The Visual Computer (2023) 39:1391–1407
https://doi.org/10.1007/s00371-022-02417-1

ORIGINAL ARTICLE

Unsupervised inner-point-pairs model for unseen-scene and online moving object detection

Xinyue Zhao[1] · Guangli Wang[2] · Zaixing He[1] · Dong Liang[3] · Shuyou Zhang[2] · Jianrong Tan[1]

Accepted: 13 January 2022 / Published online: 16 February 2022

Abstract

Moving object detection in complex environments is a challenging task. Recently, many deep learning-based methods have improved the detecting accuracy. However, because of the supervised learning nature, these methods cannot deal with unseen scene, where manually generated groundtruth of training data is unavailable. Furthermore, due to the limitation of generality, their online detection performance is not satisfactory, where the scene changes a lot over time. In this paper, we propose a new unsupervised method for moving object detection, which performs well in both seen and unseen scene, offline and online detection. It can better adapt to changes in environmental lighting and has good tolerance for dynamic backgrounds in the environment. Based on the analysis of the reflections of objects in a scene, the proposed method not only processes images at the pixel level but also takes into account the characteristics at the object level. We distinguish between the ordinary point areas and complex point areas in an image and process them separately to improve the detection accuracy. The proposed method is tested using several challenging datasets, and the results show its effectiveness and applicability in complex situations.

Keywords Moving object detection · Inner-point-pairs · Dynamic background model · Unseen-scene and online detection

1 Introduction

Detecting moving objects is an important task in video surveillance. Its goal is to extract the moving objects that pass through a scene. It has important applications in many fields, such as the intelligent monitoring, identification and tracking of intelligent robots [1–5], and so on. Over the decades, although much research has been done in this field, there are still many factors that affect the accuracy of object detection, such as strong illumination changes and dynamic backgrounds. To achieve high moving object detection accuracy, most models process a video image at the pixel level. This is helpful to improving the detection accuracy, but it is easy to ignore the connection and correlation between similar areas in the image, and it may omit some important visual information in the image.

At present, there are three main methods for moving object detection.

(1) Optical flow [6–8]: This detects a moving object by establishing the optical flow field of an image and observing the motion vector of the related pixels. This method is sensitive and can be applied to both static cameras and dynamic cameras. However, the calculation is complicated and it is easy for detection errors to occur due to the variability of the scene, such as lighting changes, background noise, etc.

(2) Frame difference [9, 10]: This compares the information of the pixels among adjacent frames. When there is an object passing through the frames, the differences between the frames will exceed the threshold. The principle of this method is simple and its detection speed is fast. However, there are obvious problems, such as the "double shadow" problem when the object moves too fast, and the "hole" problem caused by the overlap of objects between different frames.

(3) Background subtraction

In recent years, background subtraction has become the most effective and the most studied method. Some surveys [5, 17, 18, 46] have made detailed review of background

✉ Zaixing He
zaixinghe@zju.edu.cn

[1] The State Key Lab of Fluid Power & Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

[2] The State Key Lab of CAD & CG, Zhejiang University, Hangzhou 310027, Zhejiang, China

[3] College of Computer Science & Technology, Nanjing University of Aeronautics & Astronautics, Nanjing 210016, China

subtraction from different perspectives. First, a background model is established through training, and the current frame is compared to the background model or background frame to obtain foreground information. The background model is not strictly a background model; it is an algorithm process or network framework. It can be divided into three categories: (1) Supervised background subtraction based on deep learning [11–26]. These methods are based on the combination of a neural network and other detection methods to improve the detection accuracy in some scenarios. Recently, deep learning-based approaches have achieved good accuracy in several image scenarios and more methodologies are being proposed. However, as a supervised method, a substantial amount of groundtruths is necessary when training, which generally results in high costs if the data are even available. In addition, it depends heavily on the foreground objects in the training set and when applied to unseen scenes, the performance of these methods will drop a lot, which has a negative influence on practical applications. Therefore, although it achieves good offline detection accuracy, the online detection capability is poor. Some supervised methods in recent years [24–26] have shown that they can use a large number of data sets to train models and use it in unseen scene; however, they do not provide models with trained weights. The surveys [17, 18] gave a comprehensive review of background subtraction based on deep learning. For more details, we can refer to these two valuable surveys. (2) Semi-supervised background subtraction. In recent years, semi-supervised background subtraction methods based on graph signal processing have been proposed [27–29]. They obtained the instances of the videos using semantic and instance segmentation methods and used a temporal median filter as the background initialization. Then, the optical flow, intensity, and texture features of each instance was obtained, and as a node, these representations generated graph construction. Finally, a semi-supervised learning algorithm was used to train the parameters. Compared to supervised methods, it avoids hungry data learning required in deep learning and achieved good accuracy. The method is new developments in the field of computer vision and end-to-end architectures for video analysis with semi-supervised learning. (3) Unsupervised background subtraction [30–47]. First, a background model is established through training, and the current frame is compared to the background model or background frame to obtain foreground information. After a stable and anti-interference background frame is estimated, foreground information can be extracted accurately and quickly. There is no need for supervised learning, which makes background subtraction the most practical and worthy method for studying. It achieves a better effect for illumination changes and dynamic background but it still has much room for improvement.

Based on the analysis of the pixel level and object level, this paper proposes an inner-point-pairs model (IPP) based on the point pair reference for the same object. Based on the brightness characteristics of the objects in different regions of an image, the image pixels are divided into several different classes. In each pixel class, a series of reference points are established for each pixel. By building reference point models, the model determines whether the pixels belong to its original object class or belong to the foreground object. Regarding the dynamic background pixels, we can determine what object is moving or they are moving among which object classes to model the background.

After inner-point-pairs are established, the proposed model does not need to be updated in real time, and the background model will not be affected by slow or strong illumination changes. In addition, the proposed method provides a good solution to deal with dynamic backgrounds. We evaluated the proposed method on a number of datasets and compare it with most state-of-the-art methods. The experimental results show that the proposed method achieves great performance.

2 Related works

2.1 Unseen-scene detection and online detection

2.1.1 Unseen-scene detection

Unseen-scene detection refers to applying a method to a new scene whose background and foreground targets are unseen in advance and that has no available groundtruth [19]. According to the training and learning mechanisms, we can divide them into supervised methods and unsupervised methods. The difference between them is whether or not groundtruth training is required during the implementation of the method. Supervised methods need to determine the groundtruth first. In the training of convolutional neural networks, a large number of image frames and groundtruths containing foregrounds are required as the input and the quality of the training results is greatly related to the quantity and quality of the training frames. Unsupervised methods focus on judging the image to be detected according to the pixel statistics. It does not require time-consuming manual groundtruth production; therefore, its applicability is stronger and it is much more convenient.

Since groundtruth and foreground training are not required, unsupervised methods can be applied to unseen-scene detection. It can use the initial video frames to initialize the model or conduct background training, and then the detection can be executed. Supervised methods are difficult to be applied to unseen-scene detection. Due to the lack of training data with corresponding groundtruths, its detection mechanism is no longer valid.

2.1.2 Online detection

According to the functions in practical applications, detection can be divided into offline and online detection; and the methods can also be divided into two types: those only for offline detection and those able to conduct online detection. Online detection refers to using only the data that occurred in the previous period for training, and then detecting what happened in the subsequent period online. Offline detection refers to using full-time (interval extraction) data for training to detect what has happened in the time period. Whether a method can be used for online detection is essentially affected by its generalization ability: whether it can detect moving objects that have not been learned or trained.

An unsupervised method does not learn a specific groundtruth, and so it is generally applicable to online detection as long as the detection speed is appropriate. A supervised method relies on the groundtruth for learning. These methods have difficulties detecting objects that are quite different from the trained foreground in the groundtruth. Therefore, it is difficult to apply supervised methods in online detection.

Unseen-scene detection and online detection are both important for practical applications. Therefore, unsupervised methods are advantageous in practical applications. In this paper, we focus on researching unsupervised and online detection.

2.2 Background subtraction methods

Among the related background subtraction algorithms, the Gaussian mixture model (GMM) proposed by Tom SF Haines [32] was proposed early on and is widely used. Based on the brightness information of each pixel in the training datasets, the GMM and its improvements [33, 34] build a Gaussian Mixture Model and determine the foreground pixels according to whether they satisfy the trained probability models. The model needs to be constantly updated to adapt to changes in the ambient lighting and it is difficult to choose an appropriate learning rate. To solve the shortcomings of the GMM, the kernel density estimation (KDE) [35, 47] for object detection is proposed. It determines the parameters based on some training frames to achieve nonparametric estimation, making it better for complex backgrounds. Shengqin Jiang and Xiaobo Lu et al. [37] proposed a method of assigning different frame weights and replacing inefficient frames to update the background model to adapt to lighting changes and complex backgrounds, making it more effective at representing dynamic background.

The above methods are based on pixel-based models, and there are also some background modeling methods based on spatial references. M. Van Droogenbroeck et al. [38] proposed a method based on spatial sample modeling called visual background extraction (ViBe). The ViBe and its improvements [38–40] use a single frame of a video sequence to initialize the background model; and they assume that for a pixel, its neighboring points have similar characteristics. According to the spatial distribution characteristics, a sample set is created for each pixel. The background model updates via a different random rule and the ViBe is a nonparametric model. When detecting objects, it examines whether the target pixels are close to the sample value in the sample set to distinguish the foreground from the background.

In previous work, the grayscale arranging pairs model (GAP) and its improvements were proposed [41–43]. They find reference points for each pixel in an entire image based on the statistics of the image training set. These methods based on an entire reference image can well solve the problems caused by lighting changes, and do not require real-time parameter updating. Therefore, they are faster than the methods that need to constantly update the background model. The Cooccurrence Probability-based Pixel Pairs model (CP3) [42] considers some situations in complex environments, such as the movement of objects in a scene. In the Co-occurrence Pixel-Block Pairs model (CPB) [43], the authors introduced a mechanism named Hypothesis on Degradation Modification extended from CP3 to adapt the background changes and reinforce robustness to resist the “noise” in real applications and achieved better results. However, the spatial principle established by the reference points in a scene still does not have an optimal solution. Although these methods have used the statistical relationships between pixels and proposed that similar pixels can be used as a reference, they did not explain the reasons for the effectiveness of the method, and they lacked reliability in the search and establishment of similar reference points. Therefore, they can be further improved. In this paper, the underlying theoretical principles for adopting similar reference pixels are given, and a new method based on reference pixels is proposed. Particularly, some complex cases can be effectively solved according to the proposed theory.

