SN Computer Science (2020) 1:106
https://doi.org/10.1007/s42979-020-0118-5

ORIGINAL RESEARCH

A Systematic Algorithm for Moving Object Detection with Application in Real-Time Surveillance

Beibei Cui[1] · Jean-Charles Créput[1]

Received: 16 August 2019 / Accepted: 11 March 2020 / Published online: 28 March 2020

Abstract
Moving object detection and tracking from video sequences are a relevant research field since it can be used in many applications. While detection allows to return object shapes discovered in the image, tracking aims to individually identify and estimate individual trajectories of detected objects over time. Hence, detection can have a crucial impact on the overall tracking process. This paper focuses on detection. Currently, one of the leading detection algorithms includes frame difference method (FD), background subtraction method (BS), and optical flow method. Here, we present a detection algorithm based on the first two approaches since it is very adequate for fast real-time treatments, whereas optical flow has higher computation cost due to a dense estimation. A combination of FD and BS with Laplace filters and edge detectors is a way to achieve sparse detection fast. Thus, a main proposed contribution is the achievement of a systematic detection algorithm for moving target detection with a more elaborated combination of basic procedures used in real-time surveillance. Experimental results show that the proposed method has higher detection accuracy and better noise suppression than the current methods for standard benchmark datasets.

Keywords Frame difference · Background subtraction · Graphical user interface · Real-time surveillance

Introduction

Moving object detection [1–3] is an image processing process used to extract moving objects in a sequence of images, usually based on image features such as edges, colors, and textures. For real-time intelligent surveillance [4, 5], automated vehicle detection and tracking, personnel tracking, and many other applications, it is undoubtedly an indispensable area of research, not only in 2D motion observed but also in 3D scenes [6]. Globally, the objective of multi-target detection is to jointly estimate, at each observation time, the number of targets and their trajectories from noisy sensor measurements. According to the recent review [1], multiple object detection methods could be classified roughly within two classes of detection-based tracking (DBT) and detection-free tracking (DFT). The former includes a detection step of the objects prior to estimate their trajectories. The latter focuses on the tracking process exclusively, given a predefined initialization.

It is worth noting that DBT allows objects appear and disappear and has more general application, whereas its behavior mainly depends on the quality of the detection procedure, that provides observations for the detection operations as trajectory computation. In this paper, we will put the emphasis on the detection phase only. As it is the case in different detection methods [7, 8], we will also focus on the shape and contour quality of the detected objects in images. Globally, some of the most popular methods for shape detection are optical flow method [9], background subtraction (BS) method [10, 11], and frame differential (FD) method [12]. They can be considered as the most simple and straightforward methods generally used for real-time detection. Actually, convolutional neural networks (CNN) and fully convolutional networks (FCN) become more and more competitive [13], with a large scope of application, but they necessitate supervised learning using a huge amount of ground truth information to learn the network.

Optical flow method [9] estimates the displacement field between two images, so it not only needs to locate the position of each pixel accurately but also needs to find the

✉ Beibei Cui
beibei.cui@utbm.fr

[1] CIAD, Univ. Bourgogne Franche-Comté, UTBM, 90010 Belfort, France

corresponding points between two input images [14]. That is to say, optical flow method has a relatively high computational complexity. Therefore, it spends more time than other methods, so it is more complicated since it computes a dense optical flow field. As a widely used target extraction technique, the background subtraction method can extract objects with a relatively simple algorithm. Although it is relatively easy to implement, it is sensitive to the changes of the light [15]. The frame difference method is still one of the fundamental techniques in computer vision. Frame difference method has the advantage of a small amount of calculation, but it is sensitive to the noise [16], and sometimes, it seems to appear the empty phenomenon that consists of some small apertures and gaps, so its results are not accurate enough. Although there are numerous difficulties with frame difference and background subtraction, these problems are under addressing by some improved methods in recent pieces of the literature on the field.

In 2010, a new inter-frame difference algorithm combined with background subtraction for object detection and tracking was put forward by Weng et al. [17]. This algorithm not only has a low time cost but also has stronger validity and more extensive flexibility. In 2013, Gang et al. [18] propose an algorithm based on the traditional three-frame differential method combined with the Canny edge detection algorithm. In 2014, Liu et al. [19] demonstrated an approach combining background subtraction and three-frame difference to apply to underwater robots to execute underwater missions and detect a moving object by using underwater video, without being affected by the change of lighting condition and the sensitive scenes. In 2017, Wang et al. [13] proposed siamese FCNs to segment the road region elaborately for road detection. This algorithm can detect more accurate road regions than other traditional methods, and the use of location prior can promote the detection performance effectively. In 2019, Yuan et al. [20] presented an end-to-end deep learning method for traffic sign detection in complex environments. The algorithm not only utilizes the densely connected deconvolution layer and frequency hopping connection but also proposes a vertical spatial sequence attention module to obtain more context information to achieve better detection performance.

