixels in the bounding box are eliminated, and the remaining pixels are added to the background model. TFDT is a universal algorithm for background modeling and can be adapted in foreground detection algorithms such as Codebook and ViBe. It can effectively remove object's pixels and build a background model for foreground detection algorithms. In this work, TFDT is used to improve Codebook and ViBe, and the "ghost" is absent from the result, which has been verfied in the CDnet database.

Keywords: background modeling, two-frame difference, topological structure, foreground detection, universal algorithm

1 Introduction

As a combination of traditional video technology and modern communication technology, Video surveillance has attracted more and more attention at home and abroad. The intelligentization of video surveillance has become a development trend. The foreground detection through video camera has become one of the most important applications of computer vision, which has important practical value and broad application prospect in the fields of security, feature detection and tracking, product detection and military, etc. [1-2].

According to different ways of modeling, foreground detection algorithms can be divided into foreground modeling algorithms, background modeling algorithms and other algorithms. Foreground modeling algorithm is to realize the foreground detection by using the database-trained detection model, which is mainly algorithm based on deep learning. Foreground modeling algorithm can get the characteristics of the foreground object effectively through the training of the database and can adapt to the complex background environment. However, the disadvantage is that a large amount of data is required for training, and the detected object needs to be trained. That is not suitable for detecting unknown abnormal objects. The background modeling algorithm uses the background pixels to build the background model, and then updates it to simulate the real background. Foreground can be obtained by the difference of background model. The background modeling algorithm is simple and effective, which does not require database training and can detect all foreground outside the background model. However, the disadvantage is that the complex background can affect the result of detection. When there are object's pixels in the background model, "ghost" is attendant in the detection, resulting in false detection. Other algorithms mainly resort to the foreground motion and optical flow properties to realize the foreground detection. The results of such methods are usually affected by the foreground motion state and background noise.

* Corresponding Author

The background modeling has been favored for its simplicity and effectiveness. But “ghost” has always been a problem for background modeling algorithms. “Ghost” is a false object, which is to detect the background area as the object area. In order to ensure the accuracy of detection results, the elimination of “ghost” is a crucial problem. In most cases, “ghost” is caused by the background model containing the object’s pixels. Therefore, this paper proposes a “ghost” suppression algorithm called TFDT, which uses the two-frame difference method and topological structure to build the background model, and can effectively solve the problem of “ghost”.

This paper introduces related works and our contributions in the second section; the introduction of TFDT in the third section; the algorithm improvement of Codebook and ViBe in the fourth section; the experimental results and analysis in the fifth section; the summary and prospect in the sixth section.

2 Related Works and Our Contributions

For foreground detection, many researches have been done at home and abroad.

Foreground modeling algorithms: At present, the deep learning needs a lot of data for training to obtain the foreground detection model. Algorithms can be divided into algorithm based on region suggestion (e.g., Faster r-cnn) [3], algorithm based on object regression (e.g., YOLO v3) [4], algorithm based on search (e.g., AttentionNet) [5], and algorithm based on anchor-free (e.g., CornerNet) [6]. By using a large amount of data for training, the deep network can effectively detect the trained object.

Background modeling algorithms: Background modeling method is simple and effective. Roy S M et al. introduce a variety of background modeling algorithms [7]. At present, the commonly used background modeling methods are GMM, Codebook and ViBe. Zivkovic Z et al. use GMM to model the complex background pixels [8]. The parameter K of GMM to be adaptive according to the change of the background, which reduces some of the computation. Kim K et al. model each pixel with a Codebook containing multiple code elements [9]. Codebook can filter the object’s pixels in the model after the background model is initialized. Because of their complex modeling methods, GMM and Codebook algorithms are inefficient. Barnich O et al. propose the ViBe, which opens up a new idea of background modeling [10]. ViBe uses the first frame modeling and random thought to realize the foreground detection. StCharles P L et al. propose SuBSENSE and PAWCS algorithm by referring to ViBe’s idea and combining with LBSP (Local Binary Similarity Pattern, LBSP) [11-12]. The latter two algorithms obtain better detection results.

In the process of background modeling, the objects often appear. Background modeling can be affected by the object motion. The variability of object motion, such as stationary, slow and fast motion, increases the difficulty of background modeling. When the moving object is stationary or moving slowly, few or no background pixels will be collected in the object position, and the background model can mistake the object’s pixels for the background pixels, which will lead to “ghost” in the foreground segmentation. So, many researchers have improved the background modeling algorithms. Kanungo P et al. improved the Codebook [13-15], but the improved algorithm could not work out object motion variability. Xu jiuqiang and Yang Dan improved ViBe by using the two-frame difference method and multi-frame average method respectively [16-17]. These two algorithms improved the background initialization effect to some extent, but still cause “ghost” in the case of slow and stationary object movement.