The main contributions of this paper are as follows.

1. The underlying theoretical principle for inner-point-pairs is given, and a new method based on the inner-point-pairs model is proposed.
   The reflective characteristics of the same object or material are relatively similar, and the intensity difference of the pixels in the same object remain stable under an illumination change. Utilizing this principle, we can divide the image into different areas according to the different types of objects. With this theory, some complex

detection problems such as dynamic backgrounds can be solved. Furthermore, it provides a new idea for methods based on image object segmentation.

2. Specific algorithms are proposed to implement the proposed method.

In the algorithm, we establish reference point models for each pixel, and apply different modeling strategies to pixels with different properties. First, it classifies different pixels based on the reflective properties of the objects in the image. Then, it establishes reference points based on the statistical characteristics of the pixels in each class. For dynamic background points, it uses a mixed Gaussian model to analyze the membership classes. Finally, a small part of random cluttered background points is dealt with according to the correlation coefficient of the pixels.

The method is not affected by changes in the light intensity, and there is no need to update the parameters in real time. Experiments verify that it has higher accuracy. Especially in complex environments, it is robust and achieves high accuracy.

3 Proposed approach

Moving object detection from videos taken in complex environments is a challenging task. Dramatic lighting changes and dynamic backgrounds are the two most difficult challenges. By analyzing the reflective properties of the object materials, this method proposes a new solution to these problems.

3.1 Reflective properties of objects

In a period, if the ambient light is constantly changing, objects made of different materials often have different degrees of sensitivity to the changing light intensity. For example, the reflection of an object made of cloth is often insensitive to light changes. Even under severe light changes, its intensity will not change greatly. Meanwhile, the reflection of an object made of wood or stone will change greatly when the ambient light changes. Furthermore, although a single pixel has a large intensity change when the light intensity changes drastically, the differences among pixels showing the same object material is relatively small because the reflective properties of the same material are relatively consistent. This means that when the light intensity increases or decreases, the pixels showing the same object material have the same degree of increase and decrease in their intensity.

In the description of this paper, for the sake of simplicity, it is assumed that the same object is composed of the same material. If an object is composed of two or more materials with different reflective properties, it can be treated as multiple objects. If the composition and material characteristics of different objects are the same, they can also be treated as a single object. Figure 1 shows some pixels' intensity over a period of time. P, Q, and W are three points on the cloth, wood, and distant view, respectively, in the same scene. Their intensities have obvious differences when the lighting changes. The three points P_1, P_2, and P_3, which belong to the same type of object, have the same changing characteristics with each other, even if their own intensity fluctuates greatly when the ambient light changes.

From the intensity value curves of the pixel points in different parts of the image, it can be seen that the shapes of the pixel intensity curves of the same objects are similar, whereas those from different objects are quite different. At the statistical level, the intensity vectors from the same object over a period of time have a strong correlation, whereas the correlation between the intensity vectors from different objects is poor. According to this property, we can segment an image according to the category of the object. There are strong similarities between the pixels form the same object, which can be used as a basis for cross-references.

3.2 Ideal model

3.2.1 Cross-reference principles

According to the previous analysis, we can divide the pixels of an image into different classes. Each class represents a background object class. Pixels from the same object have a strong correlation. Therefore, for each pixel, we can select a set of pixels within the same object as reference points. In general, the intensity relationship between a pixel and its reference point remains stable. When a foreground object occludes the pixel, the intensity of the pixel changes greatly. However, different from the changes due to the influence of other background factors such as lighting, the foreground object only causes the intensity of the area near the pixel to change, and it has little effect on other pixels of the object. That is, most of the reference points of the target pixel will not be affected. Therefore, the appearance of the foreground object destroys the stable difference between the pixel and its reference pixels. Based on this property, the area of the foreground object can be detected. To make the detection more accurate and robust, multiple reference points can be selected for each pixel to conduct a comprehensive evaluation.

3.2.2 Dynamic background case

In some complex environments, in addition to the constantly changing ambient lighting, objects in the background can also change, such as the constantly shaking leaves in a scene, a rotating fan in the background, and large-scale background changes caused by camera shake, etc. The objects in the back-

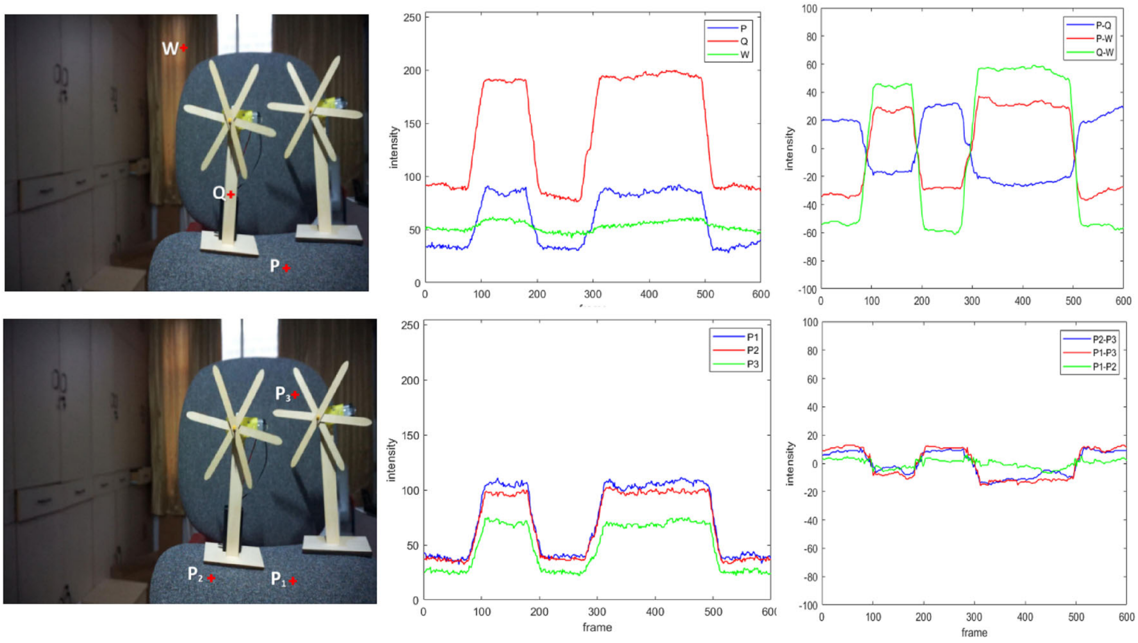


Fig. 1 The intensity values of different points in a image and their differences. (The ambient light is constantly changing.) The first row of pictures shows the intensity values of selected pixels showing different objects in the image over a period of time while the ambient light changes. From left to right: P (cloth), Q (wood), and W (distant view). The second row shows the intensity value of three different points on the same object (cloth) over a period of time. From left to right: P_1, P_2, and P_3. The third column is the differences between them

ground are moving. The image shows that the pixels in these areas are switching between different objects, as is shown in Fig. 2.

The intensities of the pixels under the influence of these complex factors often have great changes and are different from the statistical characteristics of the existing objects in the scene. However, even if these pixels are constantly changing, they always change back and forth between the objects existing in the scene. For example, the edges of leaves always change between the leaves and the background, the rotating fan always switches between the fan blade and the background, and the changes at the edges caused by camera shake are also switched back and forth between the objects on both sides. Therefore, in the dynamic background area, we should determine which kinds of objects it switches between, and set two or more sets of reference points accordingly.

One of the most critical problems that this model needs to solve when it is implemented is how to accurately segment an image into different object regions. Because the actual scene is generally unseen and complex, the current segmentation algorithms cannot achieve the ideal segmentation effect at all. However, based on the reflective property of objects analyzed above, we can use clustering instead of segmentation. We analyze the training set statistically, cluster the pixels with the same intensity characteristics in the training image frames into the same class, and treat them as the same object. The implementation of image pixel clustering is much more practical than object segmentation, and so we segment images based on a clustering algorithm. In addition, there are some important issues, such as how to distinguish between static background points and dynamic background points, how to determine the categories of the dynamic background points, etc. We implement the approach with practical algorithms in the next section.

4 Algorithms

4.1 Pixels classification

Figure 3 shows our system flow diagram. Detailed descriptions of each module are given in the following sections.

Suppose that the number of images in our training set is T, which represents the number of frames of this training video. The training set can be represented as

$$S = \{I_1, I_2, I_3, I_4 \ldots I_T\}, \tag{1}$$

where I_1, $I_2 \ldots I_T$ are the images in the sequence. The number of pixels in each image is $W \times H$. Therefore, the training set can be seen as a three-dimensional space

$$S = \{\ (u, v, t)\ \mid 0 \le u \le W,\ 0 \le v \le H,\ 0 \le t \le T\,\}. \tag{2}$$