Although there are many works on detection and tracking, there seems to be no systematic way to appear today. The overall problem remains an open field with methods having their own qualities and limits. With all of this in mind, we restrict our attention to the detection phase, and to some of the most straightforward approaches with real-time applicability, that are, background subtraction and frame difference-based methods. We propose an improved algorithm that combines many of the standard tools encountered in this setting. The approach mainly combines frame difference, background subtraction, Laplace filter, and Canny edge detector (called 3FDBS-LC). It is expected that the improved algorithm can clear the margin of a moving object and fill in blank apertures through a series of mathematical morphology operations. The new combination introduces a fusion of information from BS and FD processes and executes the FD based on three frames instead of two as usual. As presented in experiments, this new combination outperforms BS or DF separate implementations while preserving potential for real-time execution.

The rest of this paper is organized as follows: "Basic Filters and Definitions" gives detailed explanations of preprocessing and post-processing treatments. In "3FDBS-LC Object Detection Algorithm," the methodology and procedures for the main approach that we proposed are described. The experimental results are given in "Experiment and Evaluation." In "Application to Real-Time Video Processing," the proposed method is applied to actual scenes with video rate processing. Finally, conclusions are presented and suggestions are made for further research.

Basic Filters and Definitions

Preprocessing and Post-processing

Image preprocessing and post-processing play an essential role in this research. As we consider the detection phase of objects only, given a sequence of input images, the output of detection is represented as a binary image from which individual connected components directly represent detected objects. This information is the basis for further detection operations, and its quality should impact the rest of the detection operations. As a result, this binary output must reflect the object shapes with the most fidelity, delimiting contours and adequately filling object interiors. Figure 1 shows different visual representations of a detection method: rectangle box, silhouette, and contour. The binary image serves as a result for ground truth evaluation and comparison of different methods, in qualitative and quantitative ways, as presented in this paper, to compare the quality of the obtained shape.

Basic processing operations such as color conversion, image binarization, filtering processing, and edge detection are current basic operations in object detection. Most of these basic tools have straightforward fast implementations and are generally compatible with a real-time context of application. Most of these filters have $O(N)$ time complexity, with N the number of pixels. Also, their parallel implementation in graphic processing unit (GPU) system is now a matter of current fact. We detail the standard processing methods to be combined in the proposed object detection algorithm.

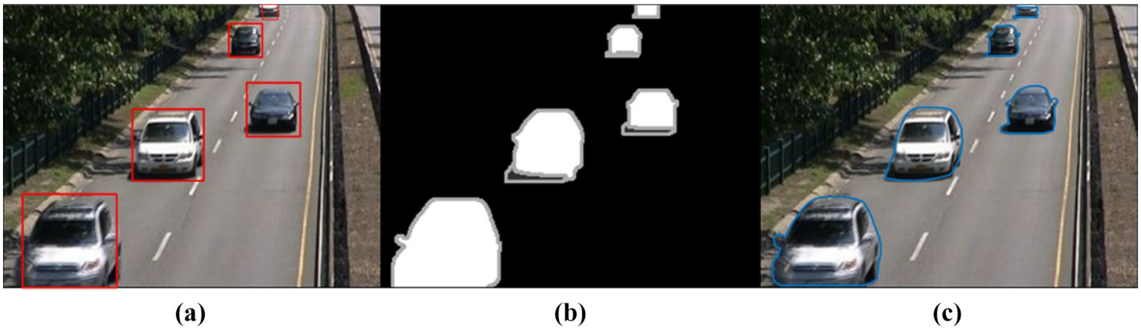


Fig. 1 **a** Rectangle box representation method, **b** silhouette representation method, **c** contour representation method

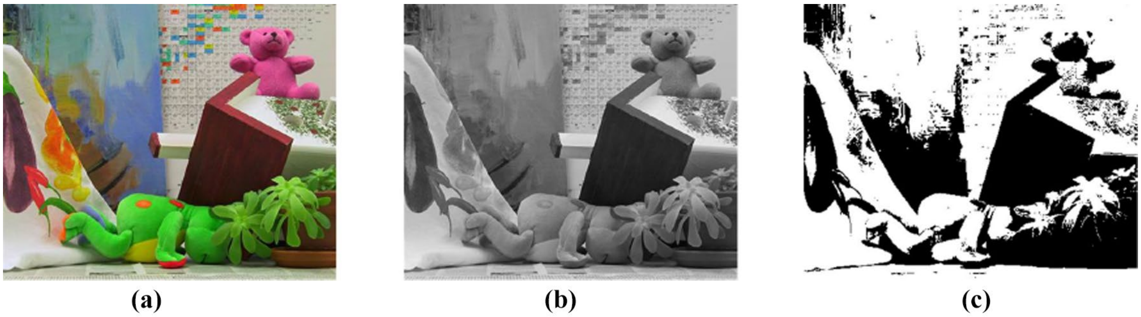


Fig. 2 From the left to the right: **a** original image, **b** grayscale image, and **c** binary image