Other algorithms: Inter-frame difference algorithms and optical flow algorithms are often used for foreground detection. Inter-frame difference algorithms include two-frame difference algorithm and three-frame difference algorithm. Optical flow algorithms include dense optical flow algorithm and sparse optical flow algorithm [18]. The detection results of these algorithms are usually affected by foreground motion and background noise.

In order to suppress “ghost” effectively, this paper proposes the TFDT algorithm for background modeling. The main contributions of this paper are as follows:

1. Since the interfering object’s pixels in the background model will produce “ghost” in the detection, this paper focuses on eliminating the object’s pixels from the background model. This paper proposes a “ghost” suppression algorithm called TFDT, which uses the two-frame difference method and topological structure to build the background model, and can effectively solve the problem of “ghost”.

2. In TFDT, the three-channel-two-frame difference method is used to detect the motion of the object, and then according to the motion information, the object’s pixels are removed by using the topological

structure. Compared with other algorithms, TFDT has better robustness to the diversity of object motion in the modeling process.

3. TFDT is a universal algorithm. It can be used with foreground detection based on background modeling method.

3 TFDT

To better illustrate the problem, the definition of smooth, slow and fast motion are given. In this paper, stationary object means that the object swings around the central axis in the swing Angle $[10^{\circ}, 90^{\circ}]$. Slow motion is to judge whether the object motion is half body difference between two adjacent frames: if the motion is less than half the body position, it can be taken as slow motion; if the movement is greater than half the body position, it can be taken as fast motion. TFDT will be introduced in detail below. Firstly, it uses three-channel-two-frame difference method to detect the changes between two adjacent frames. The purpose is to obtain the change mask containing motion information (see section 3.1). Then according to the change mask, the complete regions are obtained by topology estimation, and further get its bounding boxes (see section 3.2). Finally, the pixels in the bounding boxes are removed, and the remaining pixels are added to the model to obtain the background model without interfering object's pixels (see section 4). The following Algorithm (1) introduces the pseudo-code.

Algorithm (1):

1. S←true;
2. while S do
3. Calculate the difference image of two adjacent frames by three-channel-two-frame method;
4. The difference images of the three channels are binarized respectively;
5. The three channels are taken or operated to obtain change mask;
6. Cluster the mass point of domains in the change mask by MMD+;
7. The topological structure is generated in the same kind of mass points, and the complete region of the object is estimated by the topological structure, further get its bounding boxes;
8. Remove the pixels in the bounding boxes and add the remaining pixels to the background model;
9. if (At least one pixel is collected at each position in the background model)
10.   S←false;
11. end if
12. end while

3.1 Three-Channel-Two-Frame Difference Method

Since the video sequence is fast and continuous, there is little difference in natural factors such as light change between adjacent two frames. If the object moves between two adjacent frames, it can be detected sensitively by Two-frame difference method.

Two-frame difference method is to get the difference image by subtracting the corresponding pixels of two adjacent frames. Then the threshold processing is carried out to obtain the change mask. The traditional Two-frame difference method takes the gray image as the input image, which lose lots of color information. The three-channel-two-frame difference method takes the color image as input image. By comparing with the two methods in Fig. 1, it can be found that the three-channel-two-frame difference method retains more information of image changes.

In TFDT, each of the three channels performs difference operation to obtain difference images according to formula (1). Then according to formula (2), the difference images of the three channels are binarized to obtain the three-channel change mask. Finally according to formula (3), the three-channel mask is performs OR operated to obtain the final change mask. Three-channel-two-frame difference method is suitable for any three-channel color space, such as RGB, HSV and YUV, etc. Then, TFDT uses RGB space.

(a) (b) (c) (d)

Fig. 1. Comparison between Two-frame difference method and three-channel-two-frame difference method (image (a) and (b) represent two adjacent frames of images, image (c) and (d) represent the results of traditional two-frame difference method and three-channel-two-frame difference method)

$$diff_i(x, y) = | F_{n,i}(x, y) - F_{n-1,i}(x, y) | \quad i = 1, 2, 3 . \qquad \textbf{(1)}$$

Where $diff_i(x,y)$ represents the three-channel differential image and *i* represents the channel index.

$$R_{n,i}(x, y) = \begin{cases} 255 & diff_i(x, y) > T \\ 0 & diff_i(x, y) <= T \end{cases} \quad i = 1, 2, 3 . \qquad \textbf{(2)}$$

$$mask(x, y) = R_{n,1}(x, y) | R_{n,2}(x, y) | R_{n,3}(x, y) . \qquad \textbf{(3)}$$

In order to obtain the most information of the object, the threshold value T is set to 10, $R_{n,i}(x,y)$ represents the change mask for each channel, and $mask(x,y)$ is the resulting change mask.