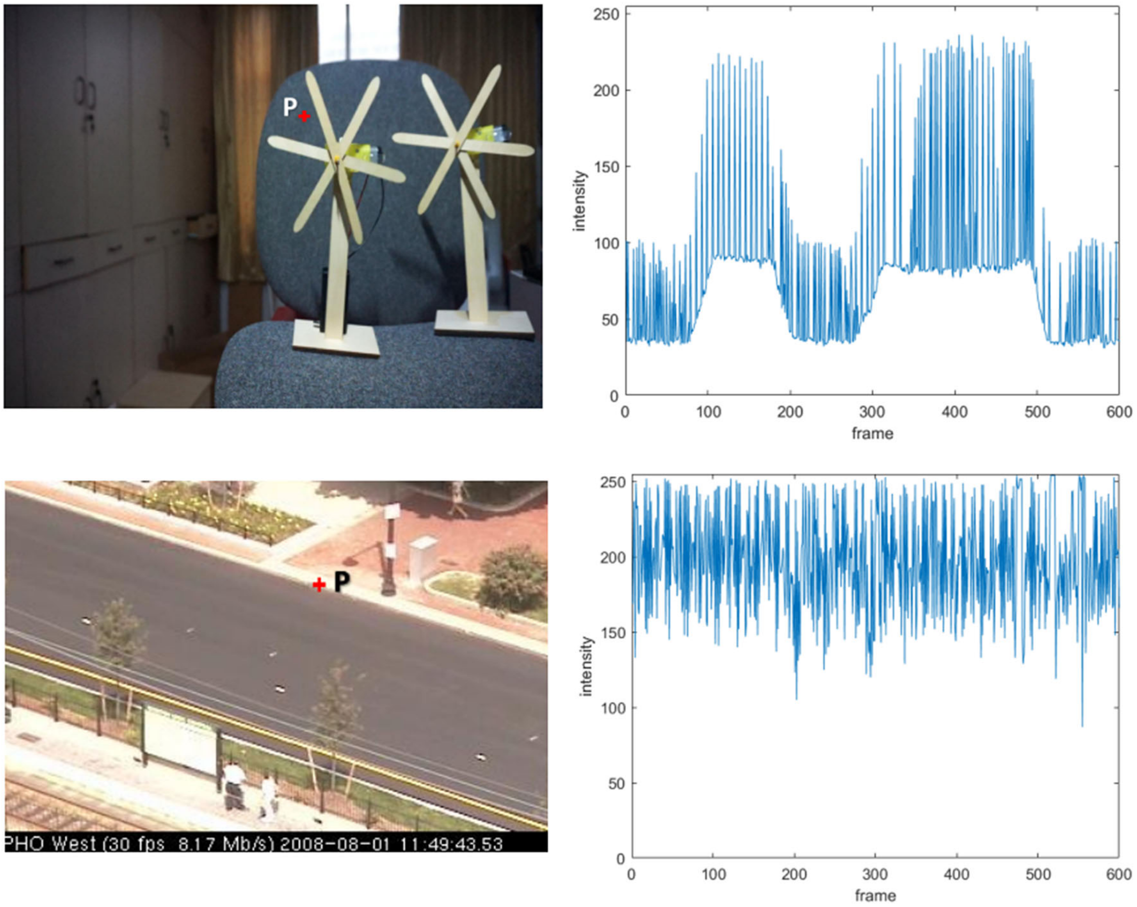
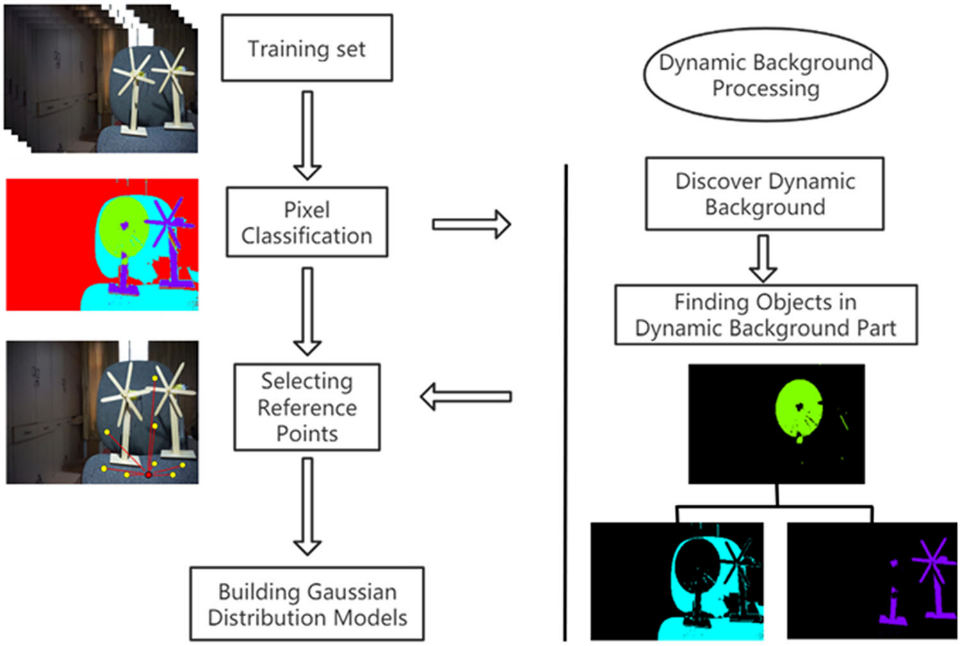


Fig. 2 Point *P* in the first row of pictures is a point in the area of the fan blade that is constantly rotating, and the right picture shows the change of its gray value with time. Point *P* in the second row of pictures is a point on the picture taken by a camera. The right picture shows the change of its gray value with time

Fig. 3 System flow diagram for modeling


Training set
Pixel Classification
Selecting Reference Points
Building Gaussian Distribution Models
Dynamic Background Processing
Discover Dynamic Background
Finding Objects in Dynamic Background Part

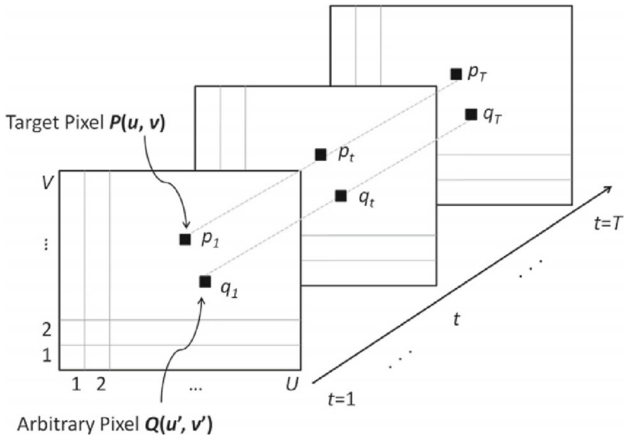


Fig. 4 Point P and Q in the training image sequence

Each element in the three-dimensional space is a pixel's intensity, whose value range is [0, 255].

Suppose P is a pixel located on (u, v) in one video frame t, which is denoted as $P_t(u, v)$. We arrange the intensity value of (u, v) in each frame so that it can be denoted as a vector

$$P(u, v) = [p_1, p_2, p_3, p_4 \dots p_{\mathrm{t}}] \tag{3}$$

The vector $P(u, v)$ is a basic parameter for a pixel. Just as Fig. 4 shows, the target pixel P and an arbitrary pixel Q located at (u', v') in the image can be expressed as follows:

$$\begin{aligned} &P(\mathrm{u}, \mathrm{v}) = [p_1, p_2, p_3, p_4 \dots p_t] \\ &Q(\mathrm{u}', \mathrm{v}') = [q_1, q_2, q_3, q_4 \dots q_t] \end{aligned} \tag{4}$$

As is shown in Fig. 1, the shapes of pixels' intensity curves from the same objects are similar whereas those from different objects are quite different. Therefore, we can classify the pixels in an image using a clustering algorithm (Fig. 5). There are many clustering algorithms that can be chosen. Considering the effectiveness and efficiency, we chose K-means clustering algorithm.

The K-means clustering algorithm can cluster the points into K clusters according to the value set of K in advance. For clustered objects, the more different they are between different clusters and the more similar they are in the same cluster, the better the clustering effect.

4.1.1 Preprocessing

To eliminate the interference of the foreground in the training frame and make clustering more effective, some preprocessing is needed.

First, the mean normalization process is performed on the vector of each pixel. This reduces the number of clusters to reduce the complexity of the operation.

$$\begin{aligned} &\overline{p} = \frac{1}{T}(I_1 + I_2 + I_3 + \dots + I_T) \\ &P'(x, y) = P - \overline{p} \cdot E \end{aligned} \tag{5}$$

$\overline{p}$ is the mean of the elements of vector P, E is a matrix with the same dimension as P, and the elements are all *1 s*.

The second is filtering. Since the training images we use are real scenes, occasionally there are foreground pixels in the image when dynamic objects pass through. For example, in the 100th frame, when a dynamic target passes through point $P\ (x, y)$, the point becomes a foreground pixel, and the intensity value changes from the original p_t to p_t'. Sometimes it will have a great impact on the background model. Therefore, filtering is necessary. In the proposed method, the median filter is used, which can effectively eliminate the noise points generated by the foreground without affecting the normal points. After the preprocessing steps are completed, K-means clustering is performed on the vectors Q (u, v), where Q represents the vector of pixels in the image, as shown in Fig. 4.

4.1.2 Determination of the value of *K* in clustering

The determination of the value of K has a great impact on the accuracy of the method. If the number of clusters is large, an image will be divided into many small blocks, and the reference points will be close together, which will cause the problem that the foreground objects cover the reference area. If the number of clusters is small, the properties of some points in the same class will not meet the requirements of the reference points because of the differences in the statistical properties of the points.

One can use the commonly named "elbow method" to determine the value of K value. Its core indicator is sum of the squared errors (SSE).

$$SSE = \sum_{\mathrm{i}=1}^{k} \sum_{p \in Ci} \left|p - \overline{p}_i\right|^2$$

C_i is the i-th cluster, p is the sample point in C_i, and $\overline{p_i}$ is the cluster center of C_i (the mean of all samples in C_i). SSE represents the clustering error of all samples. As the number of clusters k increases, the sample division will be more refined, the degree of aggregation of each cluster will gradually increase, and the SSE will naturally gradually decrease. Usually, a set of data has an optimal clustering value. When k is smaller than the optimal clustering value, increasing k will greatly increase the degree of aggregation of each cluster, and so the SSE will decrease greatly. When k exceeds

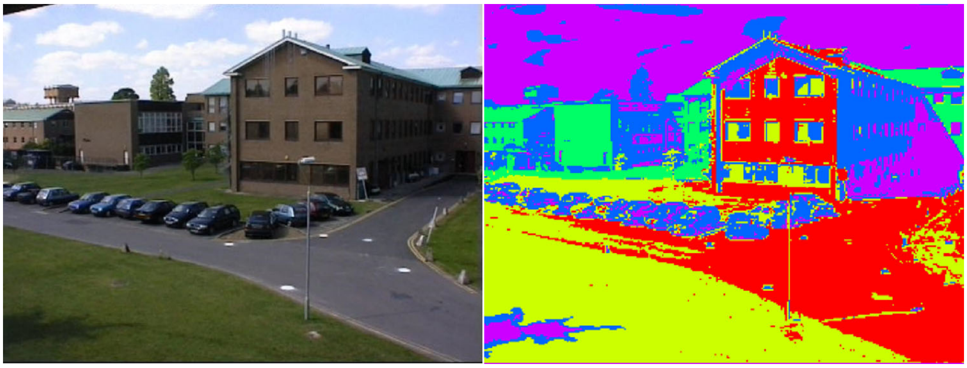


Fig. 5 Image before and after clustering. Each color represents a cluster

the optimal clustering value, the aggregation will decrease rapidly, and so the SSE will decrease slowly. The curve of the sum of the squared errors to the number of clusters looks similar to an elbow, and so it is called the "elbow method". The value of K at the inflection point of the curve is usually the optimal number of clusters.