Color to Grayscale Conversion

RGB comes from the abbreviation of three primary colors red, green, and blue. It is a model in which these three colors are added together in various ways to reproduce a broad array of colors under different weights. On the other hand, the grayscale image is one in which the value of each pixel is a sample representing a kind of light, that is to say, it carries only intensity information, varying from black to white. Since color scale images typically carry much information, when dealing with image, computer needs to read all of it is data information, which will consume more computing time, so it is not conducive to image processing and calculation. Under this situation, it is necessary to convert the color scale image to a grayscale image to increase computational efficiency.

Binarization of Image

Binary images are typically quantized to consist of two possible intensity values, usually 0 and 1, respectively, representing black and white. It is derived from the threshold division of the grayscale image: These pixels with a gray level above the specific threshold are set to 1, and the remaining pixels are set to 0. It means that it will produce an image with a white object on a black background (or vice versa, depending on the specific threshold values), usually used to separate a foreground image from the background image. The grayscale images have a grayscale value ranging from 0 to 255, where 0 is black and 255 is white, while the black and white image has only 0 and 1 values where 0 represents black, 1 represents white. The purpose of image binarization is to speed up the logical decision process when merging information. So binary images can improve recognition efficiency when performing computer recognition. Figure 2 displays the original image and its corresponding grayscale image and binary image.

Filtering Process

Filter processing is the design and realization of a rejector that satisfies the requirements of image processing. Among different kinds of filters, the most commonly used are mean filter, median filter, Gaussian filter, and Laplace filter.

Mean filter is a common linear smoothing algorithm in image processing and noise reduction. The principle of

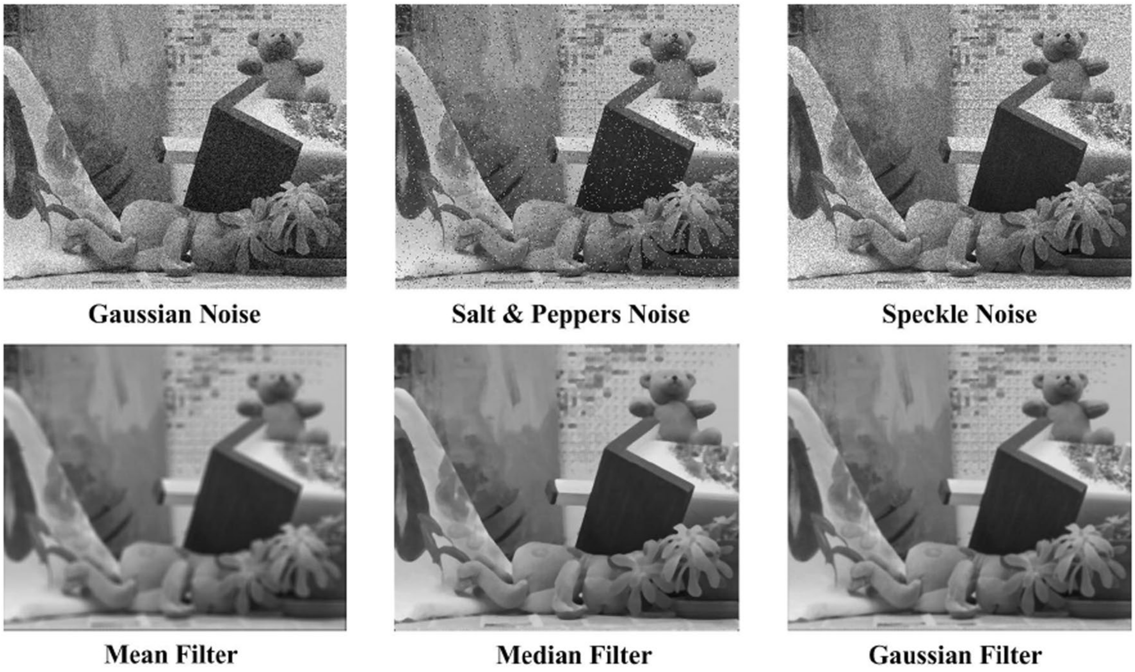


Fig. 3 The first row presents a grayscale image disturbed by Gaussian noise, salt and peppers noise, and speckle noise, respectively; the second row presents the salt and peppers noise image through different filters: Gaussian filter, median filter, and mean filter

