A Complex Dynamic Modelling Method Based on Adaptive Visual Background Extractor

Dali Qiao, Xinyu Tian, Qinghe Zheng*, Weiguang Wang

Abstract—**In today's rapidly changing computer vision and motion capture technology, the detection of moving objects has become an indispensable key link in numerous scenarios. Visual background extractor (Vibe) is a kind of method with low computation, well real-time performance, simple and efficient characteristics. Vibe has been applied in various fields. It can be classified as three stages of the initialization of the background model, extracting the foreground pixels and updating the background model to accurately capture the dynamics in the image. In this paper, the neighborhood random selection way and secondary sampling factors are proposed to select pixels for filling and updating the sample library. Firstly, the dataset with the same size as the input image is established, and then it is filled with original images based on eight-neighborhood random selection method (ENRSM). Even if the sample size is limited, it is still enough to fill the entire sample database Next, the initialization step of the background model is performed, which records the value of each pixel at the same position at the past moment, or the value of the eight-neighborhoods of the pixel. To distinguish between foreground and background, we will calculate the Euclidean distance between the individual pixels in the new input image and the pixels in the pixel sample database. Based on the calculated distance, we classify the pixels, which enables the segmentation of the foreground image. The set subsampling factor is used to update the background sample library. The detection results of the proposed method illustrate the adaptability of scene, real-time monitoring and robustness facing complex environments. Vibe method can be not affected by the speed of object movement, but there still exists the ghost phenomenon in the foreground detection image. The resolution of light and image also affect the detection results.**

Index Terms—**dynamic detection, background modelling, neighborhood random selection, secondary sampling factor, foreground image segmentation.**

I. Introduction

IN the continuous evolution of computer vision and dynamic capture technology, moving object detection has always been a high-profile research direction in the field of machine vision [1][2]. In this field, the widely used technical methods include optical flow method [3][4], the inter-frame difference method [5][6], and the background difference method [7]. Optical flow technology [4] requires powerful computing power and specific hardware devices to accurately measure the optical flow information of image pixels to achieve the detection of moving targets. In the detection process, the frame difference method [5][6] often uses time intervals or consecutive multi-frame images for differential comparison. Although the principle based on this method is relatively straightforward and the calculation process is relatively simple, the image outline generated by it may be discontinuous, and there will be holes in the outline. By comparison, background subtraction methods [8] [9][10][11] make difference between the current video frame image and the background image, so as to obtain the image of the moving object, but it has higher requirements for the background image. The background image needs to remain motion-free and instantly adjust to match changes in the surveillance scene. Due to its low computational complexity, efficient running rate, and excellent detection performance, the Vibe algorithm has attracted extensive attention and practical application in the academic community, and the relevant research results can be found in Ref. [12], [13], and [14]. In the field of computer vision, especially for moving object detection, the Vibe algorithm has shown significant contributions. Table 1 summarizes the core features of the Vibe algorithm, which provide a solid foundation for follow-up processing tasks such as target tracking, and are

Table 1. Characteristics of Vibe method.

Characteristic	Achieved effect
adaptation	The Vibe method can adapt to target changes under different lighting conditions and detect abnormal behaviors, such as intrusion and traffic violations, which makes it more effective for object detection and object tracking under various environmental conditions.
instantaneity	The Vibe continuously performs data processing on each frame of the input video. The processing speed of the data is millisecond level, and the input video is detected in real time. This method can detect all kinds of emergent situation and has good real-time performance.
robustness	The Vibe uses a pixel-based background modelling approach to distinguish between the foreground and background by building a background model for each pixel. This method possesses well robustness to illumination change, background disturbance and other factors, and can accurately detect the moving objects in complex environments.
innovation	The Vibe uses non-parametric model in background modeling and updates the background model by randomly selecting the background pixel values. Compared with the traditional parameter modelling method, this method is more flexible and adaptable, and can better deal with various complex situations in practical applications.

Manuscript received June 3, 2024; revised January 9, 2025.

This work was supported in part by Shandong Provincial Natural Science Foundation under Grant No. ZR2023QF125.

Dali Qiao is an undergraduate student of Shandong Management University, Jinan 250357, China. (email: 1843718527@qq.com)

Xinyu Tian is a lecturer of Shandong Management University, Jinan 250357, China. (email: txy@sdmu.edu.cn)

Qinghe Zheng is an associate professor of Shandong Management University, Jinan 250357, China. (corresponding author to provide phone: 89636095; fax: 89636095; e-mail: zqh@sdmu.edu.cn)

Weiguang Wang is an associate professor of Shandong Management University, Jinan 250357, China. (email: wwg@sdmu.edu.cn)

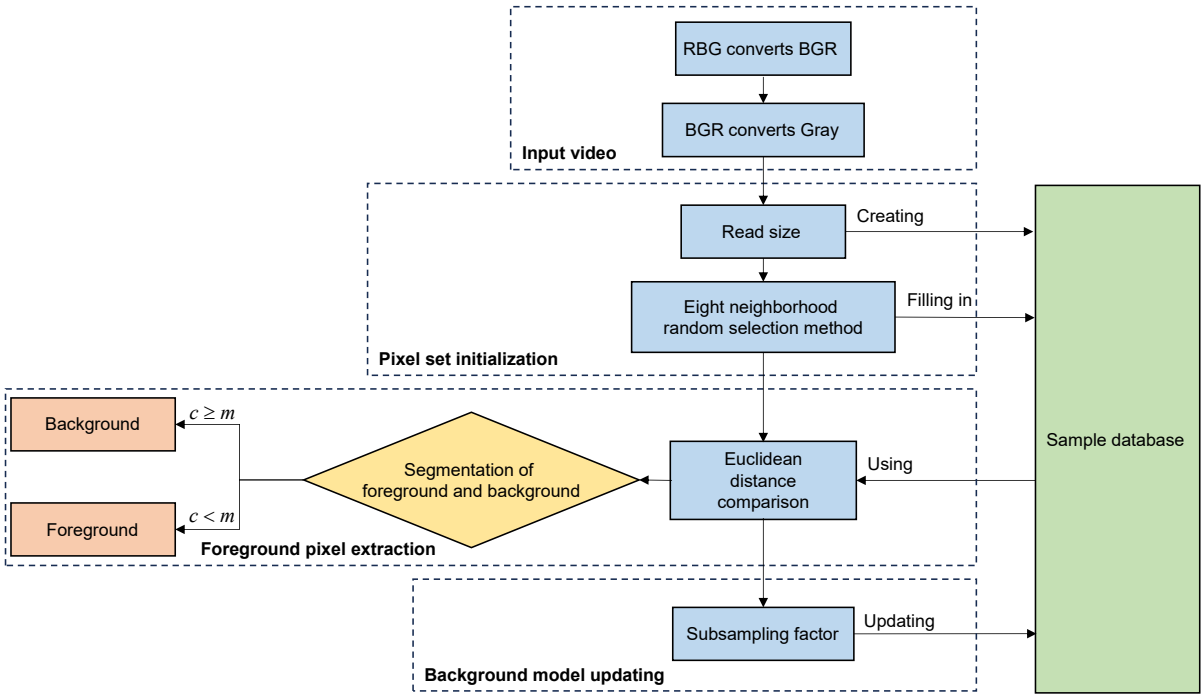

Fig. 1. Flowchart of dynamic modelling method.

widely used in many important scenarios such as intelligent monitoring systems [15], unmanned aerial vehicle applications [16], aerospace applications [17], and national defense and military [18].

The unique nature of the Vibe algorithm in the field of computer vision lays an important foundation for the subsequent execution of tasks such as object tracking [19]. For example, the Vibe method can be used for traffic monitoring [20] and vehicle identification [21] in the field of intelligent and efficient transportation [22], to help traffic management department improve traffic management and safety performance. The Vibe can also accurately capture all the dynamic effective information in the storage, greatly saving storage space, and it is more convenient to access the saved data. In the field of machine vision, Vibe can be used for tasks including human pose estimation [23][24], gesture recognition [25] and behavior analysis [26], which provides a powerful tool for various applications. In addition, Vibe can also be combined with other algorithms, such as Canny edge detection method [27], image mosaics [28], etc.

After entering the 21st century, the rapid progress of computer network and multimedia communication has prompted people's demand for information transmission speed to increase. Under the wave of technological advancement, people are increasingly preferring more intuitive and expressive forms of video communication such as animation, film and television works, video calls and video surveillance than traditional means of communication such as voice, pictures and text. The emergence of video communication has produced a large amount of video data, especially in the field of video surveillance. Video surveillance is a key tool to maintain social order today. There are a variety of camera equipment in roads, stations, campuses, airports, office buildings and other scenes. There is a widespread demand for motion capture technology in various fields such as healthcare, public security, education, and military [33][34][35]. Currently, most devices operate around the clock and are dedicated to collecting video data to ensure that information is fully recorded without missing any details. However, this mode of operation inevitably leads to a sharp increase in the amount of information stored and the inconvenience of retrieving specific information.

Video is essentially a sequence of images, a continuous, dynamic, multimedia information presentation form that contains images and sound [36]. It has the characteristics of continuity, dynamics, combination of image and sound, encoding and storage, multimedia and transmission. A video is composed of a series of static images (frames) that are played continuously at a certain speed (frame rate), resulting in a dynamic effect. When the frame rate is high enough, the human eye will perceive these static images as continuous motion. It can capture and display the changes of objects or scenes in the passage of time, such as object movement [37], expression transformation [38], and scene switching [39]. Motion capture module captures these changes in the video exactly. These changes are identified as foreground to show to the user. The essence of an image is a giant matrix. Each matrix element represents a pixel value in the image. The Vibe method can be regarded as a pixel-level dynamic modelling approach that identifies and processes each pixel separately, which makes Vibe perform well in dynamic detection tasks [40][41][42].

The paper is organized as follows. Section I introduces the research background and the related work, and outlines the characteristics of the Vibe method. Section II presents the adopted Vibe method for dynamic modelling. Section III describes the experimental results and corresponding analysis. Finally, conclusions and future work are drawn in Section IV.

II. Adaptive Visual Background Extractor

Vibe's motion capture technology is based on dividing video content into two parts: a static background and a

Fig. 2. Flowchart of video pre-processing.

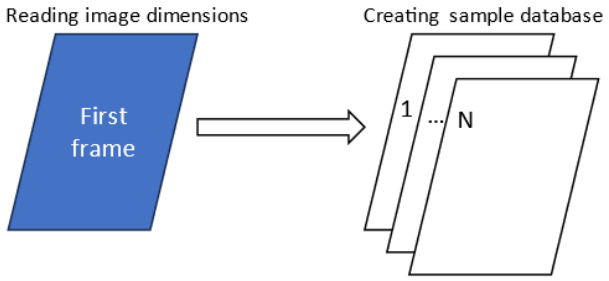

Fig. 3. Illustration of creating sample database.