In an image, the number of objects is usually determined, and so the number of pixel classes is also determined. After determining the optimal number of clusters K, the optimal number of image segmentations is also determined.

4.2 Static background model

In an image, most pixels are stable. We called these pixels the static background part. In the static background part, there are no frequent foreground or background changes. After clustering, the part can be ideally divided into several clusters where the points in the same cluster have similar statistical properties. For each target point, we evenly select the reference point in the area of the class that the target point belongs to. This means that the reference points are distributed as dispersedly as possible. It not only ensures that the reference points have similar properties to the target point, but also largely avoids the false judgment due to foreground objects covering target point and multiple reference points at the same time.

Suppose that we need to select n reference points for each pixel. For the convenience of the expression, we represent the target pixel as P. The pixel's intensity vector is

$$P = [p_1, p_2, p_3, p_4 ... p_t]$$

Its n reference points are Q_1, $Q_2 ...$ Q_n, which are represented by n vectors:

$$\begin{aligned} &Q_1 = [q_1^1, q_2^1, q_3^1, q_4^1 ... q_t^1] \\ &Q_2 = [q_1^2, q_2^2, q_3^2, q_4^2 ... q_t^2] \\ &...... \\ &Q_n = [q_1^n, q_2^n, q_3^n, q_4^n ... q_t^n] \end{aligned} \tag{6}$$

The intensity vector curves of Q and P have similar shapes and the correlation coefficient between vectors is close to *1*. In the ideal case, the intensity of each reference pixel differs from the target pixel by a constant C.

$$\begin{aligned} &C^1 = p_1^1 - q_1^1 = p_2^1 - q_2^1 = ... = p_t^1 - q_t^1 \\ &C^2 = p_1^2 - q_1^2 = p_2^2 - q_2^2 = ... = p_t^2 - q_t^2 \\ &...... \\ &C^n = p_1^n - q_1^n = p_2^n - q_2^n = ... = p_t^n - q_t^n \end{aligned} \tag{7}$$

In other words,

$$\begin{aligned} &C^1 = C_1^1 = C_2^1 = ... = C_t^1 \\ &C^2 = C_1^2 = C_2^2 = ... = C_t^2 \\ &...... \\ &C^n = C_1^n = C_2^n = ... = C_t^n \end{aligned} \tag{8}$$

However, in the actual situation, C^1, C^2, C^n, etc. are not exactly the same, but rather they float around a certain value. In other words, the values of C_1^1, C_2^1 ... C_t^1 are not exactly the same as C_1^1, but they are near C_1^1. Therefore, we can build a Gaussian distribution model for it.

$$\begin{aligned} &C_x^1 \sim N(\overline{C^1}, {\sigma^{1}}^{2}) \\ &C_x^2 \sim N(\overline{C^2}, {\sigma^{2}}^{2}) \\ &...... \\ &C_x^n \sim N(\overline{C^n}, {\sigma^{n}}^{2}) \end{aligned} \tag{9}$$

where C_x^n is the difference between the target point and the n' th reference point at any time x. $\overline{C^n}$ is the mean value of the difference between the target point and the n' th reference point over a period of time. σ^{n^2} is the statistical square difference of the difference between the target point and the n' th reference point over a period of time. Their calculation formulas are as follows.

$$\overline{C^n} = \frac{1}{t}\sum_{x=1}^{t}(p_x - q_x^n) \tag{10}$$

$$\sigma^{n\ 2} = \frac{1}{t}\sum_{x=1}^{t}(p_x - q_x^n - \overline{C^n})^2 \tag{11}$$

Therefore, for each reference point, there is an accompanying Gaussian distribution model composed of the difference C_x.

4.3 Dynamic background model

4.3.1 Recognizing dynamic pixels

As described in the first part, not all backgrounds are static or ideal in actual images. Some areas will be disturbed by various complex environmental factors, such as the dynamic background in the scene. We call these dynamic pixels.

Usually, after each image is clustered, a cluster center is generated for each cluster. The cluster center of a certain cluster represents the central index of this cluster. In a cluster with high stability, the distance between the vector of each point in the class and the cluster center is relatively close, such as the walls and roads of the same material in Fig. 5.

In this paper, we use the average Euclidean distance as a measurement, and the calculation formula is as follows.

$$d = \frac{1}{T}\sqrt{\sum_{i=1}^{T}(p_i - q_i)^2} \tag{12}$$

where p and q are the elements in vectors P and Q, respectively, and T is the dimension of the vector.

Figure 6 shows the two instances of the distance statistics between the pixels of some classes and the cluster center in some cluster. For some dynamic background points in the image, their intensities in the video are constantly changing and cannot be strictly classified into a certain cluster.

When the K-means is used to segment object classes, the dynamic pixels will be difficult to cluster into any major class. The K-means will classify them into the closest major class, but they will deviate farther from the clustering center, as shown in Fig. 6. We can filter out dynamic areas utilizing this property. In order to determine the threshold for filtering dynamic background points, the distances between the values of pixels and that of their respective cluster center are counted. The mean m and standard deviation σ of these distances are calculated. In our method, those points that exceed the mean value of 2.5σ are counted as dynamic background points.

Based on the analysis above, when the scene contains a dynamic background, the image pixels can be divided into two parts. One is the static pixels with relatively stable characteristics; and the other is the dynamic pixels, which usually switch between two or more static clusters. When the clustering method is used to segment object classes, dynamic pixels will be difficult to cluster into any large cluster or they will be classified into some cluster but will be far away from the cluster center. Therefore, we can filter the area of dynamic pixels based on statistical information.

4.3.2 Modeling dynamic pixels

Another key issue is how to find the two or more types of objects to which the dynamic background points belong. We deal with them separately from two aspects according to the actual situation. In general, the dynamic pixels' variation in the scene is usually regular or frequent. This is a common situation in real scenarios.

Figure (a) is a frame in a fan blade rotation video, Figure (b) is the intensity value change of point P in the figure, Figure (c) is the envelope extracted from Figure (b), and Figure (d) is the matched cluster center.

Figure 7 shows the statistical intensity of the pixels of the fan blades during the period of time when the background fan is rotating, which can ideally represent some situations of dynamic pixels. In the picture, the fan is spinning constantly. The rotating fan blade part in the image can be regarded as a dynamic background part. This part shifts between the fan blades and the background behind the fan. Therefore, we can view this part of the dynamic background as two objects: the fan blades and the background behind the fan. Therefore, we should use the properties of these two types of objects to represent the dynamic background.

It can be seen from the figure that due to the frequent rotation of the fan blades, the statistical characteristics of the pixels are shifting between the two states. Therefore, two combined curves can be found by solving the upper and lower envelope solutions. That is, we find the intensity characteristics of the pixel when it belongs to the fan blade and when it belongs to the background behind the fan. These two gray properties together represent the property of this dynamic pixel.

By correlating these two curves with the clustering centers of each cluster in turn, the two closest clusters can be matched. Similarly, the dynamic background points, such as the frequently shaking leaves in the scene, the dynamic edges caused by frequent camera shakes, etc. can be processed in the same way. The formula used to correlate cluster centers is as follows:

$$r = \frac{\sum_{i=1}^{n}(q_i - \overline{q})(c_i - \overline{c})}{\sqrt{\sum_{i=1}^{n}(q_i - \overline{q})^2}\sqrt{\sum_{i=1}^{n}(c_i - \overline{c})^2}} = \frac{Cov(C, Q)}{\sigma_C\sigma_Q} \tag{13}$$

where Q is the envelope curve vector and C is clustering center of each cluster.When dynamic pixels can be expressed by two static clusters, we can treat the dynamic pixels as

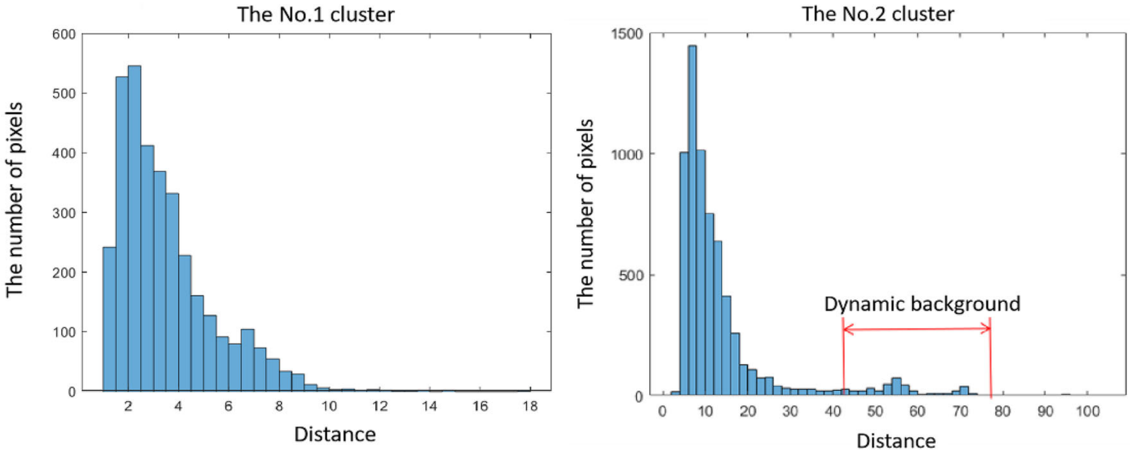


Fig. 6 Distance statistics between the pixels of some classes and the cluster center in two of clusters