mean filtering is simply like a spatial window sliding filter, which replaces the center value with the average value of all the neighbors' pixel values in the window. The window is usually squared to diminish the point where the pixel value varies significantly between pixels and pixels due to noise. Instead of using the mean value to replace all of the surrounding pixels, the Median filter replaces them with median values. Median filters can reduce not only noise but also protect the edge and other detail information of images. Since the median filter obtains a median value, but without considering the unrepresentative of the surrounding pixels, the median filter is more robust for preserving sharp edges. The calculation formula of the mean filter and median filter is defined as below:

$$\mathrm{Mean}(x, y) = \sum M(x, y)/n, \tag{1}$$

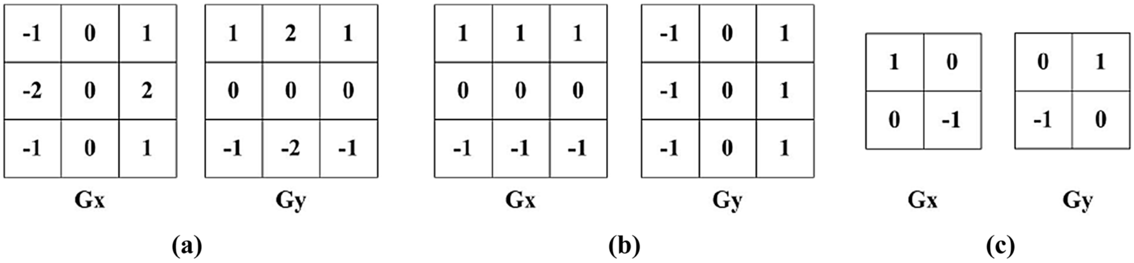
$$\mathrm{Median}(x, y) = \mathrm{med}(M_1, M_2, \ldots, M_n), \tag{2}$$

where n is the number of pixels and M is the value of each pixel. Gaussian filter is considered as an ideal time-domain filter whose impulse response is a Gaussian function. The effect of Gaussian smoothing is to blur the image like the mean filter. The Gaussian standard deviation determines the degree of smoothness. The higher standard deviation is, the larger convolution kernels will be. Gaussian outputs a weighted average of the neighborhood of each pixel, with the average weighting being more toward the value of the center pixel. This is in contrast to the uniform weighted average of the mean filter. Because of this, Gaussian provides milder smoothness and retains edges better than the mean filter of the same size. Gaussian operator is defined as

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{3}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution. Figure 3 shows a grayscale image disturbed by Gaussian noise, salt and Peppers noise, and Speckle noise, respectively. Among them, we focus on the middle one to show how to remove salt and pepper noise from an image using the mean filter, median filter, and Gaussian filter. With mean filtering, even though the noise interference can be eliminated, it is not as good as a median filter in preserving edge information. Compared with these two filters, however, the Gaussian filter is better able to remove noise without improving the sharpness of the image.

Laplacian is the second-order derivative of the Gaussian equation. Compared with the first-order differential, the second-order differential has stronger edge localization capability and a better sharpening effect. Unlike a Gaussian filter that can blur an image, the effect of image sharpening is to enhance the gray contrast and make the blurred



$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

(a)

$$G_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

(b)

$$G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

(c)

Fig. 4 Horizontal and vertical convolution kernels of **a** Sobel operator, **b** Prewitt operator, and **c** Roberts operator

image clearer. Because Laplacian is a differential operator, its application can enhance the region of grayscale mutation in the image and weaken the slowly changing region of the grayscale. Therefore, the sharpening process may choose Laplacian to process the original image to generate an image that describes the abrupt grayscale change. Finally, the Laplacian image is superimposed with the original image to produce a sharpened image. The primary method of Laplacian sharpening can be expressed as:

$$\frac{\partial}{\partial x} G_\sigma(x,y) = \frac{\partial}{\partial x} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4}$$

$$\frac{\partial^2}{\partial x^2} G_\sigma(x,y) = \frac{x^2}{\sigma^4} e^{-\frac{x^2+y^2}{2\sigma^2}} - \frac{1}{\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{5}$$

$$\nabla^2 G_\sigma(x,y) = \frac{\partial^2 G_\sigma(x,y)}{\partial x^2} + \frac{\partial^2 G_\sigma(x,y)}{\partial y^2}, \tag{6}$$

where x, y are the pixel coordinates and σ is the standard deviation of the Gaussian distribution. One proposal in this paper is to integrate the Laplacian filter into the combined BS/FD detection method. Laplacian filter will be adopted, which not only produces sharpening effects but also preserves background information. The gray value in the image can be preserved, and more details are highlighted.