Fig. 2 shows the effect of three-channel-two-frame difference method on different motion states. It can be found from image a and b that the object changes of these two frames are very small. But the three-channel-two-frame difference method can well extract the weak changes of the object accurately, which is good for estimation of the complete area of the object. The disadvantage of Two-frame difference method is that when the object moves too fast, it detects two objects (true object and false object), where the true objects are the object's pixels and the false objects are the background's pixels. But the purpose of the Two-frame difference method is not to segment the object but to obtain the change mask, so it's shortcomings could not affect TFDT.

(a) (b) (c) (d) (e) (f) (g) (h) (i)

Fig. 2. Effect of three-channel-two-frame difference method on different motion states (the first (image (a)(b) and (c)), second (image (d)(e) and (f)) and third (image (g)(h) and (i))scene are static, slow and fast motion respectively ;image (a)(b)(d)(e)(g) and h represent adjacent frames in every scenes, and image (c)(f) and (i) represent change mask in every scene)

3.2 Estimate Complete Object Area

As seen in Fig. 2, the objects moving slowly and quickly have obvious changes, so the object area can be easily estimated. The stationary object has little change, so only the information of the changing edge can be used to estimate the object area. In TFDT, the topology structure is used to estimate the complete region of the object.

Topology is widely used in computer network. Inspired by the application of mesh topology in computer network [19], mesh topology is used in TFDT to estimate the complete region of the object. The characteristic of mesh topology is that the irregular connections between nodes form a unified whole.

TFDT abstracts the connected domain into nodes, and then uses the clustering method to find the relationships of all the nodes in the change mask. The same object or adjacent object nodes are relatively close to each other, which can be classified into one group. Then, the mesh topological structure is generated within the same group of nodes to form a whole. TFDT adopts improved Max-Min-Distance (Improved Max-Min-Distance, MMD+) method to realize node clustering.

MMD+. TFDT uses MMD+ to find the clustering relationship between nodes. Traditional MMD (max-min-distance) method divides all nodes into at least two groups, but all nodes may be in the same group in the image, so it is not suitable for TFDT. The core idea of MMD+ is to find a new clustering center based on the maximum distance and conduct sample classification based on the minimum distance (all distances in this paper are L2 distance). Since the change mask may be have noise, MMD+ method adds two area thresholds $T_{\min}$ and $T_{\max}$, and improves the clustering tightness parameter θ to be adaptive.

MMD+ first calculated the region with the largest area in the change mask and denoted as $Area_{\max}$, and save its width and height as *W* and *H*, and then $T_{\max} = 1/3 \times Area_{\max}$. After many experiments, it is concluded that $T_{\min} = 200$ can be used to obtain better results. The calculation of cluster tightness θ is shown in formula (4).

$$\theta = \begin{cases} 1 & d <= W \\ 1 - 0.6 \times \dfrac{d-W}{H-W} & W < d <= H \\ 0.4 & d > H \end{cases}. \quad \textbf{(4)}$$

Where *d* represents the maximum distance between the initial node and other nodes, and the value 0.4 is a suitable value obtained through multiple tests. When *d* is less than *H*, the nodes tend to belong to the same group. The smaller *d* is, the stronger the trend will be. When *d* is not greater than *W*, all nodes are grouped together. Below are the steps for the MMD+ method:

(a) Define Z_1 as an initial node. Define $D_{i,1}$ as the distance from all the nodes to the starting node, and find the maximum distance as *d*;

(b) If $d <= W$, all nodes are grouped together; otherwise, the farthest node from Z_1 is regarded as the second cluster center and call it Z_2;

(c) Calculate the distance of all nodes from the cluster center Z_1 and Z_2, and record them as $D_{i,1}$ and $D_{i,2}$;

(d) $Dis = \max(\min(D_{i,1}, D_{i,2})) \quad i = 2, \cdots\cdots, n$. Calculate θ according to formula (4). If $Dis > \theta \times d$, the node is denoted as the third cluster center;

(e) If there are already *k* clustering centers, judge whether there is a new clustering center according to formula (5); if there is, continue to search for a new clustering center; otherwise, go to step f);

$$Dis = \max_i [\min[D_{i,1}, D_{i,2}, \cdots\cdots, D_{i,k}]] > \theta * d \quad i = k+1, k+2, ..., n. \quad \textbf{(5)}$$

(f) According to formula (6), calculate the distance between the remaining nodes and all clustering centers, and classify according to the minimum distance;

$$D_{i,j} = | x_i - Z_j | \quad i = k+1, k+2, \cdots\cdots, n, j = 1, 2, \cdots, k. \quad \textbf{(6)}$$

Since MMD+ uses the adaptive parameter θ, the clustering relationship between nodes can be effectively found. Then, the nodes are divided into one or more groups in the change mask.