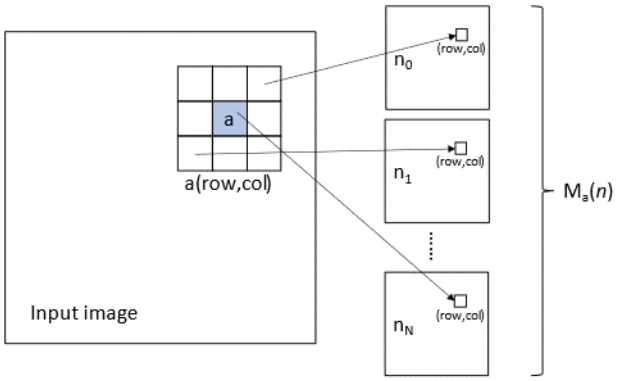

Fig. 4. Sample set of initialized pixel a, pixel values n_i, and sample database $M_a(n)$.

dynamic foreground. In order to deal with the instability of pixel changes, we have adopted ENRSM in the image update process. The whole operation process mainly consists of three major links. After summarizing the entire process, we highlighted the application of ENRSM, the initial establishment of the pixel sample database, the separation of foreground pixels, and the real-time updating of pixels.

As shown in Fig. 1, the format of each frame is firstly converted from RBG to BGR format to adapt to the compiler during the video input. Then the BGR images are converted to gray-format for easier data processing. The dimensions of the video are read when the video is input, and a sample library with the same dimensions as the video is created. The eight-neighborhood random selection method is used to fill the sample library with the pixels of the original video. The distinction between foreground and background is achieved by calculating the Euclidean distance between the sample pixel and the new pixel. When updating, the background sample library is randomly updated by the secondary sampling factor.

Vibe is a calculation method based on grayscale images. The image information used is a single-channel grayscale image (pixel values are between 0-255). The grayscale image data information is relatively simple and the processing speed is fast.

A. Eight-neighborhood random selection method (ENRSM)

Suppose a pixel a (row, col) in the first frame image, a total of 9 pixels are randomly selected from the pixel a and its eight-neighborhood pixels as the *random* selection object of the pixel sample set $M_a(n)$. To fill the pixel sample set $M_a(n)$, randomly select pixels to fill the positions randomly assigned in N matrices. The *random* selection of pixel values is done by creating a library *coff* with library capacity *nc*. There are only -1, 0, and 1 in the library *coff* to ensure that the pixel of n_i is selected among the pixels of pixel a and its 8 neighborhoods. Then we produce one of the numbers *random*, which is uniformly distributed between 0 and *nc*.

As shown in Eq. (1), the random number in the library *coff* is selected and superimposed with the original row number of pixel a, and the *row* coordinate *nrow* to be obtained is finally locked.

$$nrow=coff(random)+row \tag{1}$$

As shown in Eq. (2), a new number *random* is generated for the column coordinates, and the *random* number in the library *coff* is selected to be superimposed with the original column number *col* of pixel a. Finally, the column coordinate *nlow* to be obtained is locked.

$$nlow=coff(random)+low \tag{2}$$

Through the row coordinates *nrow* and column coordinates *nlow*, specific pixel coordinates can be finally determined.

B. Background model initialization

Video is a sequence of pictures in chronological order, one picture is one frame. The Vibe method processes every frame, every image. The essence of a picture is a matrix, and each pixel is a point in the matrix. The Vibe method processes every pixel in the matrix and the whole video at the pixel level, which ensures that there will be no omissions in the video inspection. In Vibe, background model initialization conditions are particularly simple, requiring only the first frame of the video image to complete the initialization, which allows the method to start up in a very short time and quickly get into working condition.

In Fig. 2, we show the video input after receiving video data. The video data is first converted into an image. Since the OpenCV library is used to process the data, the input video is first converted into the three-channel video data in BGR format. Then the weighted summation of the values of the three-color channels of BGR is converted into gray level map by Eq. (3) to facilitate the data processing, so that the method can process the data faster.

$$Gray = 0.2989\times R + 0.2870\times G + 0.1140\times B \tag{3}$$

where *Gray* represents the gray value, and R, G, and B represent the values of the red, green, and blue channels (in the range of 0-255), respectively. These weights (i.e., 0.2989, 0.5870, and 0.1140) are optimized based on the human eye's sensitivity to different colors. The converted grayscale image is visually close to the human eye's perception of the original color image.

Each pixel information of a grayscale map is a single grayscale value. The advantages of converting the image to grayscale map are as follows:

1. The computational complexity can be reduced.
2. The storage space and transmission bandwidth can be saved.
3. The processing of all the texture, shape and structure is

more efficient.

As shown in Fig. 3, the background model is initialized by first reading the dimensions of the input video. Then we can create n matrices of 0 pixels with the same size as the image in $M(n)$. As shown in Fig. 4, the initialization process of one pixel is taken as an example, and the initialization process of all other pixels is the same.

In the two-dimensional matrix of the grayscale image, as shown in Eq. (4), the pixels b (*nrow, nlow*) locked by the eight-neighborhood random selection method are stored in the pixel sample set $M_a(n)$ respectively. In other words, the position of N matrix original pixels a (*row*, *col*) is filled in, i.e., the initialization of a pixel is complete.

$$M_a(n) = \{n_1, n_2, ..., n_{N-1}, n_N\} \tag{4}$$

where $M_a(n)$ denotes the sample library of pixel a, n_i is the i-th sample of pixel a, i = 1, 2, ..., N where N is the total number of samples.

To initialize the background model, a uniform operation needs to be performed on all pixels. In following detection process, the background model saves N sample values for each pixel.

C. Foreground pixel extraction

As shown in the Fig. 5, the essence of Vibe method to segment foreground images is to classify all pixels of the current frame image into foreground and background. At the end of classification, the foreground image is segmented. In the classification stage, the gray value of pixel is compared and analyzed with each sample in the corresponding sample database one by one to determine whether the pixel belongs to the foreground category.

Grayscale images use grayscale color coding. In the grayscale image, each pixel of the image has only one sampled color, and the image is displayed as a grayscale from the darkest black (0) to the brightest white (255), similar to black and white photos. Gray scale is ranging from 0 in white to 255 in black, so black and white pictures are also called gray images.

As shown in Fig. 6, the distance between pixel a and sample n_i in the sample library $M_a(n)$ is different. The sample pixel n_i in the sample library is partially located within the background pixel range sphere $V_r(a)$, and partially outside the background pixel range sphere $V_r(a)$, that is, outside the radius. The number of sample pixels within the radius is equal to the background matching number c of the pixel point, as given by

$$c = V_r(a) \cap \{n_0, n_1, n_2, n_3 ... n_N\} \tag{5}$$

Using the gray value of the pixel, the Euclidean distance D between the current pixel a and the sample pixel n_i is calculated by

$$D = |a - n_i|, \text{ for } i = 1, 2, ..., N \tag{6}$$

The difference between two pixels is positively correlated with the distance D between them.

The determination of the background matching number c follows the rules defined in Eq. (7). Specifically, if the distance between a pixel and the sample in the background model is less than the preset threshold r, the situation is regarded as a successful match, and the background match number c is incremented by 1 accordingly. Conversely, if the distance is more than r, the match is considered unsuccessful, and the background match number c remains unchanged.

$$c = \begin{cases} c = c+1, \text{ if } D < r \\ c \quad\quad\quad, \text{ if } D \geq r \end{cases} \tag{7}$$

When the matching number c between the pixel and the sample pixel < m, it can be judged as the foreground pixel. if the background matching number c between pixel a and sample pixel n_i is greater than or equal to the set threshold m, the pixel can be determined to be the background pixel, as shown by

$$a = \begin{cases} g_b, \text{ if } c \geq m \\ g_f, \text{ if } c < m \end{cases} \tag{8}$$

where a represents the pixel being processed, g_b represents the background point, g_f is used to represent the foreground point, and m is the determined threshold.

D. Background model update

The pixel sample library $M(n)$ is changing at any given time. When judged as foreground, exceeding the foreground threshold will force a sample library update. There is a probability of updating the sample library by secondary sampling factor at any time after determining the background pixel. Through updating the background model continuously, any foreground changes are not missed.

When a new pixel is segmented into foreground, the background model is updated based on the historical value of the pixel.

We set a threshold for foreground matching, denoted as g_f. When a pixel is assigned to the foreground category g_f by judgment, we increment the count p by which the pixel

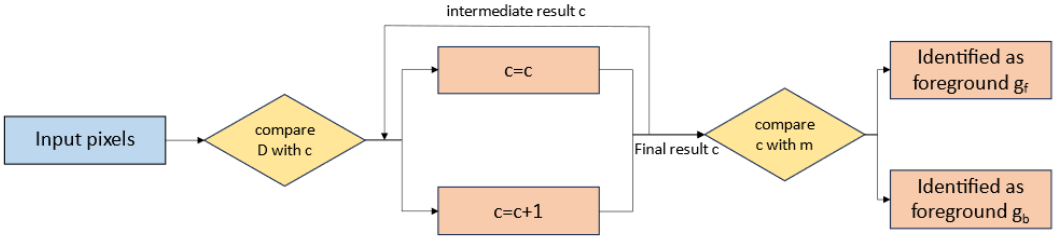

Fig. 9. Flowchart of extracting foreground pixels.

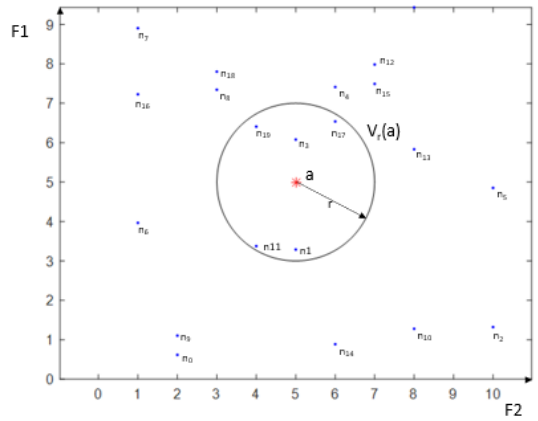

Fig. 6 Vibe pixel foreground and background segmentation model. F1 and F2 are the color distances. a is the pixel to be classified. r represents the radius of the set distance n_i denotes the pixel value stored in the library, i is the code name of the sample, i= 0,1,2... , N. $V_r(a)$ is the background pixel point range sphere.

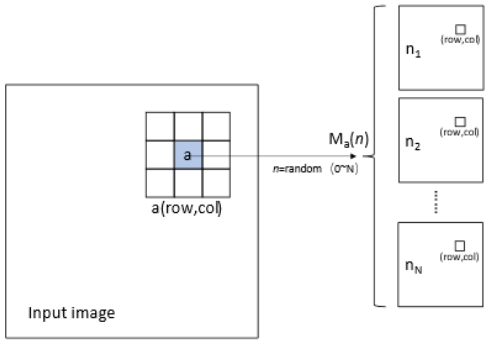

Fig. 7 Forced update of the sample library when the foreground matching number is greater than the foreground matching threshold, pixel a (row, col), pixel value n_i, sample library $M_a(n_i)$.

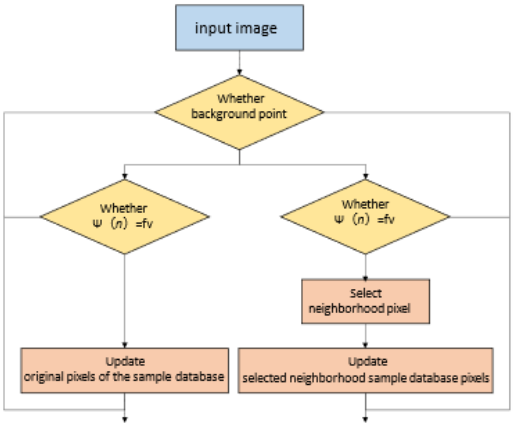

Fig. 8 Flowchart of updating the background model using the subsampling factor.

matches the foreground. When the number of consecutive times a pixel is detected as foreground is greater than the foreground matching threshold p_f, the background sample library is forced to be updated once. As shown in Fig. 7, the pixels of the current pixel are randomly filled into a pixel sample n_i. Static regions (background) are prevented from being misclassified as moving regions (foreground).