Edge Detection

Edge detection is an image processing technique used to find the boundaries of objects within an image. There are many different types of edge detection operations. Commonly used edge detection algorithms include the Sobel, Prewitt, Roberts, and Canny methods.

Sobel operator formed by a pair of 3×3 convolution kernels, one of the kernels is generated from 90° rotation of another. It is used on 2D spatial gradient measurement to calculate the approximation of the gradient function for image intensity equation, acquiring the high spatial frequency domain of the corresponding edge. Prewitt operator has a very similar derivate mask as a Sobel operator, and it is formed by a pair of 3×3 convolution kernels. Prewitt operator can also be called as derivative operators or derivative masks. It is based on the convolution of the image with a small separable and integer-valued filter in the horizontal and vertical directions. These two operators can be used in vertical direction and horizontal direction. Nevertheless, the coefficient of the derivate mask of the Sobel operator can be adjusted flexibly according to algorithm requirements. Roberts's operator is fast and easy to implement. The operator is formed by a pair of 2×2 convolution kernels. The principle of Roberts operator is achieved by computing the sum of the squares of the neighbor pixels to approximate the gradient value of an image. Figure 4 shows these three kinds of operators' horizontal and vertical convolution kernels.

Robert operator can locate the target accurately, but it is less sensitive to noise because it is not smooth. The Prewitt operator and the Sobel operator belong to the first-order differential operator, the former is the averaging filter, and the latter is the weighted averaging filter. They are good at detecting grayscale in low noise images, but they do not perform well with images under complex noise. Canny edge operator is more accurate than Sobel, Prewitt, and Roberts operators. From Fig. 5, we can see that the Canny edge detector can more completely discover the edge information of the image, so it performs better than other operators. In this work, the Canny edge detector is adopted.

Morphological Transform

Morphology processing is an operation which displays a specific structural element in an input image and generates the desired output image. The function of morphological processing is to eliminate interferences, fill small apertures, and smooth boundary. The most fundamental morphological operators are erosion and dilation.

Erosion and Dilation

Erosion is an operation by moving the structural element S with a fixed center point and repeating this step to find all the points satisfying the condition in the set of objects X.

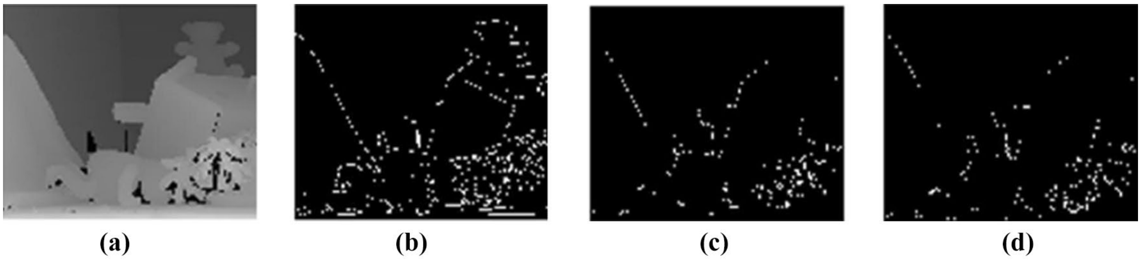


Fig. 5 **a** Original image and its corresponding processed image, **b** Canny operator, **c** Prewitt operator, and **d** Roberts operator

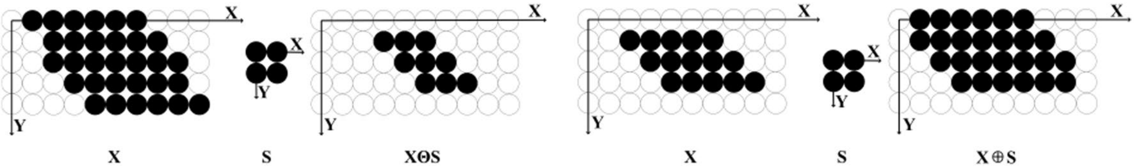


Fig. 6 Process of erosion and expansion operations

The process of erosion is to compare the origin point of the structural element S with the point of the object X; if all the points on S can be found their corresponding points on X, then these corresponding points on object X will be retained; otherwise, these points will be removed. Erosion operation can eliminate holes or noise, but also reduce the size of the affected area. Different sizes of structural elements can bring about different effects on denoising. The larger the structural element is, the more noticeable the change will be. As shown in Fig. 6, a 3 × 3 structural element is adopted to participate in erosion operation.