Generate Mesh Topology. Fig. 3 shows the process of estimating the object's complete region by mesh topology and get the region's bounding box. In Fig. 3, the two people are close, so the five incomplete areas are divided into one group. Image c shows the mesh topology generated within the same group. It can be found that the topology connects the same group of nodes into a whole, and the area covered by the mesh topology also contains a large number of object's pixels. Image d shows that the area covered by the mesh topology is filled, and the incomplete area is connected together to form a unified whole. As shown in image e, in order to eliminate the objects pixel as much as possible, the whole connected by the mesh topology is taken as a bounding box. Then remove all the pixels inside the bounding box. This will remove some background pixels, but the background pixels can be obtained after the next frame or several frames.

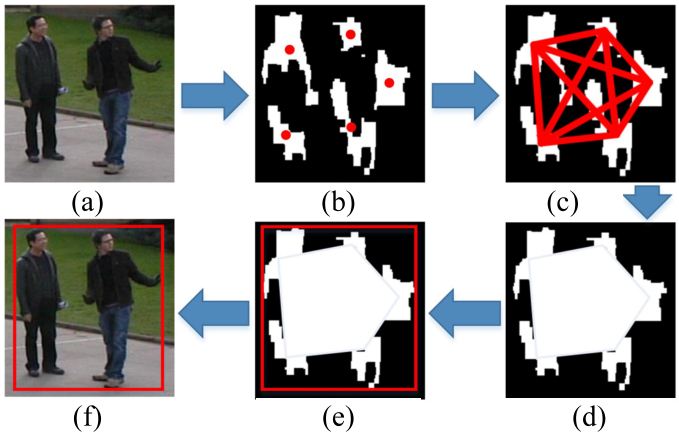


Fig. 3. Mesh topology estimates the object's complete region and get its bounding box. (image (a) represents the original picture; image (b) represents the change mask; image (c) represents the generation of mesh topology in the same group of nodes; image (d) represents that the mesh topology connects the dispersed domain into a whole; image (e) represents the bounding box of the object after connecting; image f represents the bounding box marked in the original image)

Fig. 4 shows bounding boxes in a series of images. It can be found that TFDT can accurately estimate the pixels of object and eliminate them.

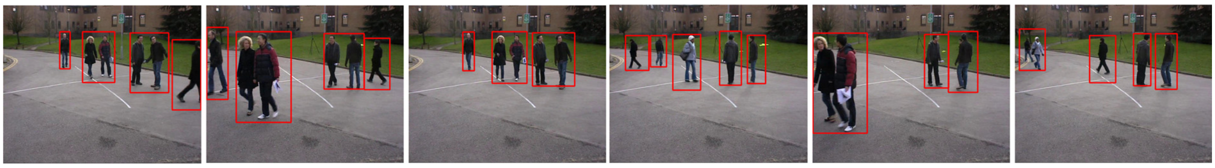


Fig. 4. Bounding boxes in a series of images

4 Improvement for Foreground Detection Algorithms

TFDT is a universal algorithm, which can improve the detection algorithms. TFDT takes video as input and output as background pixels. When TFDT is used to improve the foreground detection algorithms, only the background pixels output by TFDT are added to the background model of the foreground detection algorithms. After all background pixels have been collected by TFDT, the background model is initialized. Finally, input the video images in foreground detection algorithm to realize foreground detection. In the experiment, ViBe and Codebook are improved by TFDT. Next, the improved ViBe by TFDT and the improved Codebook by TFDT are called IViBe and ICodebook, respectively.

4.1 ICodebook

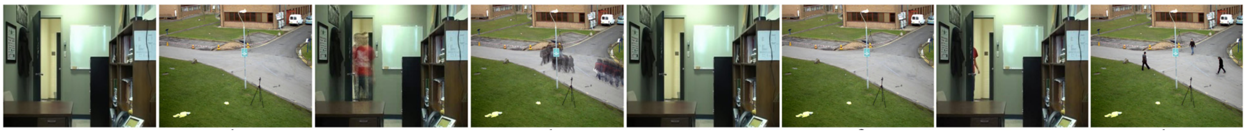
Although the Codebook can filter the pixels in the background model after initialization, it can't filter all the object's pixels when the object moves slowly or stationary during modeling. The steps to improve Codebook by TFDT are following:

(a) Input video sequence and collect background pixels by TFDT;

(b) Initialize the Codebook's background model with the background pixels;

(c) When all background pixels have been collected, the Codebook is initialized and implemented to realize foreground detection.