As shown in Fig. 8, no threshold is set for the update of background pixels, and background pixels will be updated continuously. Every time a pixel is determined to be a background pixel, the background pixel must update its corresponding position in the sample library. A subsampling factor Ψ is set in the method. Under the condition that pixel a is judged to be the background, the sample in the sample

Algorithm1 The calculation process of updating the sample model using the subsampling factor

Input: image matrix **M**

If: Pixel a is the background point

If meet the condition $\Psi(n) = fv$ **execute** Compare the randomly selected elements in the selection library $\Psi(n)$ with the adaptation value fv.

Use pixel a to update a random sample $M_a(n_i)$ in the corresponding sample library.

$M_a(n_i)=a$ $(i \in (0\sim N))$

Output：background model $M_a(n_i)$

End if

With probability 1/Ψ a sample model with eight neighborhoods is updated

If meet the condition $\Psi(n) = fv$ **execute**

The pixel $a(row, low)$ is selected by eight neighborhood selection method to obtain $b(nrow, nlow)$;

Pixel a was used to update a random sample M_b (n_i) corresponding to pixel b in the sample database;

Output: Background model $M_b(n_i)$

End if

End

database $M_a(n_i)$ is randomly selected and updated with a probability of 1/Ψ. The subsampling factor alone possesses a selection library $\Psi(n)$, $(n \in (0 \sim \Psi))$. Choose the total number of Ψ in the library $\Psi(n)$.

An adaptation value fv is set when a randomly selected element in the selection library $\Psi(n)$ matches the adaptation value fv. As shown in Eq. (9) of Algorithm I, a pixel a is used to update a random sample $M_a(n_i)$ in the corresponding sample library.

$$M_a(n_i) = a, \text{ for } i = 1, 2, ..., N \tag{9}$$

When the background pixel is accurately determined, the corresponding position information of the pixels in the eight neighborhoods will be unconditionally updated to the new pixel with a probability of 1/Ψ in the eight-neighborhood, regardless of whether the pixel is updated or not. Firstly, the secondary sampling factor is used to decide whether the update should be carried out. After the update is determined, a neighborhood pixel b of pixel a is randomly selected by ENRSM. A sample in the sample database of pixel b is updated again according to

$$M_b(n_i) = a, \text{ for } i = 1, 2, ..., N \tag{10}$$

The above steps are repeated to process all pixels. The background model can be updated once. The setting of the second sampling factor makes the method reduce the amount of calculation while maintaining the detection performance. The running quality of the method is improved. The pseudo-code of the entire process is shown in Algorithm 1.

III. Experimental Results and Analysis

A. Experimental Setting

The parameters of the video captured using iqoo9 are 720p

and 30fps. Videos are taken at different brightness, locations, and angles for testing. There are videos in the dataset with different camera motion states, different video initialization states, different brightness, different brightness mutation, different relative sizes of objects in the camera frame, different object motion speeds, different background clutter, different shooting angles, different number of moving objects, and different video resolutions.

A variety of different metrics can be used when evaluating the performance of algorithms using a range of background subtraction algorithms. These metrics usually include the following basic quantities: True positive (TP) quantifies the number of foreground pixels that the method has accurately identified. False positive (FP) counts instances where the background pixels were incorrectly classified as belonging to the foreground. True negative (TN) represents the number of background pixels correctly identified by the method. False negative (FN) refers to the number of foreground pixels that are misclassified as background.

The challenge in evaluating these background subtraction methods stems primarily from the absence of a standardized assessment framework. While various authors have put forth frameworks, they often focus primarily on showcasing the merits of their respective methods. Notably, the Percentage of Correct Classification (PCC), stands as the most prevalent metric in computer vision for gauging the effectiveness of binary classifiers. This metric comprehensively incorporates four key values:

$$PCC=\frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

This metric is adopted in our experiments. Note that a high *PCC* percentage represents only a small number of detection errors.

We also used Precision for prediction results in Eq. (12), Recall for original samples in Eq. (13), and F1-Score in Eq. (14), considering both Precision and Recall, and the Peak signal-to-noise ratio (*PSNR*) reflecting image reconstruction quality, as given by

$$Precision=\frac{TP}{TP+FP} \tag{12}$$

$$Recal=\frac{TP}{TP+FN} \tag{13}$$

$$F1=\frac{2\times Precision\times Recall}{Precision\times Recall} \tag{14}$$

where *Precision* means the probability of actually being positive among all the predicted positive samples, and *Recall* represents the probability of actually being positive among the predicted positive samples, and *F1*-score performs the comprehensive consideration of the above two requirements. *PSNR* is defined based on *MSE* (Mean square error), which can be computed by

$$MSE=\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left[I(i,j)-K(i,j)\right]^2 \tag{15}$$

$$PSNR=20\times\log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right) \tag{16}$$

Given an original image I of size $m\times n$ and a noisy image K after adding noise to it, MAX_I is the maximum pixel value of the image.

B. Data processing

In order to make the experimental data variable in the "experiments with different video resolutions" section only have resolution. To ensure accurate representation of how resolution size affects data processing, the experimental data are resized using bilinear interpolation method.

Figure 9 shows the axial interpolation of x and y. The new green point $n(x, y)$ is obtained from any blue point $n(i, j)$. Let the mapping function of the image be n. At any point (i, j), its pixel value is $y = n(i, j)$. Where i ranges from 0 to width; The range of j is between 0 and height. Suppose we have four pixels $n(x_1, y_1)$, $n(x_2, y_1)$, $n(x_1, y_2)$, $n(x_2, y_2)$. As shown in Eqs. (17) and (18), new pixel points $n(x, y_1)$, $n(x, y_2)$ are obtained from these four pixels by linear interpolation in the X-axis direction.

$$f(x,y_1)=\frac{x_2-x}{x_2-x_1}\times f(x_1,y_1)+\frac{x-x_1}{x_2-x_1}\times f(x_2,y_1) \tag{17}$$

$$f(x,y_2)=\frac{x_2-x}{x_2-x_1}\times f(x_1,y_2)+\frac{x-x_1}{x_2-x_1}\times f(x_2,y_2) \tag{18}$$

The new orange point $n(x, y)$ is obtained from the green point $n(i, j)$. As shown in Eq. (19), the new pixel point $n(x, y)$ is obtained from these four pixels by linear interpolation in the X-axis direction.

$$f(x,y)=\frac{y_2-y}{y_2-y_1}\times f(x,y_1)+\frac{y-y_1}{y_2-y_1}\times f(x,y_2) \tag{19}$$

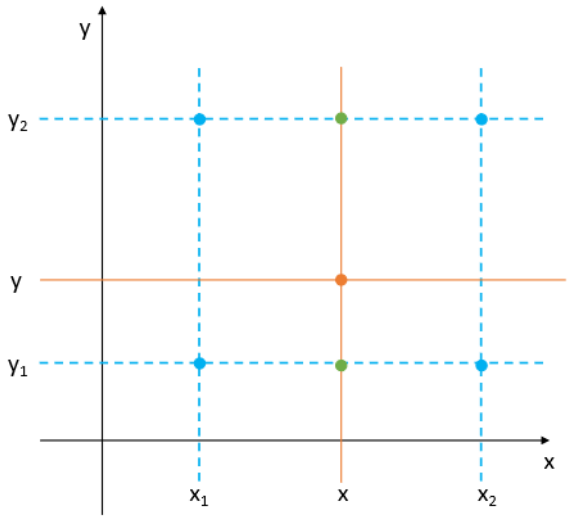

Fig. 9. The x and y axis interpolation.

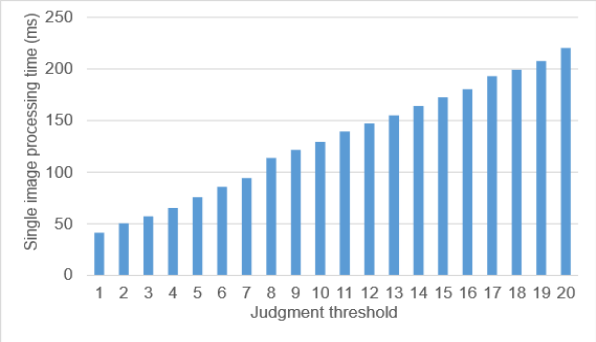

Fig. 10. Single image processing time for m ranging from 1 to 20. The other parameters of Vibe were set to N = 20, R = 20, Ψ = 16 and Video pixel is 1280×720.

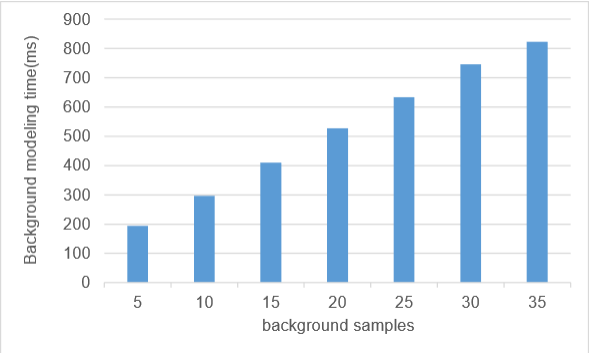

Fig. 11. Background modeling time for N ranging from 1 to 20. The other parameters of Vibe were set to m = 2, R = 20, Ψ = 16 and the video pixel is 1280×720.

Table 2. Single image processing time for different pixel images.

Video frame size	640×360	1280×720	2560×1440
Image processing time (ms)	13.11	50.46	202.25

Table 3. Performance evaluation of different scenarios.

Type	Precision	Recall	F1-Score	PCC (%)	PSNR
a1	0.9748	0.6284	0.7642	91.62	**7.53**
a2	**0.9906**	0.8201	0.8973	96.65	7.48
a3	0.9742	0.7896	0.8722	98.55	5.39
a4	0.9544	**0.8573**	**0.9032**	**98.96**	4.96

The whole image is processed in the above way to obtain a picture of the specified size.

C. Experimental environment

We used a Dell G15 5520 computer as an experimental setup, which is equipped with a 12th generation Intel(R) Core(TM) i7-12700H 2.30GHz processor based on the X64 architecture and runs a 64-bit operating system. Before the start of the experiment, we set the initial parameters as follows: the time sampling factor Ψ is set to 16, the color distance radius R is set to 20, the judgment threshold m is set to 2, and the number of background samples N is set to 20.

D. Experimental results

To verify the effect of parameter Settings on Vibe processing speed, we compared single m processing times from 1 to 20, all other factors being equal. Since m determines the threshold of pixel classification in the step of foreground and background segmentation, every time m increases, all pixels need to be compared with more samples in the sample library. As shown in Fig. 10, we can observe the relationship between the speed of individual image processing time and the m value, and the Vibe processing time for a single image will inevitably increase. Since only N is the parameter that has the greatest influence on the background modeling time, we conducted a comparative experiment on the setting of N in order to pay attention to the background modeling time. The impact of different settings for N can be clearly seen in Fig. 11. The setting of N will definitely drive the background modeling time, which shows that Vibe is sufficiently adjustable to theoretically adjust the parameter settings to meet the needs.