The process of dilation is similar to erosion operation, and it's a comparison between the points of structural element S and the points on the object X with moving structural element. If there is at least one point be overlapped with the point on X, this pixel will be recorded; if there is no coincidence point for all the elements, it indicates that there is no pixel corresponding to the structural element S and the object X. Results of dilation will also be influenced both by the size and by shape of a structural element. Dilation operation can fill some of the gaps in the moving target area; it also can inflate the edge pixels of a moving object.

Opening and Closing

Opening operation is defined as carrying out the erosion operation first and then performing dilation operation by using the same structural element. It can be expressed by the following formula:

$$Dst1 = open(src, elem) = dilate(erode(src, elem)), \tag{7}$$

where $Dst1$ represents the result of the final operation, scr stands for the object X and $elem$ denotes the structural element S. Opening operations can eliminate tiny objects, separate the objects at subtle joints, and smooth the boundaries of large objects, but without significantly altering the area of the object. On the contrary, the following closing operation

$$Dst2 = close(src, elem) = erode(dilate(src, elem)) \tag{8}$$

is defined as the dilation operation followed by the erosion operation. The closing operation can fill some of the small gaps in the moving target area, connect the objects closer to each other, smooth the boundary of the target, and keep the size of target unchanged.

Morphology Post-processing

Much morphological processing is expressed as a combination of erosion and dilation. In this algorithm, the closing operation will be employed; the mathematics morphology formula is:

$$close(A, B) = erode(dilate(A, B), B) = (A \oplus B) \ominus B. \tag{9}$$

where $\oplus$ is the dilation operator, $\ominus$ is the erosion operator, A is the image, B is a structural element, specified as 3 × 3 matrix. Therefore, after performing such closing morphological processing on the binary image, the small apertures are filled, and the small gaps are connected.

3FDBS-LC Object Detection Algorithm

After image conversion and Laplace filter processing, the most critical parts we propose now are the frame difference method, the background subtraction method, and a combination with edge detection. Standard approaches and our proposed new combination are presented in this section. Considering the disadvantages of frame difference and background subtraction, which are easy to disturb by the sensitivity of noise and brightness, adding edge detection occupies a significant role, because of its independence with the external influence. To get better precision on edge width, we use the Canny detector, which is one of the most accurate edge detection methods. Then, we, respectively, present the edge detector, the BF and BS methods separately, and our new combination method called 3FDBS-LC, in the following sections.

Canny Edge Detector

Canny edge detector is came up with John F. Canny in 1986. The Canny algorithm is designed to meet three main criteria: low error rate, good localization, and mark uniqueness. Owing to its optimality to meet with the three criteria, the canny operator experienced a multistage process:

(a) Use a Gaussian filter to smooth the image and filter out the noise. In order to minimize the impact of noise on edge detection, noise must be filtered out to prevent false detection. The Gaussian convolution kernel H of size $(2k + 1) \times (2k + 1)$ is given below:

$$H_{ij} = \frac{1}{2\pi\sigma^2}\exp\left\{-\frac{(i-(k+1))^2+((j-(k+1))^2}{2\sigma^2}\right\}, \tag{10}$$

where W_S is the size of window; the brightness value of the pixel P is the convolution of H and W_S. The size of the Gaussian convolution kernel can affect the performance of the Canny detector. The larger the size is, the lower the sensitivity of the detector to noise will be. Generally, 5×5 is a relatively good trade-off.

(b) Calculate the gradient intensity and direction of each pixel in the image. The edges in the image can point at all directions, so the Canny algorithm uses multiple operators to detect the image. The gradient intensity value G and direction θ are defined in

$$G_x = S_x * W_S \tag{11}$$

$$G_y = S_y * W_S \tag{12}$$

$$G = \sqrt{G_x^2 + G_y^2} \tag{13}$$

$$\theta = \arctan(G_y/G_x), \tag{14}$$

where S_x denotes the operator in the x direction for detecting the edge in the y direction and S_y denotes the operator in the y direction for detecting the edge in the x direction. G_x and G_y represent the gradient values in the x and y directions, respectively.

(c) Apply non-maximum suppression to eliminate spurious response from edge detection. Non-maximum suppression is an edge sparse technique that helps to suppress all gradient values outside the local maximum to zero. As shown below, the gradient is divided into eight directions, namely E, NE, N, NW, W, SW, S, and SE. The gradient direction of the pixel P is θ; then, the gradient linear interpolation G_{P1} and G_{P2} of the pixels $P1$ and $P2$ is defined as follows:

$$\tan\theta = G_y/G_x \tag{15}$$

$$G_{P1} = (1 - \tan\theta) \times E + \tan\theta \times \mathrm{NE} \tag{16}$$

$$G_{P2} = (1 - \tan\theta) \times W + \tan\theta \times \mathrm{SW}. \tag{17}$$

(d) Use double-threshold detection to determine the true and potential edges.