Through the above steps, the background model of Codebook can be built by TFDT. Image a、b、c and d of Fig. 5 shows the comparison of the initial modeling between the ICodebook and the Codebook. By comparison, it can be found that ICodebook has better effect than Codebook.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 5. Codebook, ICodebook, ViBe and IViBe reconstruction of the background (image (a) and (b) represent the initial background built by the ICodebook, and (c) and (d) represent the initial background built by the Codebook; image (e) and (f) represent the initial background built by the IViBe; (g) and (h) represent the initial background built by ViBe)

4.2 IViBe

ViBe is initialized through the first frame modeling. When the objects appear in the first frame, ViBe adds object’s pixels into the background model and detect “ghost”. The steps of using TFDT to improve ViBe are as follows:

(a) Input video sequence, use TFDT to collect background pixels and record the collection times of background pixels in each position;

(b) Calculate the average of the background pixels collected from each position to obtain the initial background;

(c) Use the background as the initial frame of ViBe, and ViBe is implemented to realize foreground detection.

Through the above steps, the background model of ViBe can be built by TFDT. Image (e)(f)(g) and h of Fig. 5 shows that the IViBe and the ViBe calculate the initial background in two scenes. To verify that TFDT can eliminate object’s pixels, there are objects in each frame of the modeling stage. Image e and f respectively need 62 and 12 frames to build the background model by TFDT. The number of modeling frames used by TFDT to improve any detection algorithms are consistent in the same scenes. From the Fig. 5, it can be found that TFDT eliminates the object’s pixels and obtains the real background image.

Through the ICodebook and IViBe, it can be found that TFDT can effectively eliminate object’s pixels and build a good background model.

5 Experimental Results and Analysis

In order to verify the effectiveness of TFDT, Codebook 、ViBe、Subsense、Pawcs 、IViBe and ICodebook are compared in CDnet database. Codebook and ViBe are the original algorithms, Subsense and Pawcs are the state-of-the-art algorithms. ICodebook and IViBe are the improved algorithms by TFDT. The information of the selected scenes are shown in Table 1. Since the test data is part of the database, the initial frame and test frame represent the number of frames in the original database. Video sequences of scenes have object’s pixels in each frame of the modeling stage. Because the comparison algorithms don’t make the experiment under the setting conditions, the experiment of the comparison algorithms in this paper are obtained by using their open source code under the setting conditions.

Table 1. The information of scenes

Scene	Initial frame	Test frame	Scene	Initial frame	Test frame
CameraParameter	1120	1160	PETS2006	40	300
CopyMachine	1450	1551	backdoor	1657	1749
cubicle	6945	7059	LightSwitch	673	695
office	585	1117	busStation	935	1099
pedestrians	607	667	sofa	2685	2734

The experiment will verify the effectiveness of TFDT from three aspects, as follows:

1. When there are objects in the background modeling stage, the TFDT can eliminate the object’s pixels and build an effective background model.

2. The improved algorithm by TFDT can effectively suppress "ghost" in the detection results.
3. When there is no object in the modeling stage, the background model built by TFDT won't affect the detection.

In order to make the experiment sufficient, the algorithms are tested in multiple scenes, the scenes include complex situations such as stations and offices. In the scenes, the object motion is changeable, including the stationary, slow and fast motion. Fig. 6 shows the background model built by each algorithm. From Fig. 6, it can be found that the background built by ICodebook and IViBe does not contain object's pixels, while the background built by other algorithms contain object's pixels. It is the most difficult to build the background in busStation scene, because the modeling stage contains multiple objects and the objects are stationary for a long time. In this scene, ICodebook and IViBe can still build a decent background model, while the background built by other algorithms contain a large number of object's pixels. Therefore, the TFDT can eliminate the object's pixels in the background modeling stage and build an effective background model.