We present a single image processing schedule for images with different pixels in Table 2, where it can be seen that the processing time of a single image is similar to the increasing factor of the phase pixels. This is because Vibe processes images at the pixel level, and all pixels are manipulated in the same way, which means that people need more powerful running equipment to process sharper pictures while maintaining processing speed.

To significantly prove the performance of the proposed algorithm in different situations, experiments are carried out in four different kinds of situations, and the results are summarized in Table 3, in which a1 denotes the detection of moving objects is relatively large and insufficient light, a2 represents the moving objects that are relatively large and well-lit, a3 is the moving object that is relatively small and the light is insufficient, and a4 represents the moving objects that are relatively small and well-lit. The experimental results show that Vibe algorithm has the best detection results in the case of sufficient light, and the precision rate, recall rate, *F1* score and *PCC* can reach 99.06%, 85.73%, 90.32%, and 98.96%, respectively. And the *PCC* can also reach 91.62% in a bad environment. Obviously, the Vibe method has a certain ability to adapt to the environment. We can observe that the PSNR value of smaller moving objects in the image is smaller, which means that the processed image has a lot of differences from the original image. Obviously, we have effectively extracted the foreground of the image.

Then we perform a series of experiments under different conditions.

(1) Comparison results of different camera motion states

Figure 12 shows the comparative experimental results under different camera motion states. Camera shakes and movement will cause a lot of misjudgment of the background. Because when the camera produces motion, the relative image in the image will move. At this point, all moving objects are considered foreground. According to the experiments with different camera motion states, this method is only applicable to the situation of fixed camera. If the camera will inevitably produce jitter, the larger the range of jitter, the greater the impact on the detection results.

(2) Comparison experiments of different video initialization states

Figure 13 shows the comparative experimental results under experiments with different video initialization states. This method needs some time to initialize the background model. The background model is continuously updated during the run of the method. At first, the method needs some time to adapt the background model to the scene. The detection results will produce some misjudgments when the

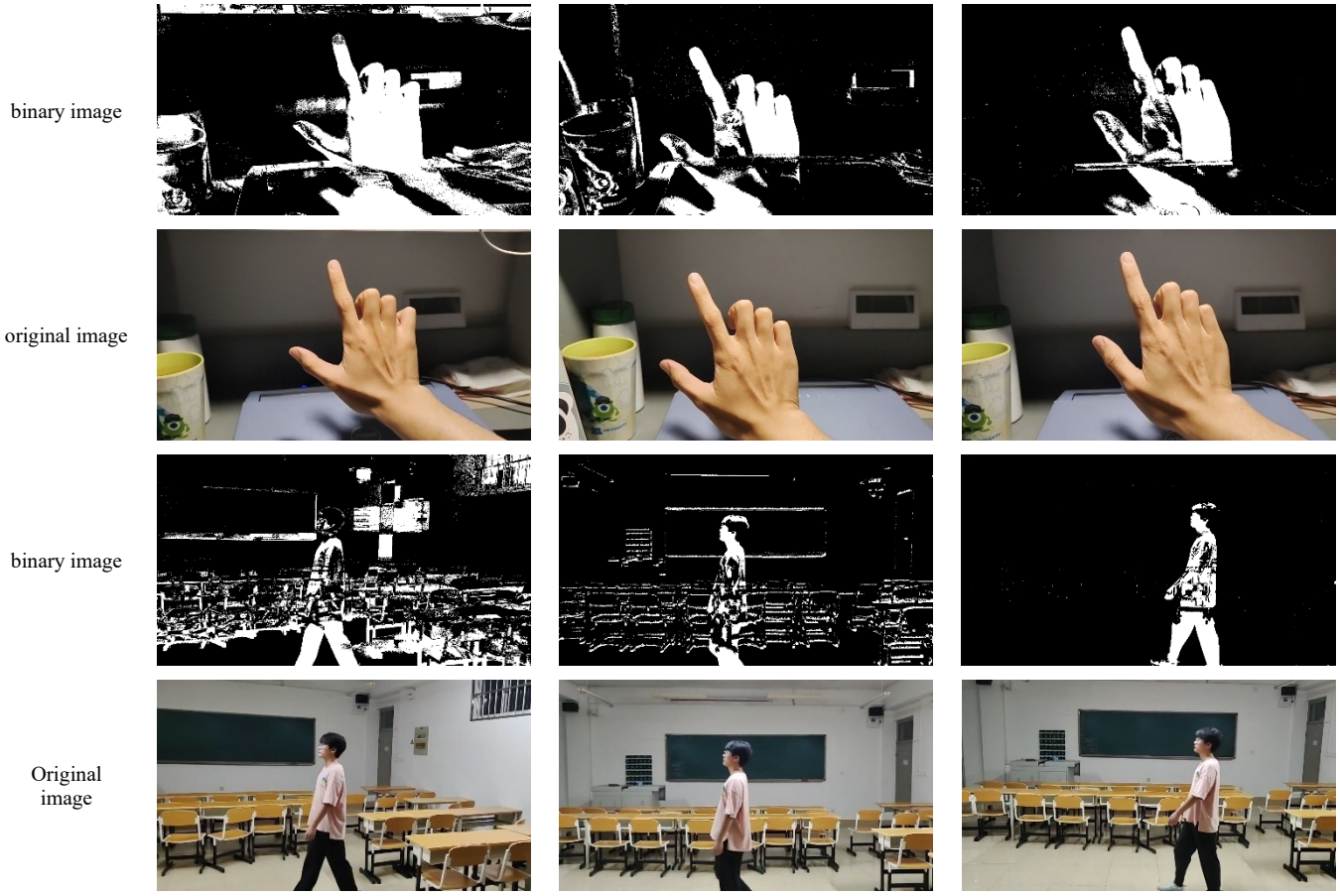

Fig. 12. Comparative experimental results under different camera motion states.

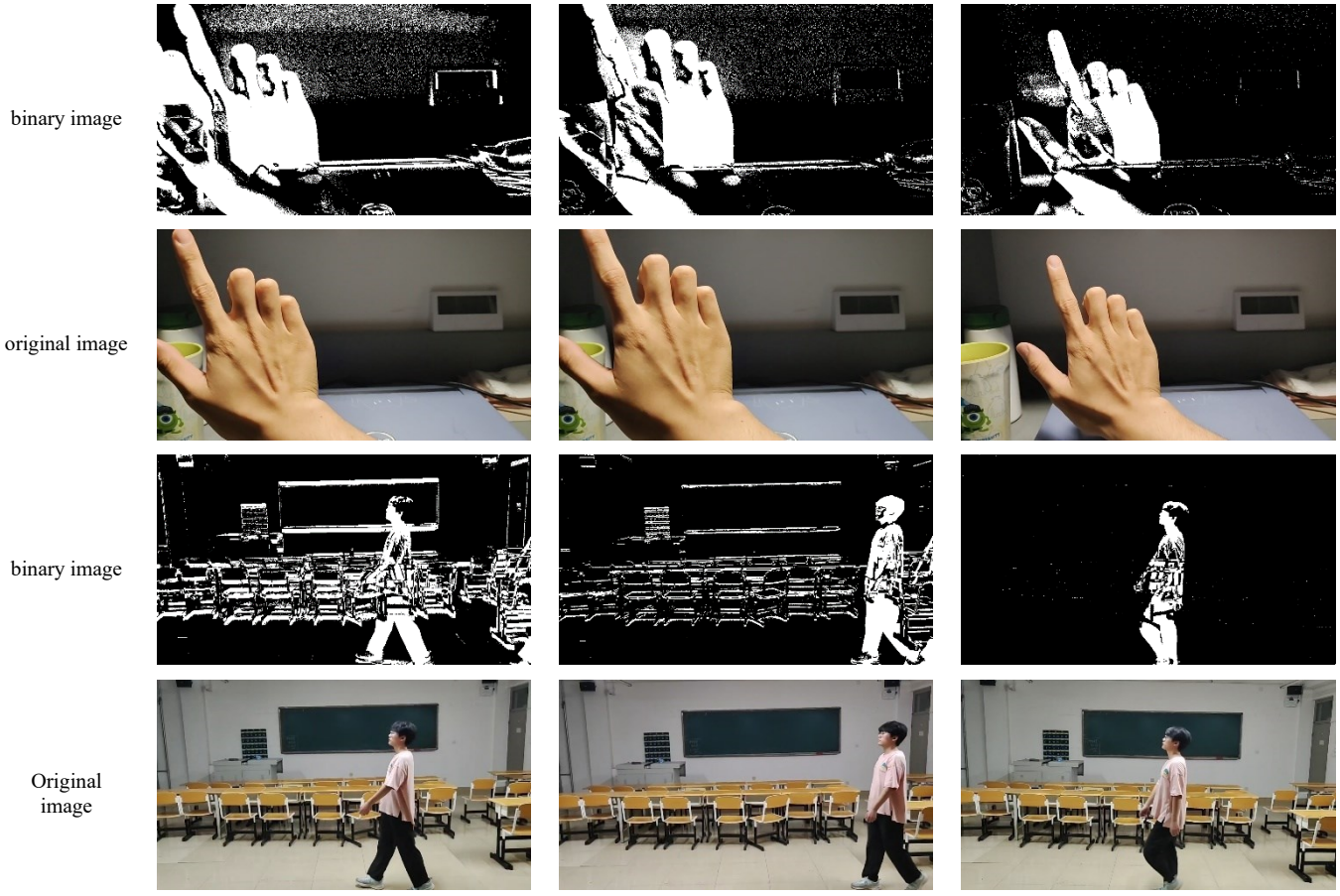

Fig. 13. Comparative experimental results under different camera motion states.