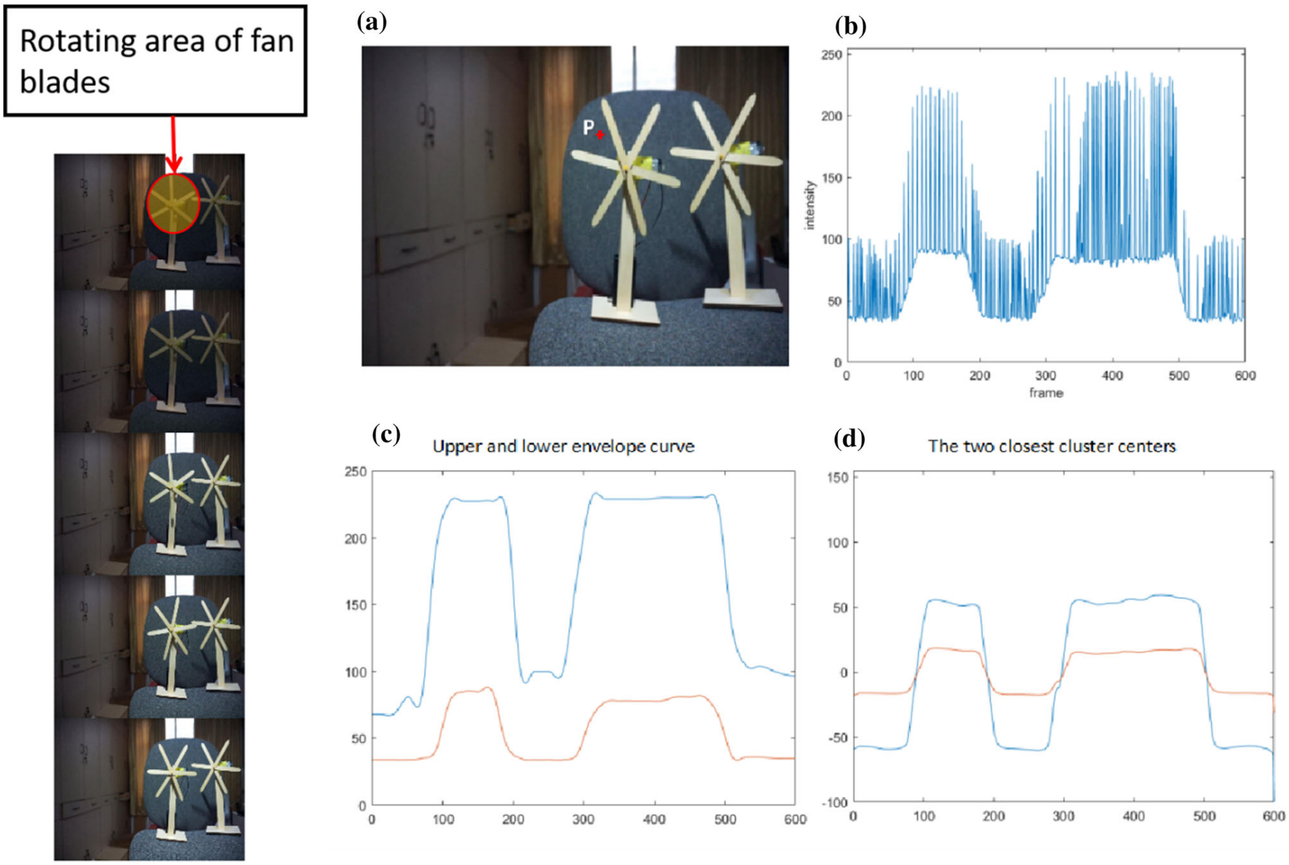


Fig. 7 Processing of dynamic background points

points that satisfy the statistical characteristics of two types of objects. Then, we select the reference points in the two types of objects and model them as static background pixels.

In a few cases, the dynamic pixels in the scene are irregular and do not change frequently. We call them messy pixels. Usually, we cannot accurately find the static clusters to represent the dynamic pixels by calculating the envelope. We can pick the pixels that are most similar to the target pixel in the entire image as its reference points. In other words, we calculate the correlation coefficient between each point in the image and the target pixel. Then, we find the pixels with the largest correlation coefficient.

In our algorithm for dealing with dynamic backgrounds, we process the first case: the dynamic pixels' variation is usually regular or frequent. Then, we calculate the envelopes for the dynamic pixels and find the cluster centers that match. When the envelope cannot be found or the envelope is quite different from the cluster center that matches, we treat them as the second case.

One possible result after such processing is that there may be a small number of dynamic pixels for which we cannot find suitable reference points, leading to the result of unreliable detection and some noise points appearing on the image. However, such points are usually isolated that can be dealt with using simple post-processing.

4.4 Object detection

According to the theory and algorithm described above, after training, we have obtained the following information:

1. The types of the pixels, whether they are static or dynamic;
2. Reference points $Q_1, Q_2, Q_3 \ldots Q_n$ of each target pixel. For dynamic pixels, we pick two or more group of reference points in the corresponding clusters that represent them, respectively.
3. Gaussian distributions formed by the intensity difference between the target pixel and its reference points.

When conducting foreground detection, we calculate the difference between the target point and its reference points. If the difference value still matches the Gaussian distribution that was established after trained, we can treat the target pixel as a background point. Otherwise, if this relative relationship is destroyed, it means that the pixel is likely to be occupied by the foreground object.

$$\omega_k = \begin{cases} 1 & if|P - Q_k - c_k| \geq \eta \cdot \sigma_k \\ 0 & else \end{cases} \tag{14}$$

$$F = \sum_{k=1}^{n} \omega_k \tag{15}$$

$$result = \begin{cases} foreground & if \quad F > th \\ background & else \end{cases} \tag{16}$$

where ω_k is the k-th judgment result, P is the target pixel, Q_k is the intensity at the k-th reference point, c_k is the mean value of the difference between target point and reference point in a Gaussian distribution, σ_k is the corresponding standard deviation, and η is a threshold constant. The detection threshold is controlled by changing the value, thereby optimizing the detection result. n is the total number of reference points. F is the comprehensive judgment result of all sample reference points, and th is the judgment threshold, which means that if the number of positive points is greater than the threshold value th, the pixel is judged to be the foreground.

For dynamic pixels, because we do not know the state when conducting detection, we need to use its two types of reference points for verification. If their relative relationship to both groups of reference points is broken, we can treat the target pixel as a foreground point. The logical expression is as follows:

$$result = \begin{cases} foreground & if \quad result_1 > th \text{ and } result_2 > th \\ background & else \end{cases} \tag{17}$$

where $result_1$ and $result_2$ are the results of the two types of state judgment.

5 Experimental results

To evaluate the proposed method, we selected several representative datasets to test it. Three of them come from the CDnet-2014 dataset, and two are videos taken by us.

Datasets (a), (b), and (c) are the CDnet-2014 datasets. Datasets (b) and (c) are cases where there are a large number of dynamic background points in the scene. Datasets (d) and (e) are the video frames of the scenes taken by us. In data set e, the light source of the scene changes slowly, and the fans in the background are constantly rotating. In data set f, the scene light source changes rapidly several times, and the fan in the background is also constantly rotating. Dataset (a) can be regarded as a baseline, and the datasets (b), (c), (d), and (e) all contain complex dynamic backgrounds.

We use three indicators to evaluate the quality of the test results: precision, recall and F-measure. The formulas for the evaluation measures are as follow.

$$Precision = (\frac{TP}{TP + FP}) \tag{18}$$

$$Recall = (\frac{TP}{TP + FN}) \tag{19}$$

$$F{-}measure = \frac{2Precision \cdot Recall}{Precision + Recall} \tag{20}$$

where *TP*, *FP* and *FN* stand for the numbers of true positive pixels, false positive pixels and false negative pixels, respectively. The *F-measure* is the weighted harmonic mean of the Precision and Recall [48].

We chose some other methods to compare with the proposed method. The GMM [31] and KDE [34], which are two well-known traditional algorithms, are the two main basic standard techniques that are often used to make basic comparisons. ViBe [38], which is the leading unsupervised technique for foreground detection, has a fast speed when conducting detection and has a good effect. The CPB [43] is a recent algorithm for moving object detection using the idea of reference spatial pixels and it has a good effect under strong illumination changes and dynamic backgrounds. FgSegNet_M is a supervised deep learning algorithm [14].

To compare the results of each algorithm under different conditions, we divide the experiment into two parts: offline detection and online detection.

In the offline detection experiment, the scenes are known and the groundtruth is available. We can select any number of image frames and any part of the images in training set as the training set and use it to initialize the model and establish the algorithm models. Its purpose is to achieve the best detection effect and achieve the highest detection accuracy.

Online detection experiments are used to compare the performance of each algorithm in practical applications. When conducting online detection, only events that have occurred are available. In online experiments, we divide the training set into two parts. The first part is considered to be the available image frames that are used to initialize the model or train the network and the second part is considered to be the part that is not yet been used for detection. We considered two situations: seen scenes and unseen scenes. The difference is that some groundtruths have been marked based on the already acquired image frames in seen scene whereas there is no groundtruth in unseen scene. The experiment for the Online detection of an unseen scene is more in line with the actual application scene, and so those results are better able to reflect the effectiveness of the methods.

5.1 Experimental details

In our method, there are some parameters that need to be set manually. In the experimental verification, we chose a unified parameter calculation method. The parameters we need to determine in the experiment are as follows. The number of clusters K, reflecting the number of object properties in the image, is set to 5. The total number of reference points n (in formula (5)) is set to 16. With more reference points, it is more conducive to make the correct judgement of the target point. However, after numerical study, we found that when the number of reference points exceed 15, the influence on the accuracy is minimal. Considering that more reference points demand more calculation, in order to integrate accuracy and calculation efficiency, we have chosen the number of reference points as 16. The judgment threshold value th (in formula (15)) is set to 14. It is correlated with the number of reference points n and reflects the fault tolerance rate of the method. In the experimental evaluation, we set two points as fault tolerance. In fact, in the experiment, when the threshold is set to 10–14, the impact on accuracy did not show a big difference. But when the threshold th is less than 10, the result showed higher recall and lower precision. The value of η (in formula (13)), representing the severity of the judgment, is set to 2. After clustering, the points in the same cluster have been guaranteed to have similar characteristics. Thus, when there is no foreground object, the target point is similar to the reference point, and when foreground object covers the target point, there will be a significant different between target point and reference point. The coefficient η as 2 is adequate for our method.