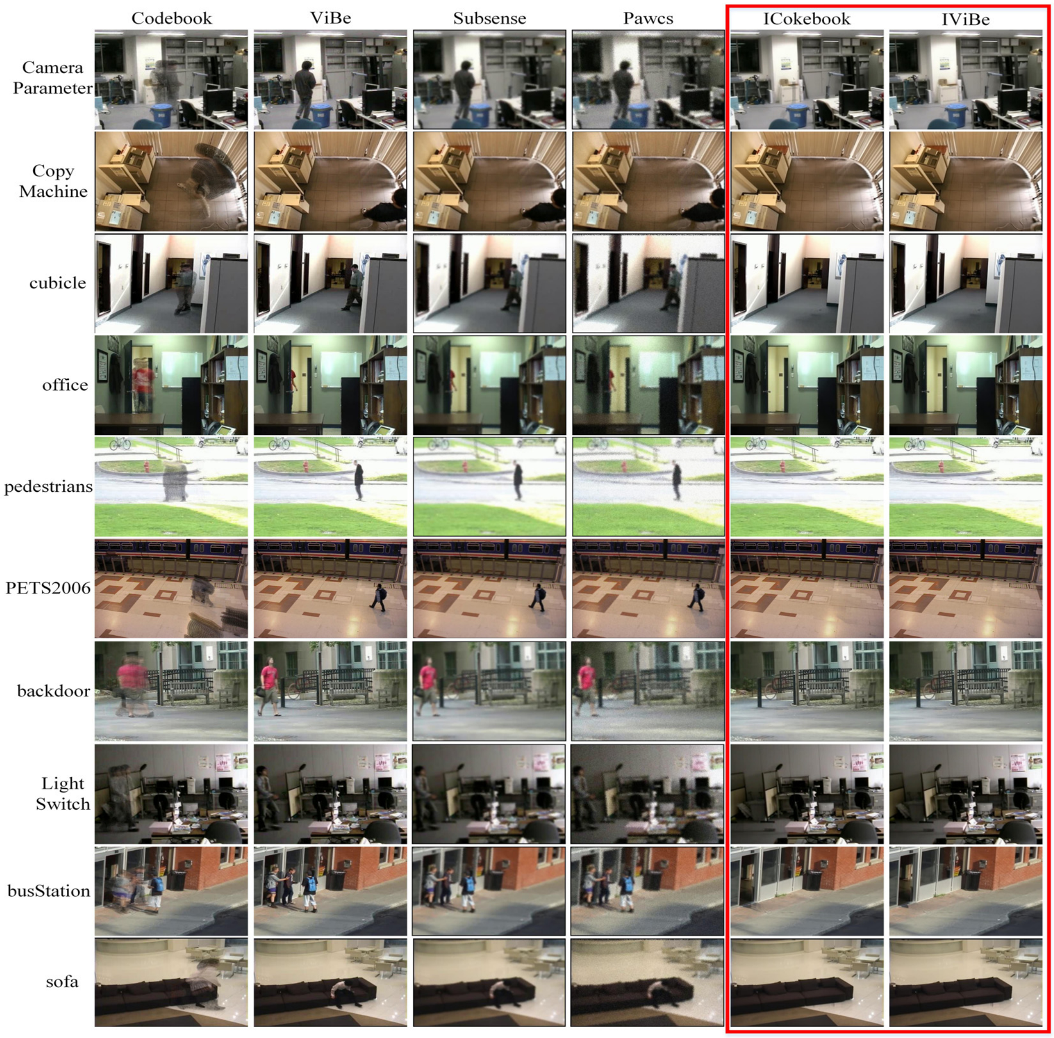


Fig. 6. The background model built by each algorithm

Fig. 7 shows the detection results of each algorithm. As can be seen from Fig. 7, the Codebook and ViBe generate large "ghosts" that cause false detection. Subsense only produces "ghost" in the scene of busStation and CameraParameter. Subsense and Pawcs remove "ghost" in a short time by quickly updating the background model. However, when the distinction between the objects and background is low, these two algorithms can update the object's pixels into the background. Therefore, Subsense and Pawcs often have non-ideal detection results, such as the result in the senses of LightSwitch and backdoor. Because the TFDT can eliminate the object's pixels in the background model, there is no "ghost" in the detection results of ICodebook and IViBe.

In this paper, the evaluation indexes in literature [20] are used to analyze the detection results of each algorithm. The indicators are accuracy (P), recall (R), false alarm rate (FPR) and missed detection rate (FNR), which are defined as follows:

$$P = \frac{T_P}{(T_P + F_P)}, R = \frac{T_P}{T_P + F_N}, FPR = \frac{F_P}{F_P + T_N}, FNR = \frac{F_N}{T_N + F_P}. \quad \textbf{(7)}$$

Table 2 shows the average results of each algorithm in all scenes. It can be seen from table 2 that the performance of the improved algorithm by TFDT has been significantly improved. Compared with Codebook, ICodebook's accuracy rate increased by 18.56% ($(P_{\text{ICodebook}} - P_{\text{Codebook}}) / P_{\text{Codebook}}$, the change of other indices is calculated in the same way), recall rate increased by 5.84%, false alarm rate decreased by 84.70%, and missed detection rate decreased by 17.69%; Compared with ViBe, IViBe's accuracy rate increased by 74.67%, recall rate increased by 2.61%, false alarm rate decreased by 95.85%, and missed detection rate decreased by 6.21%. Compared with state-of-the-art algorithms, ICodebook has better indexes than Subsense, and the recall rate and false alarm rate of ICodebook are also better than Pawcs.