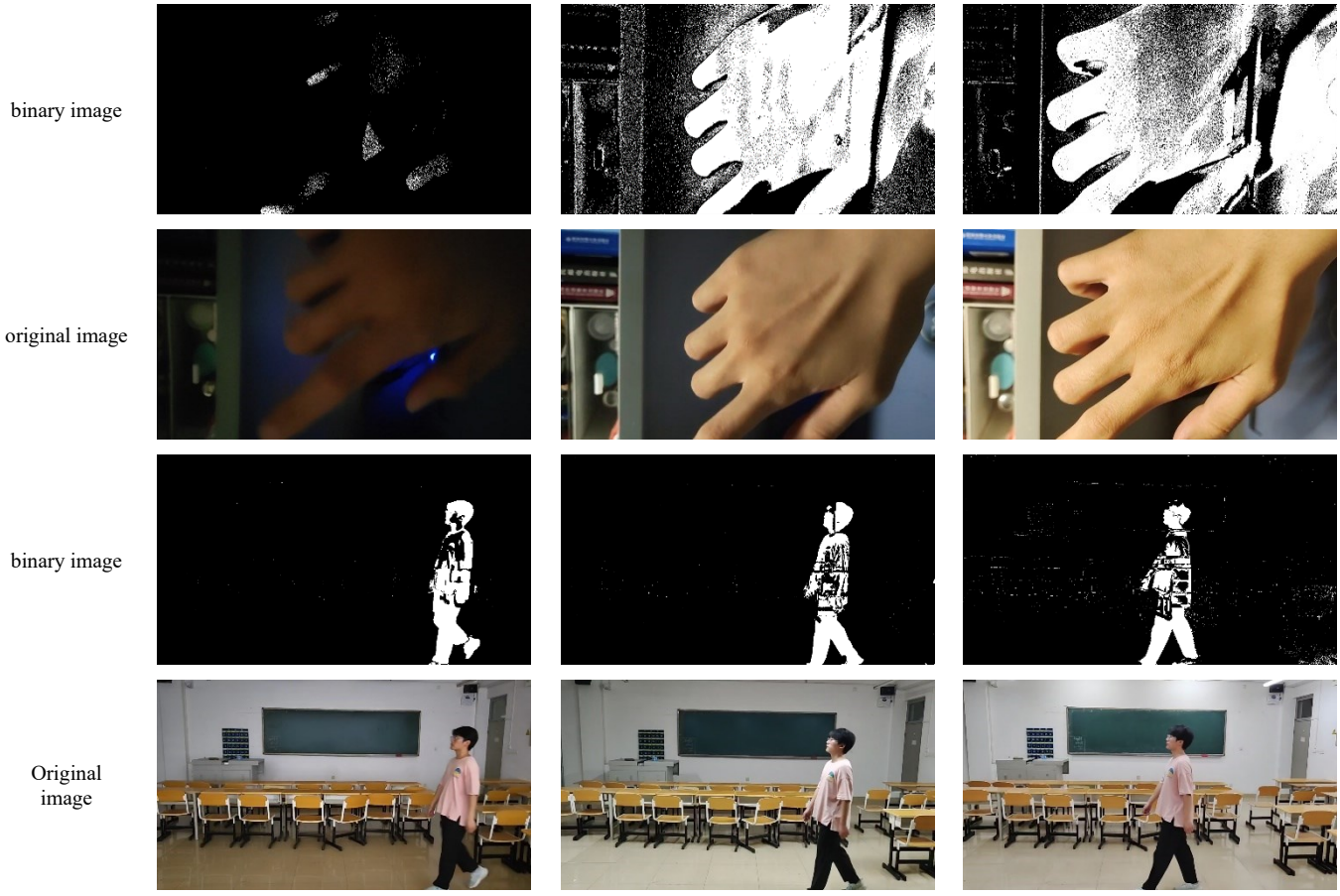

Fig.14. Comparative experimental results under different brightness.

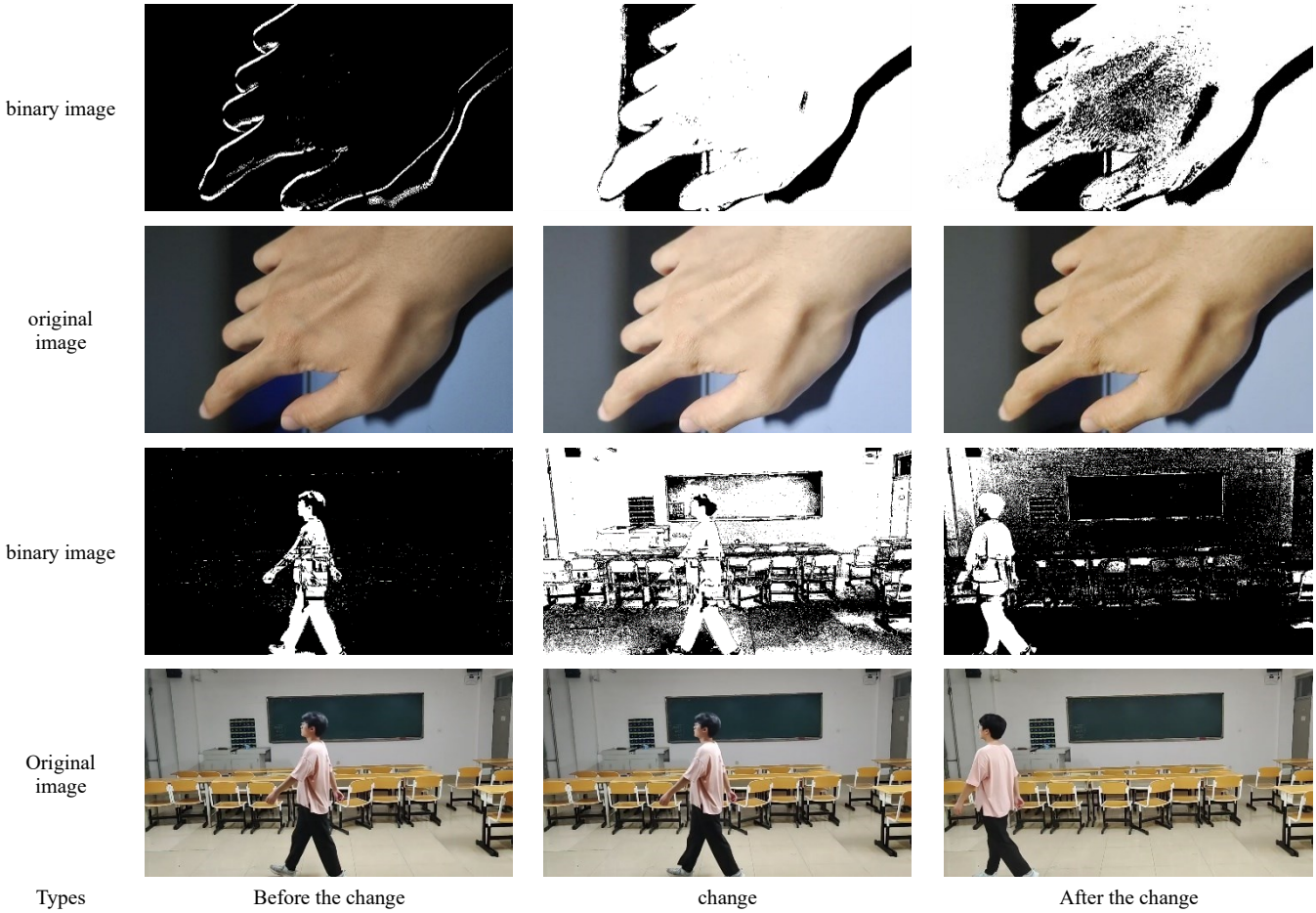

Fig.15. The result of processing the video when the brightness of the scene increases dramatically.

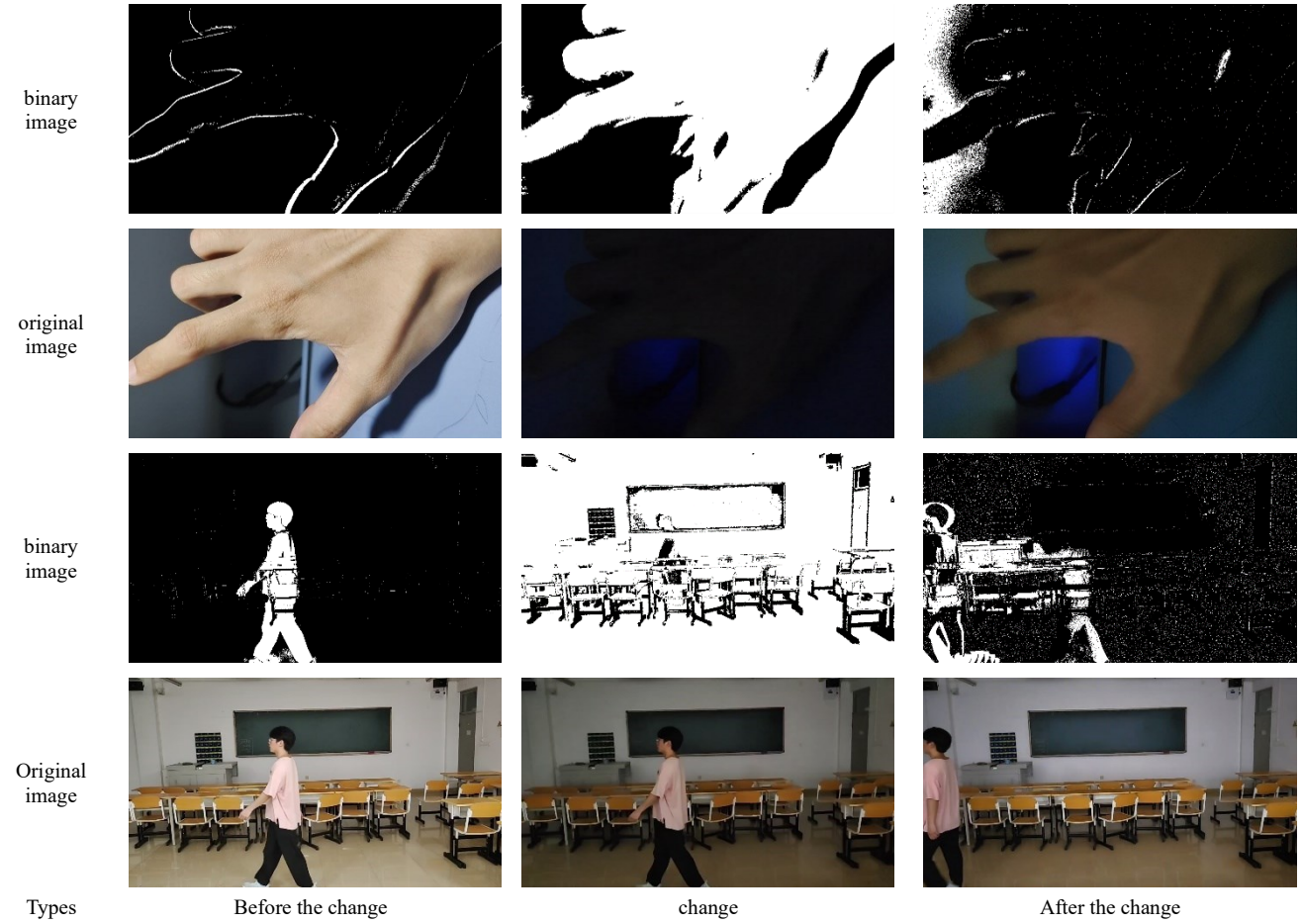

Fig. 16. Comparative experimental results for sudden dimming.

Fig. 17. Comparative experiment: The detection results of moving objects of different relative sizes in the video.