(e) Finish the edge detection by suppressing the isolated weak edges.

The detailed pseudo-code description for the following three steps is presented in Algorithm 1.

Algorithm 1 Canny edge detection.

```
Require: $Dataset(\ G_P, G_{P1}, G_{P2})$
function NON-MAXIMUM SUPPRESSION($G_P$)
    if $G_P \geq G_{P1}, G_P \geq G_{P2}$ then
        $G_P \rightarrow edge$
    else
        $G_P \rightarrow Suppressed$
    end if
end function

function DOUBLE-THRESHOLD($G_P$)
    if $G_P \geq HighThreshold$ then
        $G_P \rightarrow StrongEdge$
    end if
    if $LowThreshold \leq G_P \leq HighThreshold$ then
        $G_P \rightarrow WeakEdge$
    end if
    if $G_P \leq LowThreshold$ then
        $G_P \rightarrow Suppressed$
    end if
end function

function SUPPRESS ISOLATED LOW THRESHOLD POINTS($G_P$)
    if $G_P == LowThreshold$ then
        $G_P \rightarrow StrongEdge$
    else
        $G_P \rightarrow Suppressed$
    end if
end function
```

Frame Differencing Method

The frame difference method can be implemented on a series of consecutive images. Gray values and gradient vectors are used to determine information for moving objects. The method calculates the difference between two consecutive images by comparing the point-by-point gray values to obtain a frame difference image. The formula for the difference between two frames can be written as

$$D_k(x, y) = |f_k(x, y) - f_{k-1}(x, y)|, \tag{18}$$

where the current frame image gray value is f_k, the adjacent frame image gray values is f_{k-1}, and D_k is image after difference between f_k and f_{k-1}. We define R_k as the binary conversion of the difference image. If $D_k(x, y) > T$, $R_k(x, y)$ belongs to foreground and set to 1; on the contrary, it belongs to the background and it will be set to 0, where T is a fixed empirical threshold.