Table 2. Quantitative analysis of Codebook, ViBe, Subsense, Pawcs, ICodebook and IViBe (the evaluation index data is the average result of the all scenes in Fig. 7)

Algorithm	*P*	*R*	*FNR*	*FPR*
Codebook	0.81158	0.71425	0.01933	0.01340
ViBe	0.55903	0.63672	0.02351	0.03158
Subsense	0.95324	0.73283	0.01988	0.00224
Pawcs	**0.98154**	0.63625	0.02480	**0.00102**
ICodebook	0.96224	**0.75599**	**0.01591**	0.00205
IViBe	0.97647	0.65336	0.02205	0.00131

Fig. 7. The detection results of each algorithm

In order to prove that TFDT can complete modeling without affecting the detection performance of the original algorithm, this paper compares Codebook with ICodebook, as well as ViBe with IViBe. In the experiment, there is no object in the modeling stage, which is to rule out the influence of “ghost” on the original algorithm. In this paper, four scenes are selected for the experiment. Fig. 8 shows the detection results of the four algorithms. Through comparison, it can be found that the detection of Codebook and ICodebook are basically consistent, the detection of ViBe and IViBe are also basically consistent. Table 3 is the qualitative analysis of the four algorithms, and the data is the average value of the four scenes. Through comparison, it can be found that the indices before and after the improvement are basically consistent. Therefore, when there is no target in the background modeling stage, the detection result of the improved algorithms by TFDT are as good as the original algorithms. TFDT don’t affect the detection performance of the original algorithms.

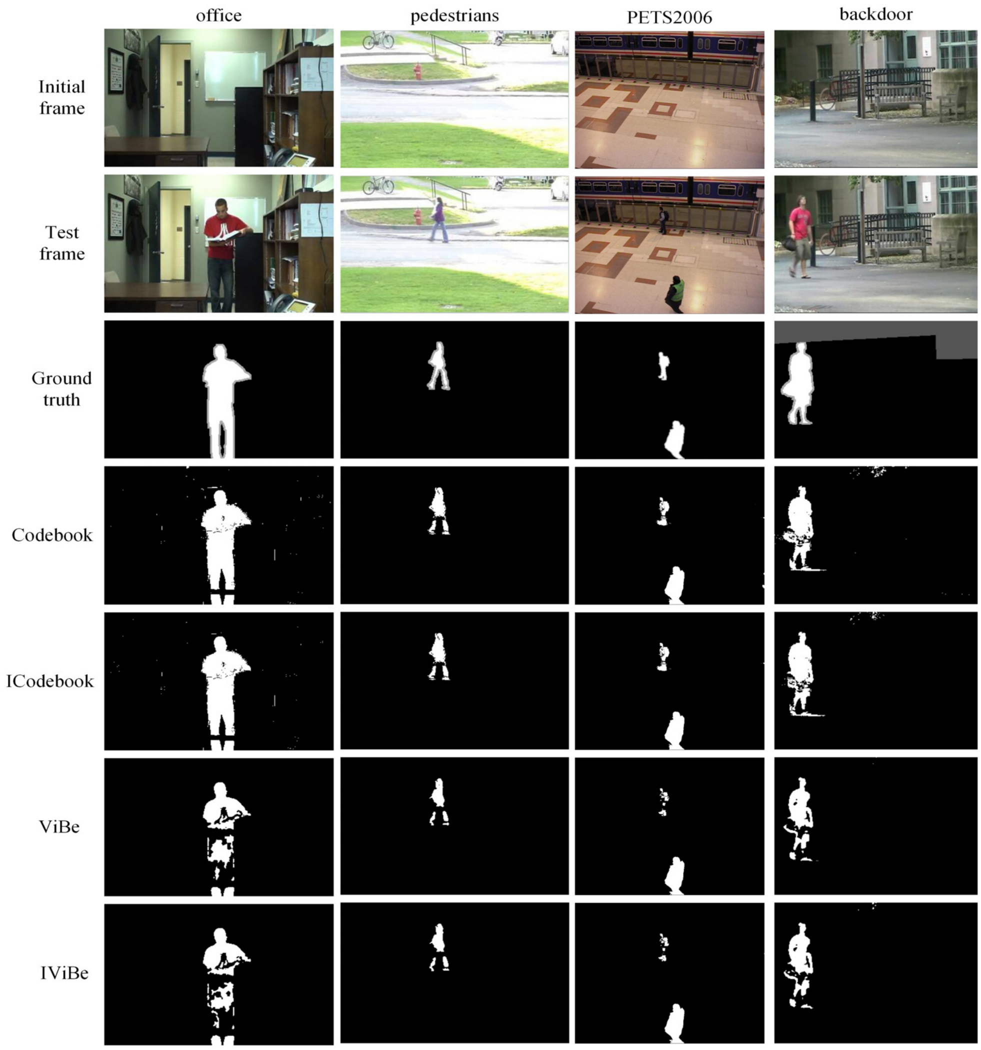


Fig. 8. The detection results of each algorithm

Table 3. Quantitative analysis of Codebook, ViBe, ICodebook and IViBe (the evaluation index data is the average result of the four scenes in Fig. 8)

Algorithm	*P*	*R*	*FNR*	*FPR*
Codebook	0. 96110	0. 80394	0. 00993	0. 00180
ICodebook	0. 96907	0. 78846	0. 01106	0. 00127
ViBe	0. 98920	0. 62056	0. 02312	0. 00022
IViBe	0. 98927	0. 61641	0. 02365	0. 00022

6 Summary and Prospect

TFDT is a universal algorithm for modeling algorithms. Firstly, the three-channel-two-frame difference method is used to detect the movement between two adjacent frames. Then the complete target region and its bounding box are estimated with topological structure. Finally, remove the pixels in the bounding boxes and add the remaining pixels into the background model. After testing in multiple scenes in CDnet database, the improved algorithm by TFDT can suppress the "ghost" caused by the object's pixels which enter the background model.