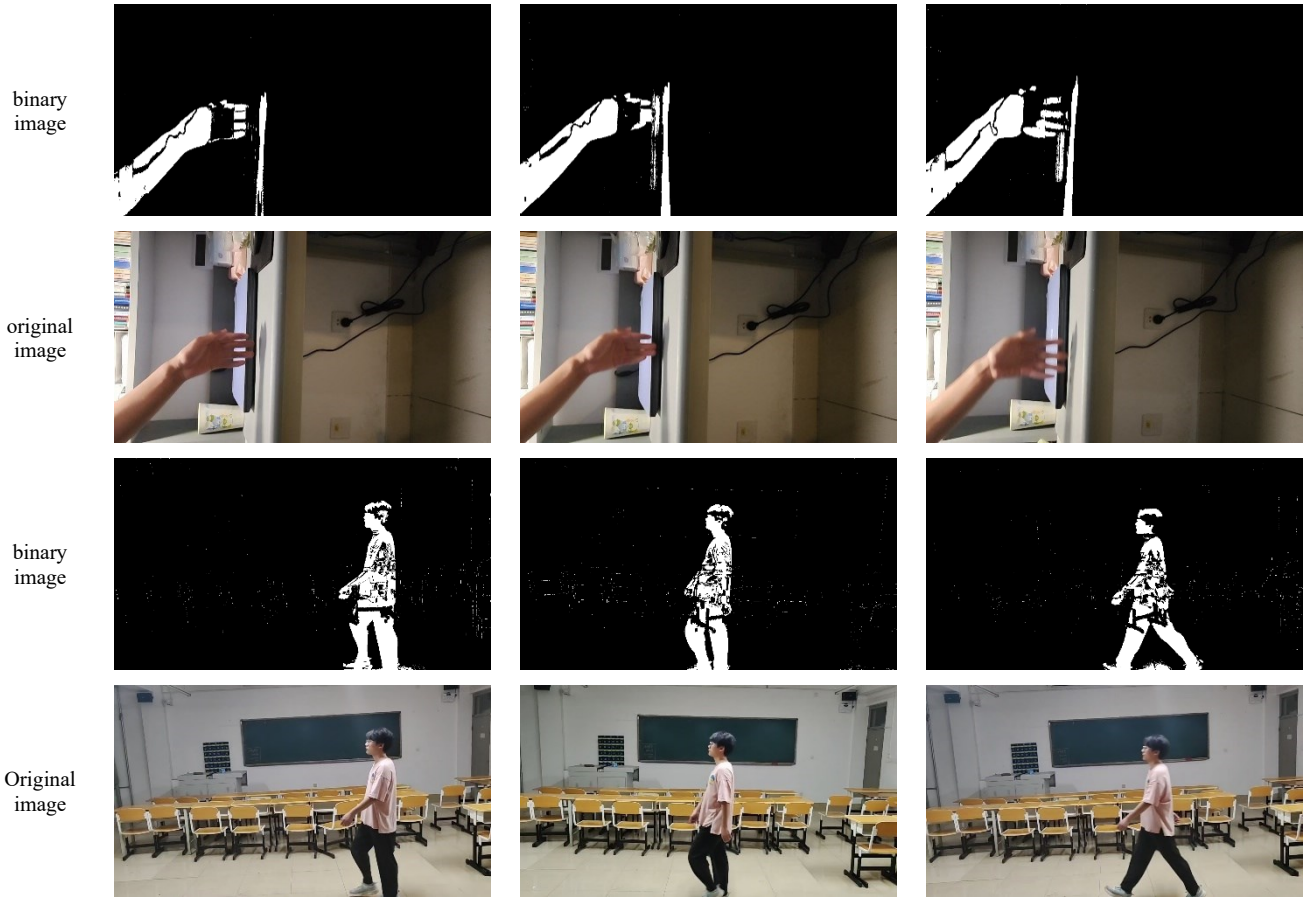

Fig. 18. Comparative experiment: detection results of objects with different moving speeds.

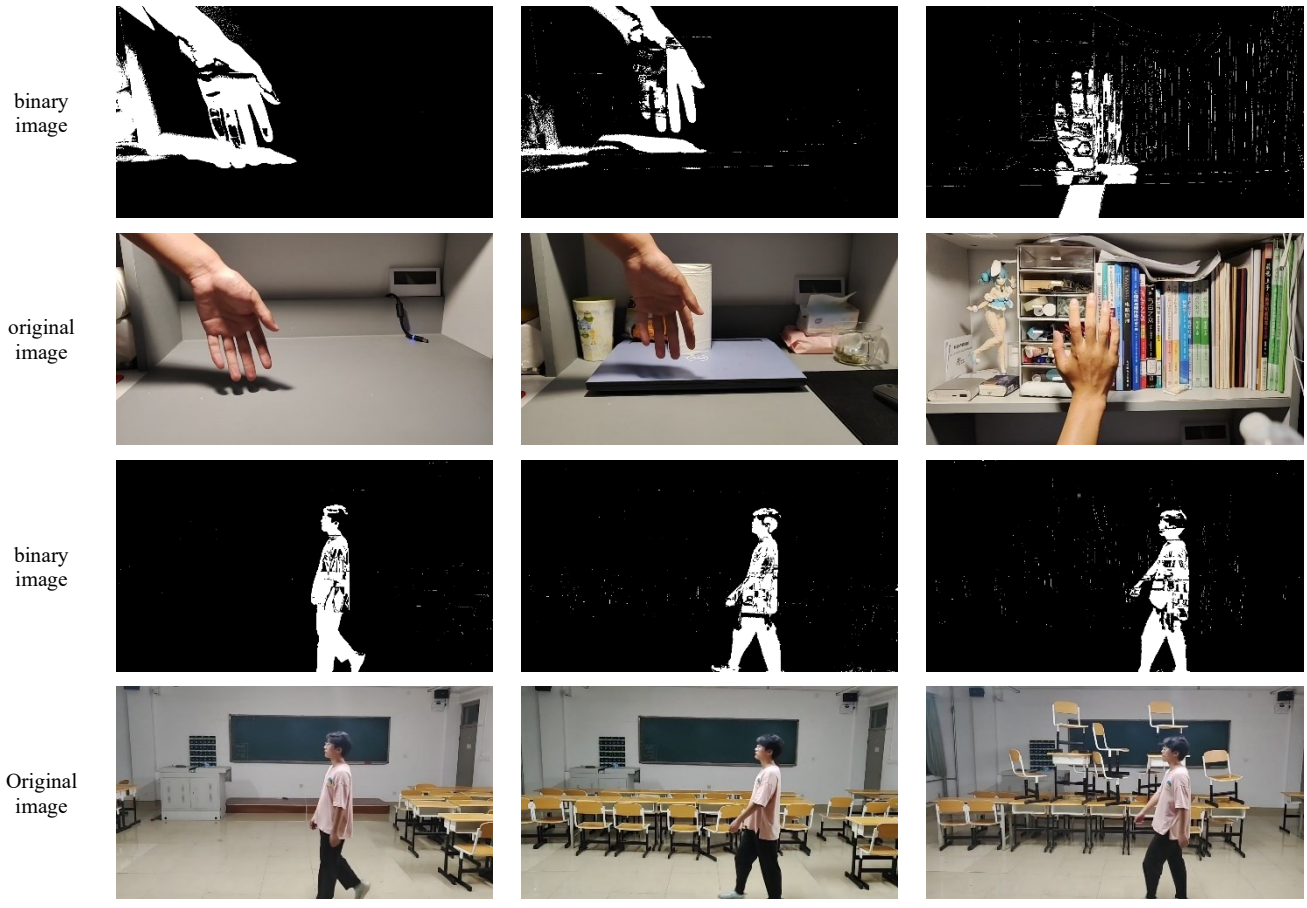

Fig. 19. Comparative experimental results under different background clutter degree.

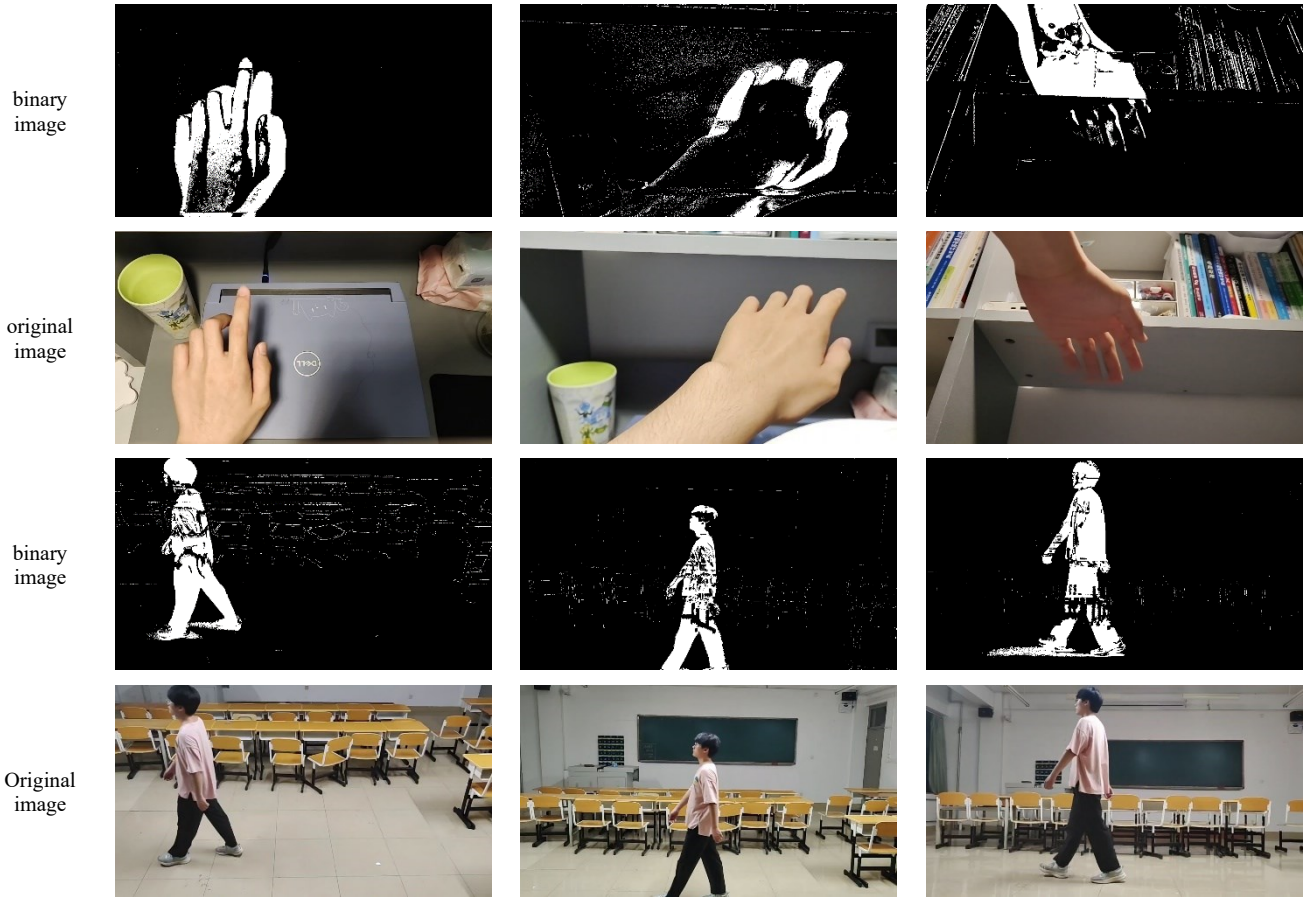

Fig. 20. Comparative experimental results under different shooting angles.