The environment configuration for the experiment is as follows: The CPU is Intel Core i5-7300HQ, the RAM is 32 G and the GPU is NVIDIA GeForce GTX 1050, GPU memory is 2 G. The algorithm is implemented in MATLAB. The number of frames to initialize the background is 600. The average time taken to build the background model is about 285 s.

5.2 Offline detection

To make the comparison fairer, the number of frames in training sets is set to 600 and they are extracted from the dataset at equal intervals. This is the best condition for the experimental algorithms. The KDE can use the images of each time period in the dataset to establish the sampling set to obtain a more reasonable and credible background probability. For the CPB, the statistics of pixels can be better represented. For FgSegNet_M (supervised method) [14], the images include more foreground objects and complex backgrounds to be learned. For the proposed method, the longer the time span is, the more fully the attributes of image pixels and objects can be reflected. The experimental results are shown in Fig. 8 and Table 1.

From the comparison of experimental results, we can see that our method achieves promising results compared to the classic and state-of-the-art unsupervised methods. In addition, for FgSegNet_M [14], because it is a supervised method and needs the groundtruths for neural network training, it usually has higher accuracy than unsupervised methods. However, our method has a comparable detection quality under strong illumination changes and dynamic backgrounds without the groundtruth or signals.

5.3 Online detection of seen scene and unseen scene

In this part, we take the first 600 frames of the data set as training set to initialize the model, and the rest of the image frames as the test set.

For the GMM and ViBe, their mechanism is as follows: use the first frame to initialize the background model, and then update the model using the input image. Therefore, the offline detection and online detection are implemented in the same way, and so their offline detection and online detection results are the same. For the KDE, CPB, and proposed method, changing the training set or sampling set will have a certain impact on the results. For all the unsupervised methods, since no groundtruth is required, the results

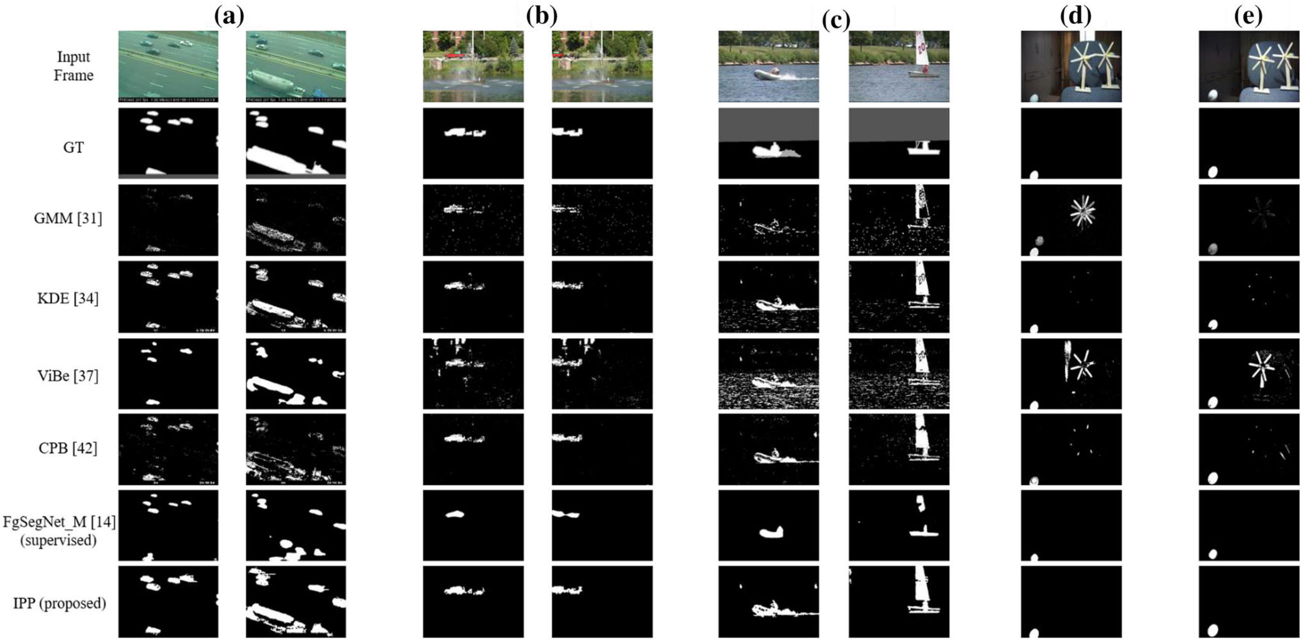


Fig. 8 Visual comparison of various methods for offline foreground detection

Table 1 Comparison of the proposed method with other methods in offline detection using the precision, recall, F-measure

	(a)			(b)			(c)		
	precision	recall	F-measure	precision	recall	F-measure	precision	Recall	F-measure
GMM [31]	0.9455	0.5948	0.7302	0.5468	0.2198	0.3135	0.5406	0.1877	0.2786
KDE [34]	0.9836	0.8817	0.9258	0.8617	0.8042	0.8320	0.5203	0.7756	0.6228
ViBe [37]	0.2328	0.8077	0.3615	0.3042	0.7832	0.4309	0.6172	0.6069	0.6121
CPB [42]	0.8991	0.7524	0.8192	0.9025	0.7037	0.7908	0.7365	0.7003	0.7179
FgSegNet_M [14] (supervised)	**0.9978**	0.9127	**0.9534**	**0.9837**	0.7894	0.8759	**0.9247**	0.8649	**0.8938**
IPP	0.9354	**0.9334**	0.9344	0.9731	**0.9436**	**0.9564**	0.8051	**0.9023**	0.8509
	(d)			(e)			Average		
	precision	recall	F-measure	precision	recall	F-measure	precision	Recall	F-measure
GMM [31]	0.3013	0.6488	0.4777	0.4151	0.5628	0.4776	0.4509	0.4048	0.3703
KDE [34]	0.9033	0.7369	0.8117	0.6872	0.6511	0.6686	0.8156	0.7854	0.7909
ViBe [37]	0.1578	0.8646	0.2658	0.1798	0.7923	0.2924	0.2984	0.7709	0.3925
CPB [42]	0.8127	**0.9107**	0.8589	0.8655	**0.8546**	0.8600	0.8433	0.7843	0.8094
FgSegNet_M [14] (supervised)	**0.9800**	0.8448	**0.9027**	**0.9970**	0.7698	**0.8628**	**0.9766**	0.8363	**0.8977**
IPP	0.9143	0.8793	0.8965	0.9372	0.7334	0.8229	0.9074	**0.8647**	0.8817

in seen scenes (with the groundtruth) and unseen scenes (no groundtruth) have no difference. The experimental results are shown in Fig. 9 and Table 2 We can see from the results that for online detection, due to the limitations of the training set, the detection results of these methods are somewhat affected, but the impact is small. Therefore, we can conclude that in offline detection, unsupervised methods can still maintain their detection quality. In addition, among all the algorithms involved in the experiment, the proposed method performs best.

Regarding FgSegNet_M, it is a supervised method and depends on numerous foreground training samples with groundtruths, which is difficult to achieve in online detection. We experiment in two ways.

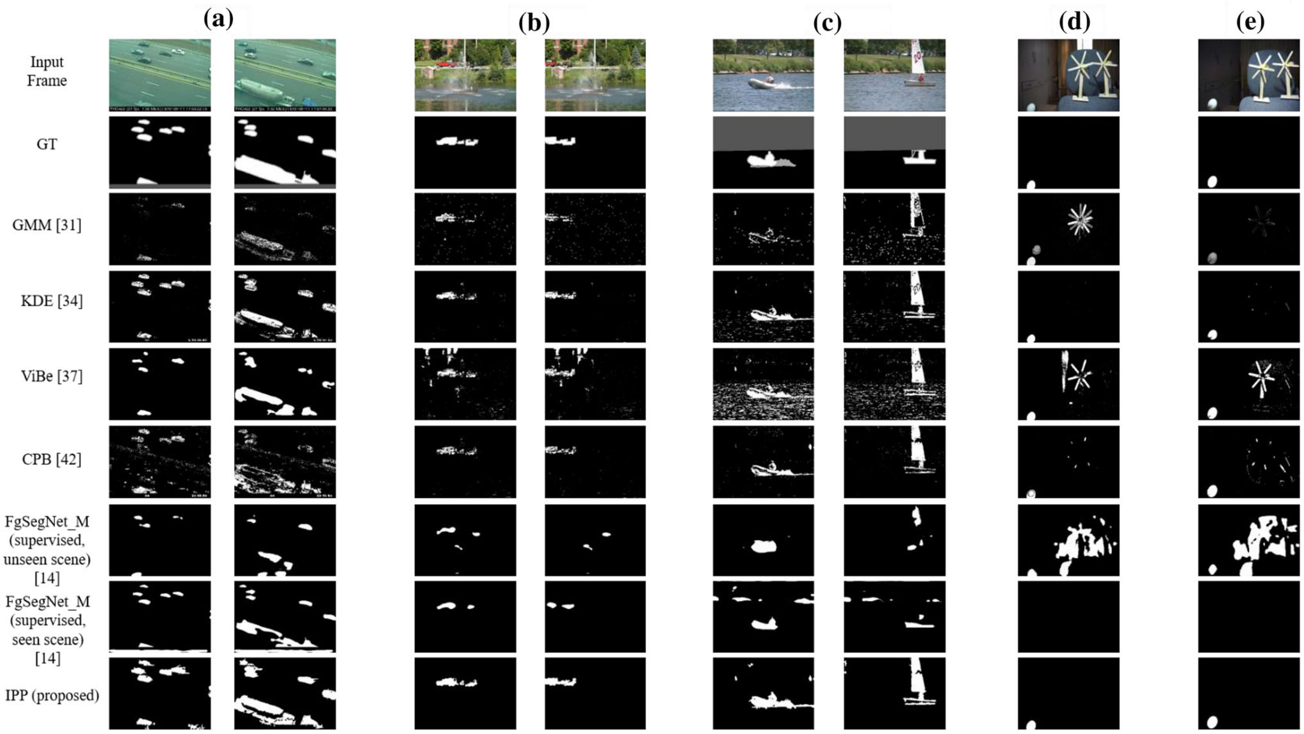

(a)
(b)
(c)
(d)
(e)
Input Frame
GT
GMM [31]
KDE [34]
ViBe [37]
CPB [42]
FgSegNet_M (supervised, unseen scene) [14]
FgSegNet_M (supervised, seen scene) [14]
IPP (proposed)