The existence of completely stationary objects at the beginning of the video is a tricky issue. It is difficult to detect when the object is completely stationary. Meanwhile, a completely stationary object can leave "ghost" in its initial position after movement. This problem usually acquires a method of fast updating the background model, but the method can easily update the slow moving or stationary objects into the background. Therefore, modeling and detection in the case where there is a completely stationary target at the beginning of the video sequence is a problem to be solved. It is a worthwhile research direction in the future.

Acknowledgements

This research was funded by the National Natural Science Foundation of China under grant 61872027 and 61573057, the project of China Railway Corporation, grant number 2017J004-H.

References

[1] X.-F. Liu, H. Zhou, Q. Han, M.-E. Zan, D. Han, Visual gait recognition research review, Miniature microcomputer system 39(8)(2018) 1685-1692.

[2] M.-E. Zan, H. Zhou, D. Han, G. Yang, G.-L. Xu, Overview of particle filter object tracking algorithm, Journal of computer engineering and applications 55(5)(2019) 8-17.

[3] S.-Q. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, in: Proc. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015.

[4] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, in: Proc. Computer Vision and Pattern Recognition, 2018.

[5] D. Yoo, S. Park, J.-Y. Lee, A.S. Paek, I.S. Kweon, Attentionnet: Aggregating weak directions for accurate object detection, in: Proc. Proceedings of the IEEE International Conference on Computer Vision, 2015.

[6] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Proc. Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[7] S.M. Roy, A. Ghosh, Foreground Segmentation Using Adaptive 3 Phase Background Model, IEEE Transactions on Intelligent Transportation Systems 66(7)(2019) 1-10.

[8] Z. Zivkovic, Improved adaptive Gaussian mixture model for background subtraction, in: Proc. ICPR(2), 2004.

[9] K. Kim, T.H. Chalidabhongse, D. Harwood, L. Davis, Real-time foreground–background segmentation using Codebook model, Real-Time Imaging 11(3)(2005) 172-185.

[10] O. Barnich, M.V. Droogenbroeck, ViBe: A Universal Background Subtraction Algorithm for Video Sequences, IEEE Transactions on Image Processing 20(6)(2011) 1709-1724.

[11] P.L. StCharles, G.A. Bilodeau, R. Bergevin, Universal Background Subtraction Using Word Consensus Models, IEEE Transactions on Image Processing 25(10)(2016) 4768-4781.

[12] P.L. StCharles, R. Bergevin, SuBSENSE: A Universal Change Detection Method with Local Adaptive Sensitivity, IEEE Transactions on Image Processing 24(1)(2014) 359-373.

[13]P. Kanungo, A. Narayan, P.K. Sahoo, S. Mishra, Neighborhood based Codebook model for moving object segmentation, in: Proc: 2017 2nd International Conference on Man and Machine Interfacing (MAMI), 2017.

[14] S.-A. Li, P.-X. Liu, G.-L. Han, Moving object detection based on Codebook algorithm and three-frame difference, Int. J. Signal Process Image Process Pattern Recognition 10(3)(2017) 23-32.

[15] C.-D. Huang, W.-G. Huang, B. Yan, Video pedestrian detection based on Codebook background modeling, Sensors and Microsystems 36(3)(2017) 144-146.

[16] J.-Q. Xu, P.-P. Jiang, H.-B. Zhu, W. Zuo, Improvement of ViBe algorithm for moving object detection, Journal of Northeastern University (natural science editartion) 36(9)(2015) 1227-1231.

[17] D. Yang, F. Dai, Improvement of ViBe algorithm for moving object detection, Chinese journal of image and graphics 23(12)(2018) 1813-1828.

[18] L. Chen, J. Shen, W. Wang, B. Ni, Video object segmentation via dense trajectories, IEEE Transactions on Multimedia 17(22)(2015) 2225-2234.

[19] H.-B. Zhu, Network topology discovery technology analysis, Network security technology and application 3(2017) 26-27.

[20] D. Zeng, X. Chen, M. Zhu, M. Goesele, A. Kuijper, Background subtraction with real-time semantic segmentation, IEEE Access 12(7)(2019) 1-16.