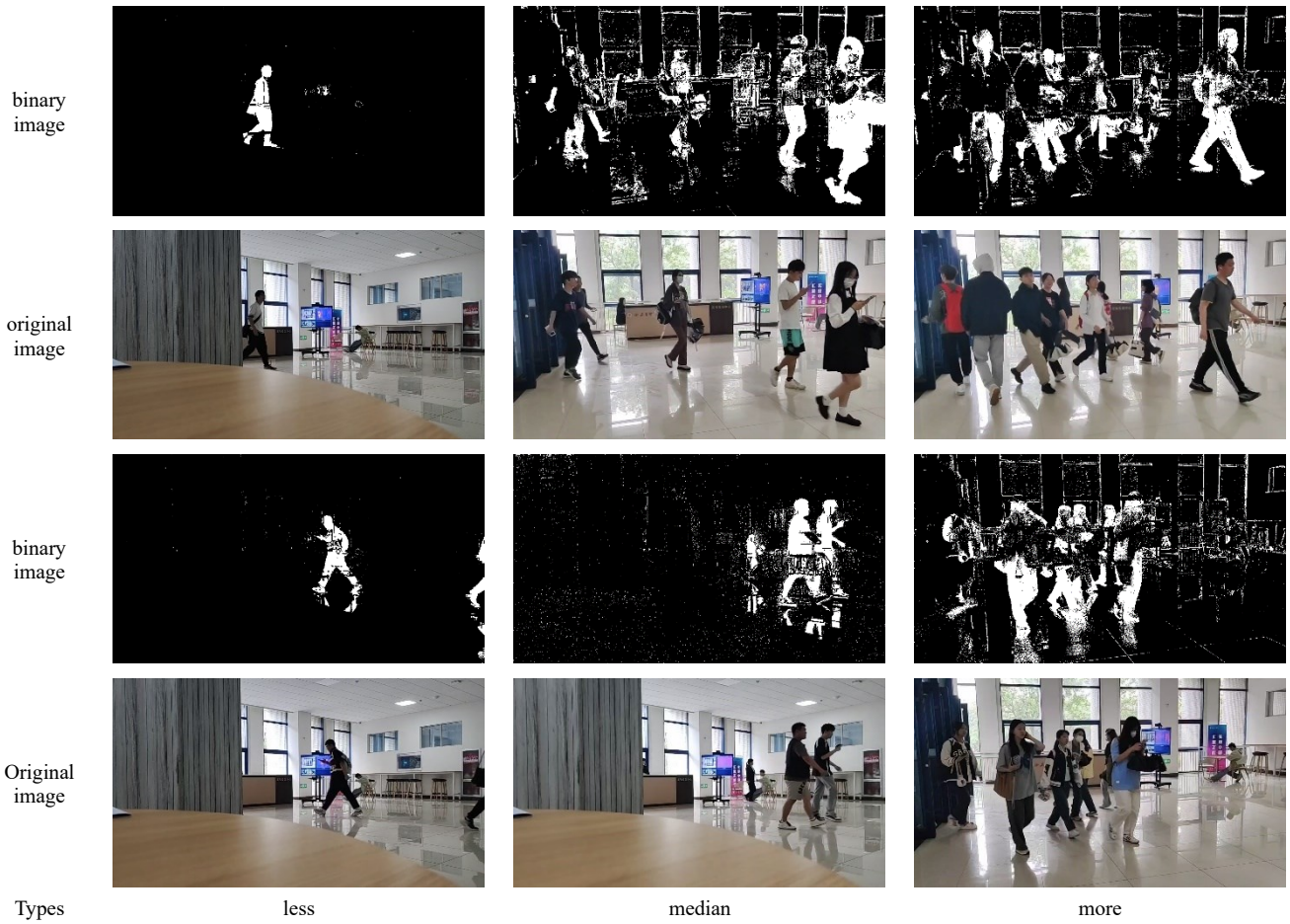

Fig. 21. Comparative experimental results with different number of moving objects.

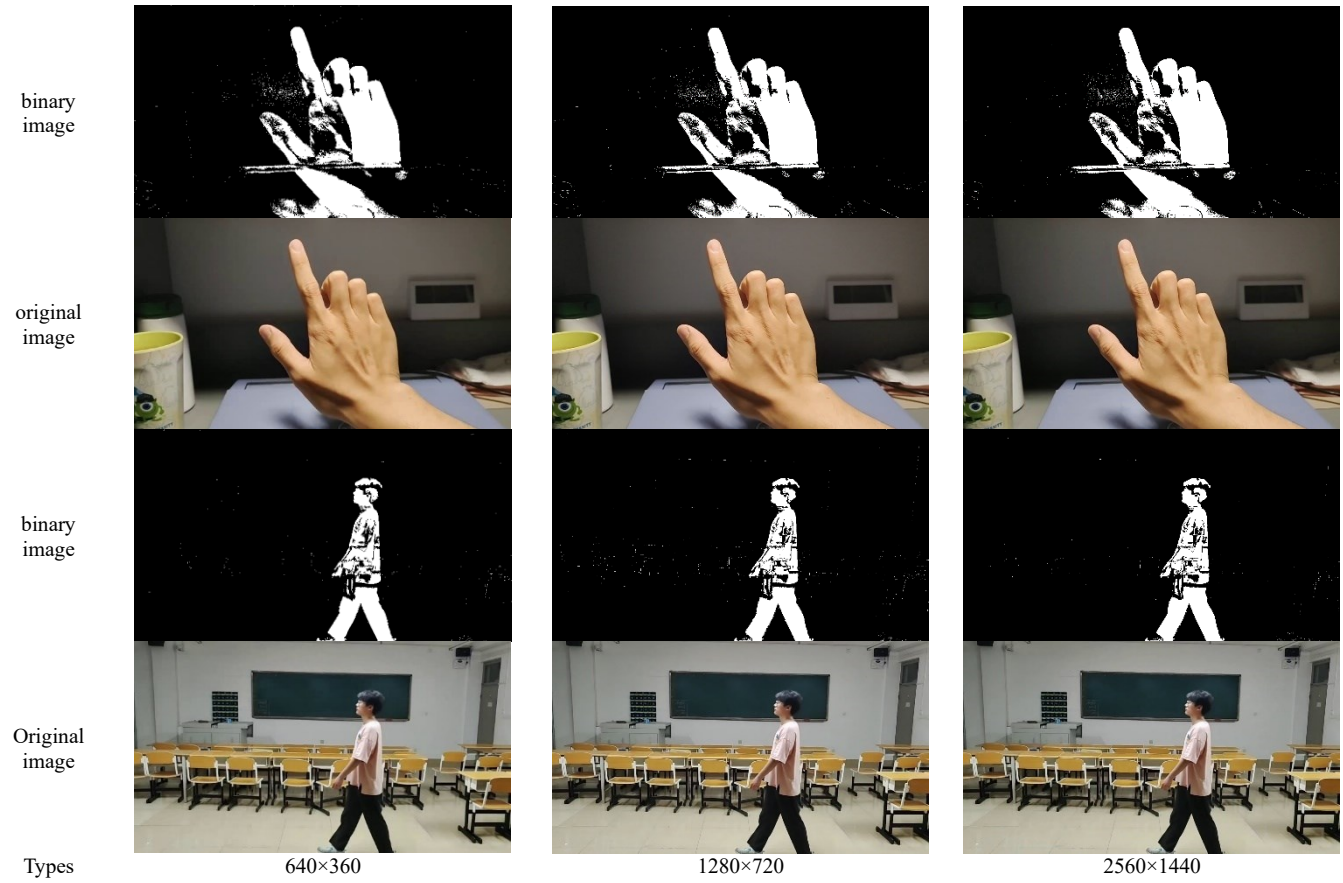

Fig. 22. Comparative experimental results under different video resolutions

scene is not adapted. With the progress of initialization, the sample model update is more perfect, and the background noise and misjudgment are gradually reduced. After 4 to 7 seconds, the method enters the state of normal detection. Without restarting the method, the Vibe method will always maintain normal detection.

(3) Contrast experiment with different brightness

Figure 14 shows the comparative experimental results under different brightness. When the lens is not fixed, the background can be misclassified as the foreground due to the camera shake. Even if the lens is fixed, the background will be misclassified as foreground if it is not initialized. As the light decreases, fewer parts of the foreground object are detected. As the light increases, more details of the foreground object are detected. But there will be backgrounds around the foreground that will be misjudged as foreground. The method has the best detection effect in well-lit places, and the detection effect of the method will be reduced in dim scenes.

(4) Experiments with sudden changes in brightness

Figure 15 shows the sudden brightening and Figure 16 shows the sudden darkening. When there is a sudden change in the brightness of the light, the region with the change in brightness can be misclassified as a moving object (foreground). However, with the continuous update of the background model, the misjudged regions will be gradually corrected. The more severe the brightness change, the more areas will be misclassified. The smaller the change in brightness mutation, the faster the error correction.

(5) Experiments when the relative size of objects in the lens frame is different

Figure 17 shows the experimental results when the relative size of the objects in the lens frame is different. When moving objects (foreground) are presented in the image, some noise will be generated in the detected image. The amount of noise in an object in an image is proportional to the relative size of the image. The smaller the relative size of moving objects in an image, the less noise they contain and the less impact on image quality, so that the contours of these moving objects appear sharper and more precise in the detection results. On the contrary, the larger the relative size, the greater the influence on the surrounding picture, and its own contour will produce a certain deformity. When the moving area in the screen is too large, the flashing white phenomenon will occur.

Table 4. Performance evaluation of different scenarios.

Setting	Precision	Recall	F1-Score	PCC	PSNR
0.05	0.9773	0.8205	0.7967	92.15%	6.22
0.1	0.9811	0.8254	0.7889	93.90%	6.04
0.5	0.9330	0.8117	0.8241	95.81%	6.13
1	0.9311	0.8091	0.8860	96.07%	5.85
1.5	0.9439	0.8223	0.9201	94.20%	5.77
2	0.9461	0.8281	0.9024	93.73%	5.90

(6) Experiments with objects moving at different speeds

Figure 18 shows the experimental results when the object is moving at a speed. The moving speed of the object changes from slow to fast. The interior, edges, level of detail and clarity of the image are very similar. The moving speed of the visible object does not have any effect on the detection effect. No matter the object moves at any speed, the method can capture the moving object without missing detection.

(7) Experiments with different levels of background clutter

Figure 19 illustrates the experiments with different levels. As the background becomes increasingly cluttered, the noise generated at the edges of background objects intensifies, subsequently amplifying its disruptive effect on the foreground object. Moreover, the ghost left by the edge noise of the background object will destroy the architecture. On the contrary, the more concise the background, the less noise is generated in the frame. The foreground objects are also less affected by the background. The results demonstrate that the method will have a relative impact on the detection results in the occasion of cluttered background. The detection effect is best in the scene with relatively simple background.

(8) Experiments with different shooting angles

Figure 20 shows the experimental results with different shooting angles. When the shooting angle is a top-down view, the image is most complete, and the contour and edge details of the moving object can be clearly captured. When looking at head-up, there is a small amount of mutilation and almost only the edge can be captured. Looking up can only capture a small number of moving objects and the objects captured are incomplete. Based on the previous experimental analysis, it is concluded that this phenomenon is due to the lower background complexity when looking down, and the background is less affected by the detection process, so the detection results are more complete. Part of the reason is that well-lit scenes are easier to detect, and the light source is mostly above. The area of the wall, ceiling, or sky in the inspection screen increases when the inspection device looks up, and the area of the ground in the inspection screen increases when the inspection device looks down. And these images are usually simple scenes. The background is mostly complex when looking up, which makes the objects in the detection results of the head-up angle inspection prone to some vulnerabilities.

(9) Experiments when the number of moving objects varies

Figure 21 shows the experimental results obtained with different numbers of moving objects. As long as there is a moving object in the scene, it must be detected. When there are few moving objects, the image can be clearly detected. When there are a lot of moving objects, due to the influence of ghost phenomenon, the more moving objects have a greater impact on detection. Especially when overlap occurs, ghosting has a serious impact on the detection results.

(10) Experiments with different video resolutions

Figure 22 shows the comparative experimental results under different video resolutions. The details of the detection results will be clearer with the increase of the resolution, and the edge of the object will be detected more accurately. The higher the resolution, the more pixels in the image. The method processes the image based on pixels, and the number of pixels plays a decisive role in the degree of detail detected. In Table 4, we report the precision, recall, F1-score, PCC, and PSNR under different settings. The impact of six different modeling intervals on the modeling results has been analyzed. According to the experimental results, it can be seen that the modeling time within a certain range has little effect on the modeling results of Vibe, indicating its good adaptability.