Fig. 9 Visual comparison of various methods for online foreground detection

Table 2 Comparison of the proposed method with other methods in online detection using the precision, recall, F-measure

	(a)			(b)			(c)		
	precision	recall	F-measure	precision	recall	F-measure	precision	recall	F-measure
GMM	0.9455	0.5948	0.7302	0.5468	0.2198	0.3135	0.5406	0.1877	0.2786
KDE	0.9814	0.8814	0.8397	0.7827	0.8056	0.7940	0.5336	0.7943	0.6383
ViBe	0.2328	0.8077	0.3615	0.3042	0.7832	0.4309	0.6172	0.6069	0.6121
CPB	0.8868	0.7499	0.8126	0.6883	0.9004	0.7802	0.7887	0.6445	0.7094
FgSegNet_M [14] (unseen scene)	0.9916	0.8595	0.9209	0.2636	0.3696	0.3077	0.9288	0.7309	0.8180
FgSegNet_M [14] (seen scene)	**0.9918**	**0.9723**	**0.9810**	**0.9920**	0.7459	0.8193	**0.9965**	0.8071	**0.8780**
IPP	0.9198	0.9340	0.9268	0.9794	**0.9633**	**0.9712**	0.8467	**0.9007**	0.8729

Except for FgSegNet_M, other methods are all unsupervised methods. Since no groundtruth is required, the results of each method in seen scenes (with the groundtruth) and unseen scenes (no groundtruth) have no difference

5.3.1 No groundtruth (unseen scene)

In an unseen environment, there is no groundtruth prepared in advance. Nevertheless, the convolutional neural network has been trained in other scenes and has learned some foreground shape and motion patterns. The experimental results are shown in Fig. 9. We can see from the visual display that some common objects, such as small vehicles and ships, can be detected. However, the quality is poor, as shown in Fig. 9 columns (a), (b), and (c). However, when there is a large area dynamic background in image, the method fails, as shown in Fig. 9 columns (e) and (f).

It can be concluded from both the theory and experiment that the supervised learning method almost fails without the groundtruth. The unsupervised method is obviously better than the supervised learning method.

5.3.2 With the groundtruth (seen scene)

The first 600 frames of the dataset are regarded as the occurrence times, which can be used to produce the groundtruths and conduct training. The rest of the dataset is used for testing.

In the experiments for datasets (a), (b), and (c), because the cars and boats in the scenes have appeared many times in the first 600 frames, after the neural network is trained, they can still be successfully detected, but the detection quality has decreased significantly. In the experiments for datasets (e) and (f), the ball appears very rarely in the training set (the first 600 frames). As a result, the algorithm can hardly detect the existence of balls when conducting detection, as shown in Fig. 9. Therefore, in online detection, if a new object appears in online detection, the supervised method will not be able to detect it unless the object is continuously added to the training set for training, which is undoubtedly difficult to achieve in online detection. In addition, because the training set is optimal, its detection quality is also worse than those of the unsupervised methods.

From above experiments, we can conclude that unsupervised methods are much more effective than supervised methods during online detection and the proposed method performs well in both offline and online detection.

6 Conclusion

In this paper, we propose a new model and provide a new idea for foreground detection at both the pixel and object levels. It does not need to be updated in real time and can be used for online detection. We conducted comparative experiments using different methods and provided the analysis results of supervised methods and unsupervised methods for offline detection and online detection. The proposed model achieves a promising result and performs robust detection under strong illumination changes and dynamic backgrounds.

Funding This study was funded by the National Natural Science Foundation of China (51775498 and 51775497) and the Zhejiang Provincial Natural Science Foundation (LY21E050021).

Declarations

Conflict of interest The authors declares that they have no conflict of interest.

References

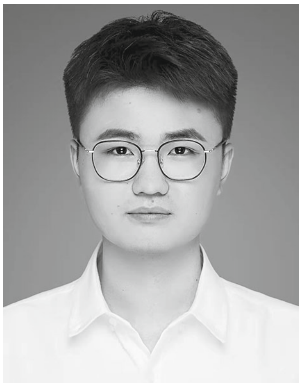
1. Losada, C., Mazo, M., Palazuelos, S.E., Pizarro, D., Marrón, M., Velasco, J.F.: Identification and tracking of robots in an intelligent space using static cameras and an XPFCP. Robot. Auton. Syst. **61**(2), 75–85 (2013). https://doi.org/10.1016/j.robot.2012.11.007
2. C. Ma, X. Yang, Chongyang Zhang, and M.-H. Yang, Long-term correlation tracking. In: 2015 IEEE conference on computer vision and pattern recognition (CVPR), Boston, MA, USA, Jun. 2015, pp. 5388–5396. doi: https://doi.org/10.1109/CVPR.2015.7299177.
3. Amraee, S., Vafaei, A., Jamshidi, K., Adibi, P.: Anomaly detection and localization in crowded scenes using connected component analysis. Multimed. Tools Appl. **77**(12), 14767–14782 (2018). https://doi.org/10.1007/s11042-017-5061-7
4. Bouwmans, T.: Traditional and recent approaches in background modeling for foreground detection: an overview. Comput. Sci. Rev. **11–12**, 31–66 (2014). https://doi.org/10.1016/j.cosrev.2014.04.001
5. T. Bouwmans and B. Garcia-Garcia, "Background Subtraction in Real Applications: Challenges, Current Models and Future Directions," *[cs]*, Jan. 2019, Accessed: May 25, 2021. [Online]. Available:
6. Y.-W. Choi, Y.-S. Chung, S.-I. Lee, and S.-G. Lee, "Rear object detection method based on optical flow and vehicle information for moving vehicle. In: 2017 ninth international conference on ubiquitous and future networks (ICUFN), Jul. 2017, pp. 203–205. doi: https://doi.org/10.1109/ICUFN.2017.7993776.
7. Liang, C.-W., Juang, C.-F.: Moving object classification using a combination of static appearance features and spatial and temporal entropy values of optical flows. IEEE Trans. Intell. Transp. Syst. **16**(6), 3453–3464 (2015). https://doi.org/10.1109/TITS.2015.2459917
8. Sengar, S.S., Mukhopadhyay, S.: Detection of moving objects based on enhancement of optical flow. Optik **145**, 130–141 (2017). https://doi.org/10.1016/j.ijleo.2017.07.040
9. O. ELHarrouss, D. Moujahid, S. E. Elkaitouni, and H. Tairi, "Moving objects detection based on thresholding operations for video surveillance systems. In: 2015 IEEE/ACS 12th international conference of computer systems and applications (AICCSA), Nov. 2015, pp. 1–5. doi: https://doi.org/10.1109/AICCSA.2015.7507180.
10. Yeh, C.-H., Lin, C.-Y., Muchtar, K., Lai, H.-E., Sun, M.-T.: Three-pronged compensation and hysteresis thresholding for moving object detection in real-time video surveillance. IEEE Trans. Industr. Electron. **64**(6), 4945–4955 (2017). https://doi.org/10.1109/TIE.2017.2669881
11. M. Braham and M. Van Droogenbroeck, Deep background subtraction with scene-specific convolutional neural networks. In: 2016 international conference on systems, signals and image processing (IWSSIP), Bratislava, Slovakia, May 2016, pp. 1–4. doi: https://doi.org/10.1109/IWSSIP.2016.7502717.
12. K. Lim, W.-D. Jang, and C.-S. Kim, Background subtraction using encoder-decoder structured convolutional neural network. In: 2017 14th IEEE international conference on advanced video and signal based surveillance (AVSS), Aug. 2017, pp. 1–6. doi: https://doi.org/10.1109/AVSS.2017.8078547.
13. Wang, Y., Luo, Z., Jodoin, P.-M.: Interactive deep learning method for segmenting moving objects. Pattern Recogn. Lett. **96**, 66–75 (2017). https://doi.org/10.1016/j.patrec.2016.09.014
14. Lim, L.A., Keles, H.Y.: Foreground segmentation using a triplet convolutional neural network for multiscale feature encoding. Pattern Recogn. Lett. **112**, 256–262 (2018). https://doi.org/10.1016/j.patrec.2018.08.002
15. Hu, Z., Turki, T., Phan, N., Wang, J.T.L.: A 3D atrous convolutional long short-term memory network for background subtraction. IEEE Access **6**, 43450–43459 (2018). https://doi.org/10.1109/ACCESS.2018.2861223
16. Sakkos, D., Liu, H., Han, J., Shao, L.: End-to-end video background subtraction with 3d convolutional neural networks. Multimed. Tools Appl. **77**(17), 23023–23041 (2018). https://doi.org/10.1007/s11042-017-5460-9