IV. Conclusion

Vibe method belongs to the moving object detection in the continuously developing field of computer vision. Vibe method has pixel-level video processing technology. For faster data processing, the input data is converted to grayscale images. Eight neighborhood random selection method is used to process the original data to fill the sample library. All input images will be compared with the sample pixels in the sample library to obtain the Euclidean distance and threshold. Foreground pixels are extracted to segment all foreground images. Locations judged to be foreground for a long time are forced to update the sample bank. The Vibe method proposed in this paper uses the subsampling factor to update the background model, which is different from the classical method of replacing the oldest value first. Each background pixel and a neighboring pixel have a $1/\Psi$ probability to update the sample bank to adapt to the changes of the scene.

The effectiveness of the Vibe algorithm is affected by a variety of object characteristics and environmental factors, including the relative size of the object, lighting conditions, mutual occlusion at viewing angles, background complexity, and the state of the detection device, the choice of viewing angle, and the resolution setting. It is important to note that although the method exhibits stability in detecting the speed of moving objects, it is extremely sensitive to sharp changes in light conditions, which can lead to significant errors in the detection results. In addition, ghosting in detection videos can create the illusion that objects appear to overlap, and this illusion is more likely to occur in the presence of a complex background. Resolution and brightness are closely related to the effect of detecting object details. The method does not miss detection events and is suitable for occasions that need to capture all dynamics. In the future, we will carry out the following research. The appearance of ghost phenomenon is considered to be reduced. The noise generated by the edge of the background and other noise are studied to be eliminated. These will make the detection results of the foreground image more complete.

References

[1] J. S. Kulchandani, and K. J. Dangarwala, "Moving object detection: Review of recent research trends," IEEE International conference on pervasive computing (ICPC), pp. 1-5, 2015.

[2] D. Fortun, P. Bouthemy, and C. Kervrann. Optical flow modeling and computation: A survey," Computer Vision and Image Understanding, vol. 134, pp. 1-21, 2015.

[3] S. S. Sengar and S. Mukhopadhyay, "Moving object detection based on frame difference and W4," Signal, Image and Video Processing, vol. 11, pp. 1357-1364, 2017.

[4] S. Sanches, A. Sementille, I. Aguilar, V. Freire, "Recommendations for evaluating the performance of background subtraction algorithms for surveillance systems," Multimedia Tools and Applications, vol. 80, pp. 4421-4454, 2021.

[5] Q. Zheng et al., "A real-time transformer discharge pattern recognition method based on CNN-LSTM driven by few-shot learning," Electric Power Systems Research, vol. 219, 109241, 2023.

[6] D. Pierpaolo et al., "Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity," Applied Sciences, vol. 13, no. 13, 7507, 2023.

[7] Q. Zheng et al., "A real-time transformer discharge pattern recognition method based on CNN-LSTM driven by few-shot learning," Electric Power Systems Research, vol. 219, 109241, 2023.

[8] B. Garcia-Garcia, T. Bouwmans, A. Silva, "Background subtraction in real applications: Challenges, current models and future directions," Computer Science Review, vol. 35, 100204, 2020.

[9] Y. Luo, H. Zhou, Q. Tan, X. Chen, and M. Yun, "Key frame extraction of surveillance video based on moving object detection and image similarity," Pattern Recognition and Image Analysis, vol. 28, pp. 225-231, 2018.

[10] L. Ding and A. Goshtasby, "On the Canny edge detector," Pattern Recognition, vol. 34, no. 3, pp. 721-725, 2001.
[11] M. Poongodi, M. Hamdi, and H. Wang, "Image and audio caps: automated captioning of background sounds and images using deep learning," Multimedia Systems, vol. 29, no. 5, pp. 2951-2959, 2023.
[12] D. Sudha and J. Priyadarshini, "An intelligent multiple vehicle detection and tracking using modified Vibe method and deep learning method," Soft Computing, vol. 24, no. 22, pp. 17417-17429, 2020.
[13] S. Saponara, A. Elhanashi, and Q. Zheng, "Recreating fingerprint images by convolutional neural network autoencoder architecture," IEEE Access, vol. 9, pp. 147888-147899, 2021.
[14] K. S. Bhat, M. Saptharishi, and P. Khosla, "Motion detection and segmentation using image mosaics," In IEEE International Conference on Multimedia and Expo. ICME200, vol. 3, pp. 1577-1580, 2000
[15] R. Lienhart, S. Pfeiffer, and W. Effelsberg, "Video abstracting," Communications of the ACM, vol. 40, no. 12, pp. 54-62, 1997.
[16] S. Mahajan et al., "Vibe: A design space for visual belief elicitation in data journalism," Computer Graphics Forum, vol. 41, no. 3, pp. 477-488, 2022.
[17] Q. Zheng et al., "Fine-grained image classification based on the combination of artificial features and deep convolutional activation features," IEEE/CIC International Conference on Communications in China (ICCC), Qingdao, China, 2017.
[18] Q. Zheng et al., "Robust automatic modulation classification using asymmetric trilinear attention net with noisy activation function," Engineering Applications of Artificial Intelligence, vol. 141, 109861, 2025.
[19] S. Saponara, A. Elhanashi, and Q. Zheng, "Developing a real-time social distancing detection system based on YOLOv4-tiny and bird-eye view for COVID-19," Journal of Real-Time Image Processing, vol. 19, no. 3, pp. 551-563, 2022.
[20] W. Zheng, K. Wang, and F. Wang, "A novel background subtraction method based on parallel vision and Bayesian GANs," Neurocomputing, vol. 394, pp. 178-200, 2020.
[21] Q. Zheng and M. Yang. A video stabilization method based on inter-frame image matching score," Global Journal of Computer Science and Technology, vol. 17, no. 1, pp. 35-40, 2017.
[22] M. O. Tezcan, P. Ishwar, J. Konrad, "BSUV-Net 2.0: Spatio-temporal data augmentations for video-agnostic supervised background subtraction," IEEE Access, vol. 9, pp. 53849-53860, 2021.
[23] Q. Zheng et al., "Static hand gesture recognition based on Gaussian mixture model and partial differential equation," IAENG International Journal of Computer Science, vol. 45, no. 4, pp. 569-583,2018.
[24] A. Cioppa, V. Droogenbroeck, and M. Braham, "Real-time semantic background subtraction," IEEE International Conference on Image Processing (ICIP), pp. 3214-3218, 2020.
[25] R. Kalsotra, and S. Arora, "Background subtraction for moving object detection: explorations of recent developments and challenges," The Visual Computer, vol. 38, no. 12, pp. 4151-4178, 2022.
[26] X. Tian, Q. Zheng, and N. Jiang, "An abnormal behavior detection method leveraging multi-modal data fusion and deep mining," IAENG International Journal of Applied Mathematics, vol. 51, no. 1, pp. 92-99, 2021.
[27] S. Sengar, and S. Mukhopadhyay, "Moving object detection using statistical background subtraction in wavelet compressed domain," Multimedia Tools and Applications, vol. 79, no. 9, pp. 5919-5940, 2020.
[28] J. H. Giraldo, and T. Bouwmans, "Semi-supervised background subtraction of unseen videos: Minimization of the total variation of graph signals," IEEE International Conference on Image Processing (ICIP), pp. 3224-3228, 2020.
[29] A. Kushwaha et al., "Dense optical flow based background subtraction technique for object segmentation in moving camera environment," IET Image Processing, vol. 14, no. 14, pp. 3393-3404, 2020.
[30] Q. Zheng et al., "A bilinear multi-scale convolutional neural network for fine-grained object classification," IAENG International Journal of Computer Science, vol. 45, no. 2, pp. 340-352, 2018.
[31] F. Villa-Gonzalez, et al., "SDR-based monostatic chipless RFID reader with vector background subtraction capabilities," IEEE Transactions on Instrumentation and Measurement, vol. 72, 2023.
[32] Q. Zheng et al., "CLMIP: cross-layer manifold invariance based pruning method of deep convolutional neural network for real-time road type recognition," Multidimensional Systems and Signal Processing, vol. 32, no. 1, pp. 239-262, 2021.
[33] N. Jiang et al., "A municipal PM2. 5 forecasting method based on random forest and WRF model," Engineering Letters, vol. 28, no. 2, pp. 312-321, 2020.
[34] J. Shyi, and S. Kim, "HEBGS: Homomorphic encryption-based background subtraction using a fast-converging numerical method," IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-5, 2023.
[35] Q. Zheng et al., "Rethinking the role of activation functions in deep convolutional neural networks for image classification," Engineering Letters, vol. 28, no. 1, pp. 80-92, 2020.
[36] L. Yue et al., "An autoencoder based background subtraction for public surveillance," IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. 104, no. 10, pp. 1445-1449, 2021.
[37] Y. Irawan, R. Wahyuni, and H. Herianto, "Morse code receiver on invisible light using background subtraction method," Journal of Robotics and Control, vol. 2, no. 4, pp. 283-286, 2021.
[38] J. Li et al., "Dynamic hand gesture recognition using multi-direction 3D convolutional neural networks," Engineering Letters, vol. 27, no. 3, pp. 490-500, 2019.
[39] Q. Zheng et al., "Reconstruction error based implicit regularization method and its engineering application to lung cancer diagnosis," Engineering Applications of Artificial Intelligence, vol. 139, 109439, 2025.
[40] Q. Zhang et al., "Segmentation of hand gesture based on dark channel prior in projector-camera system," IEEE/CIC International Conference on Communications in China (ICCC), Qingdao, China, pp. 1-6, 2017.
[41] W. Jia et al., "Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector," IET Image Processing, vol. 15, no. 14, pp. 3623-3637, 2021.
[42] Q. Zheng et al., "Near-infrared image enhancement method in IRFPA based on steerable pyramid," Engineering Letters, vol. 27, no. 2, pp. 352-363, 2019.

Dali Qiao is pursuing his bachelor's degree at the Shandong Management University. His research interests include deep learning, computer vision, and pattern recognition.

Xinyu Tian is a lecture of the Shandong Management University. She received the B.E. degree from Shandong Jiaotong University in 2014. She received the M.E. degree from Shandong University in 2018. Her research interests include signal processing, pattern classification, computer vision, and intelligent computing.

Qinghe Zheng is an associate professor of the Shandong Management University. He received the B.E. degree from Xi'an University of Posts and Telecommunications in 2014. He received the M.E. and Ph.D. degrees from Shandong University in 2018 and 2022, respectively. He has co-authored about 50 peer reviewed scientific journal articles and holds 11 patents. He is a member of IEEE and guest editors of several journals, including Sensors, Symmetry, and Electronics. He is also TPC members of some international academic conferences, including ICAML 2020, ICBTA 2020, BDMIP 2020, IVPAI 2020, ICCAES 2021. His research interests include image processing, pattern recognition, deep learning, and edge computing.

Weiguang Wang is an associate professor of the Shandong Management University. He has presided one special research project on social science planning, one scientific research plan project for higher education institutions, and one philosophy and social science planning project. He participated in two projects of the development plan of safety production science and technology. He has published more than 20 papers, and obtained 4 utility model patents and 5 software copyrights. His research interests include information security and intelligent algorithms.