17. Bouwmans, T., Javed, S., Sultana, M., Jung, S.K.: Deep neural network concepts for background subtraction: a systematic review and comparative evaluation. Neural Netw. **117**, 8–66 (2019). https://doi.org/10.1016/j.neunet.2019.04.024
18. M. Mandal and S. K. Vipparthi, An empirical review of deep learning frameworks for change detection: model design, experimental frameworks, challenges and research needs. *IEEE Trans. Intell. Transp. Syst.*, pp. 1–22, 2021, doi: https://doi.org/10.1109/TITS.2021.3077883.
19. Patil, P.W., Murala, S.: MSFgNet: a novel compact end-to-end deep network for moving object detection. IEEE Trans. Intell. Transport. Syst. **20**(11), 4066–4077 (2019). https://doi.org/10.1109/TITS.2018.2880096
20. Lim, L.A., Yalim Keles, H.: Foreground segmentation using convolutional neural networks for multiscale feature encoding. Pattern Recogn. Lett. **112**, 256–262 (2018). https://doi.org/10.1016/j.patrec.2018.08.002
21. Lim, L.A., Keles, H.Y.: Learning multi-scale features for foreground segmentation. Pattern Anal. Appl. **23**(3), 1369–1380 (2020). https://doi.org/10.1007/s10044-019-00845-9
22. Chen, Y., Wang, J., Zhu, B., Tang, M., Lu, H.: Pixelwise deep sequence learning for moving object detection. IEEE Trans. Circuits Syst. Video Technol. **29**(9), 2567–2579 (2019). https://doi.org/10.1109/TCSVT.2017.2770319
23. Cai, X., Han, G.: Background subtraction based on modified pulse coupled neural network in compressive domain. IEEE Access **8**, 114422–114432 (2020). https://doi.org/10.1109/ACCESS.2020.3003724
24. M. O. Tezcan, P. Ishwar, and J. Konrad, BSUV-Net: a fully-convolutional neural network for background subtraction of unseen videos. In: 2020 IEEE winter conference on applications of computer vision (WACV), Snowmass Village, CO, USA, Mar. 2020, pp. 2763–2772. doi: https://doi.org/10.1109/WACV45572.2020.9093464.
25. Tezcan, M.O., Ishwar, P., Konrad, J.: BSUV-Net 2.0: spatio-temporal data augmentations for video-agnostic supervised background subtraction. IEEE Access **9**, 53849–53860 (2021). https://doi.org/10.1109/ACCESS.2021.3071163
26. Patil, P.W., Dudhane, A., Murala, S., Gonde, A.B.: Deep adversarial network for scene independent moving object segmentation. IEEE Signal Process. Lett. **28**, 489–493 (2021). https://doi.org/10.1109/LSP.2021.3059195
27. Giraldo, J.H., Javed, S., Bouwmans, T.: Graph moving object segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **01**, 1–1 (2020). https://doi.org/10.1109/TPAMI.2020.3042093
28. J. H. Giraldo and T. Bouwmans, “GraphBGS: Background Subtraction via Recovery of Graph Signals,” *[cs]*, Nov. 2020, Accessed: Jul. 05, 2021. [Online]. Available:
29. J. H. Giraldo and T. Bouwmans, Semi-supervised background subtraction of unseen videos: minimization of the total variation of graph signals. In: 2020 IEEE international conference on image processing (ICIP), Oct. 2020, pp. 3224–3228. doi: https://doi.org/10.1109/ICIP40778.2020.9190887.
30. B. Azeez and F. Alizadeh, Review and classification of trending background subtraction-based object detection techniques. In: 2020 6th international engineering conference “sustainable technology and development" (IEC), Erbil, Iraq, Feb. 2020, pp. 185–190. doi: https://doi.org/10.1109/IEC49899.2020.9122929.
31. Bao, J., Jia, Y., Cheng, Y., Xi, N.: Saliency-guided detection of unknown objects in RGB-D indoor scenes. Sensors (2015). https://doi.org/10.3390/s150921054
32. Haines, T.S.F., Xiang, T.: Background subtraction with dirichlet-process mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **36**(4), 670–683 (2014). https://doi.org/10.1109/TPAMI.2013.239
33. Bianco, S., Ciocca, G., Schettini, R.: Combination of video change detection algorithms by genetic programming. IEEE Trans. Evol. Comput. **21**(6), 914–928 (2017). https://doi.org/10.1109/TEVC.2017.2694160
34. R. Wang, F. Bunyak, G. Seetharaman, and K. Palaniappan, Static and Moving Object Detection Using Flux Tensor with Split Gaussian Models. In: 2014 IEEE conference on computer vision and pattern recognition workshops, Jun. 2014, pp. 420–424. doi: https://doi.org/10.1109/CVPRW.2014.68.
35. Elgammal, A., Duraiswami, R., Harwood, D., Davis, L.S.: Background and foreground modeling using nonparametric kernel density estimation for visual surveillance. Proc. IEEE **90**(7), 1151–1163 (2002). https://doi.org/10.1109/JPROC.2002.801448
36. S. Liao, G. Zhao, V. Kellokumpu, M. Pietikäinen, and S. Z. Li, Modeling pixel process with scale invariant local patterns for background subtraction in complex scenes. In: 2010 IEEE computer society conference on computer vision and pattern recognition, Jun. 2010, pp. 1301–1306. doi: https://doi.org/10.1109/CVPR.2010.5539817.
37. Jiang, S., Lu, X.: WeSamBE: a weight-sample-based method for background subtraction. IEEE Trans. Circuits Syst. Video Technol. **28**(9), 2105–2115 (2018). https://doi.org/10.1109/TCSVT.2017.2711659
38. Barnich, O., Van Droogenbroeck, M.: ViBe: a universal background subtraction algorithm for video sequences. IEEE Trans. Image Process. **20**(6), 1709–1724 (2011). https://doi.org/10.1109/TIP.2010.2101613
39. M. Hofmann, P. Tiefenbacher, and G. Rigoll, Background segmentation with feedback: the pixel-based adaptive segmenter. In: 2012 IEEE computer society conference on computer vision and pattern recognition workshops, Jun. 2012, pp. 38–43. doi: https://doi.org/10.1109/CVPRW.2012.6238925.
40. Zhong, Z., Zhang, B., Lu, G., Zhao, Y., Xu, Y.: An adaptive background modeling method for foreground segmentation. IEEE Trans. Intell. Transp. Syst. **18**(5), 1109–1121 (2017). https://doi.org/10.1109/TITS.2016.2597441
41. Zhao, X., Satoh, Y., Takauji, H., Kaneko, S., Iwata, K., Ozaki, R.: Object detection based on a robust and accurate statistical multi-point-pair model. Pattern Recogn. **44**(6), 1296–1311 (2011). https://doi.org/10.1016/j.patcog.2010.11.022
42. Liang, D., Kaneko, S., Hashimoto, M., Iwata, K., Zhao, X.: Co-occurrence probability-based pixel pairs background model for robust object detection in dynamic scenes. Pattern Recogn. **48**(4), 1374–1390 (2015). https://doi.org/10.1016/j.patcog.2014.10.020
43. W. Zhou, S. Kaneko, M. Hashimoto, Y. Satoh, and D. Liang, Foreground Detection based on co-occurrence background model with hypothesis on degradation modification in background changes. In: 2018 12th France-Japan and 10th Europe-Asia Congress on Mechatronics, Sep. 2018, pp. 77–82. doi: https://doi.org/10.1109/MECATRONICS.2018.8495871.
44. Vijayan, M., Ramasundaram, M.: Moving object detection using vector image model. Optik **168**, 963–973 (2018). https://doi.org/10.1016/j.ijleo.2018.05.012
45. Xu, Y., Ji, H., Zhang, W.: Coarse-to-fine sample-based background subtraction for moving object detection. Optik **207**, 164195 (2020). https://doi.org/10.1016/j.ijleo.2020.164195
46. Giveki, D., Soltanshahi, M.A., Yousefvand, M.: Proposing a new feature descriptor for moving object detection. Optik **209**, 164563 (2020). https://doi.org/10.1016/j.ijleo.2020.164563
47. A. Elgammal, D. Harwood, and L. Davis, Non-parametric model for background subtraction. In: computer vision — ECCV 2000, Berlin, Heidelberg, 2000, pp. 751–767. doi: https://doi.org/10.1007/3-540-45053-X_48.

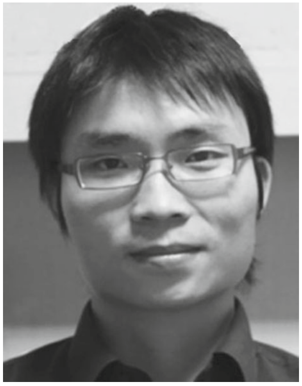
48. Y. Wang, P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar, CDnet 2014: an expanded change detection benchmark dataset. In: 2014 IEEE conference on computer vision and pattern recognition workshops, Columbus, OH, USA, Jun. 2014, pp. 393–400. doi: https://doi.org/10.1109/CVPRW.2014.126.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xinyue Zhao received her M.S. degree in Mechanical Engineering from Zhejiang University, China in 2008, and her Ph.D degree in Graduate School of Information Science and Technology from Hokkaido University, Japan in 2012. She is currently an associate professor in the Department of Mechanical Engineering, Zhejiang University, China. Her research interests include machine learning and machine vision.



Guangli Wang received the B.Sc. Degree in mechanical engineering from Zhejiang University, China, in 2020.He is currently working toward the M.S. degree in mechanical engineering at Zhejiang University, Hangzhou, China. His research interests include pattern recognition, machine learning and machine vision.



Zaixing He received his B.Sc. and M.Sc. degrees in Mechanical Engineering from Zhejiang University, China in 2006 and 2008, respectively. He received his Ph. D. degree in 2012 from the Graduate School of Information Science and Technology, Hokkaido University, Japan. He is currently an associate professor in the Department of Mechanical Engineering, Zhejiang University. He is senior member of IEEE. His research interests include deep learning, robotic vision, visual intelligence of manufacturing equipment's, and image understanding.

Dong Liang received B.E. degree and M.E. degree from Lanzhou University (LZU), China, in 2008 and 2011, respectively. He received Ph.D. from Graduate School of Information Science and Technology, Hokkaido University, Japan, in 2015. He is now an assistant professor in Department of Computer Science & Engineering, Nanjing University of Aeronautics and Astronautics (NUAA), China. His research interests include computer vision and pattern recognition.

Shuyou Zhang received his M.S. degree in Mechanical Engineering and the Ph.D. degree in State Key Lab. Of CAD&CG from Zhejiang University, China, in 1991 and 1999, respectively. He is currently a professor in the Department of Mechanical Engineering, Zhejiang University, China. He is also the administer of Institute of Design Engineering in Zhejiang University. His research interests include human-robot interaction, semantic segmentation and engineering and computer graphics.

Jianrong Tan is an academician of China Engineering Academy, and is currently a professor at State Key Laboratory of CAD&CG, Zhejiang University. He received the M.S. from Huazhong University of Science and Technology, Wuhan, China in 1987 and the Ph.D. from Zhejiang University in 1992. His main research interests include virtual-reality-based simulation, machine learning, CAX and robotics.