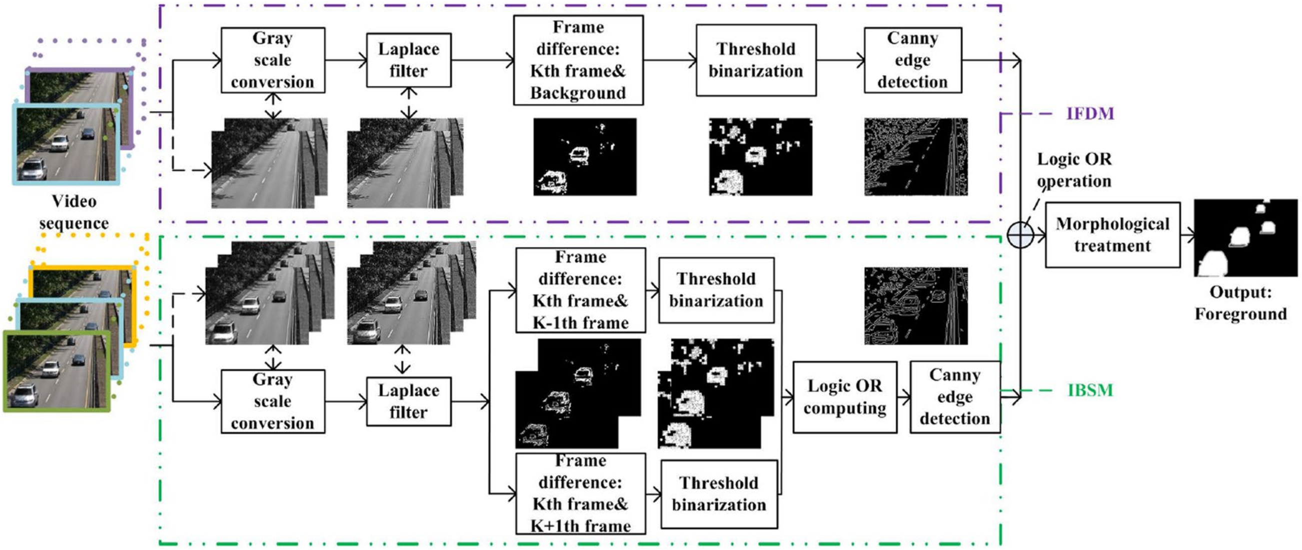


Fig. 7 Framework of our improved algorithm

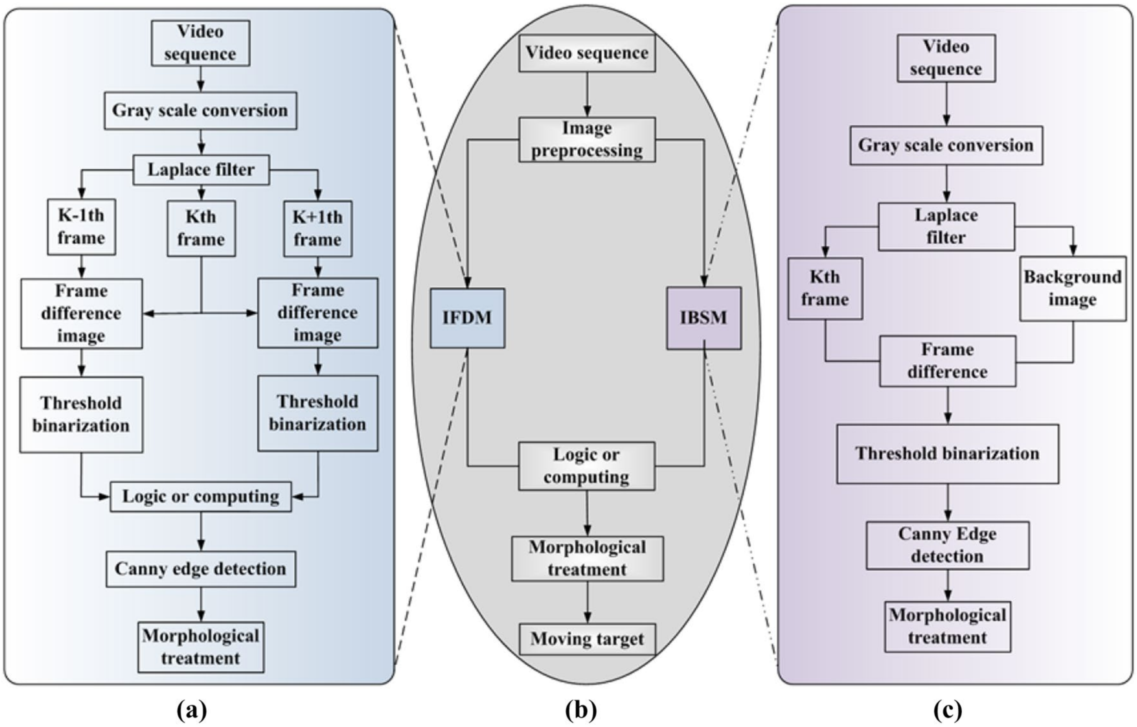


Fig. 8 Flowchart of the proposed algorithm: **a** frame difference component, **b** main algorithm, **c** background subtraction component

The disadvantages of the difference between the two frames are the generation of foreground aperture and ghosting problems. In contrast, the three-frame difference method can better weaken this problem. This is achieved by subtracting the current frame image with the previous frame and the subsequent frame, respectively. After that, a logical OR operation is performed based on these results, as done by Zhang et al. [12]. Here, we will analyze it in detail and name it the traditional frame difference method (FD). Besides, when dealing with a complex scene, the edge information of the moving target is easily affected by the background scene. This edge information cannot be extracted entirely. Conversely, Canny edge detection is good at getting the edge information of an object. Therefore, three-frame differences can be combined with Canny edge detection.

Background Subtraction Method

The principle of background subtraction is to subtract the background image from the current frame using difference computation. The process can be divided into the following two steps. First, the current frame image K_{th} and the background image are obtained from the video sequence. Second, a difference calculation is performed between the current frame image and the latest background image to obtain a frame difference image. Zhang and Liang [10] uses this background subtraction (BS) method with morphological filtering and contour projection analysis as post-processing. However, due to noise, shadows, and light interference, the results of the difference may be irrelevant. The challenge is to propose a background optimization method that can filter these unavoidable disturbances while correctly detecting moving objects. Therefore, an improved background subtraction method has been proposed in which an accurate Canny detector is inserted.

Proposed 3FDBS-LC Detection Method

An outline of the entire processing flowchart of this 3FDBS-LC method is summarized in Fig. 7 and in Fig. 8. The pseudo-code description for the method is presented in Algorithm 2. First, after converting a color image into a grayscale image, the Laplace filter occupying the dominant character will sharpen the outline and detail of the grayscale target. Secondly, three-frame difference and background difference operations are performed separately. Then, threshold binarization and Canny edge detection are performed to identify and extract edge information. Finally, the combination of these two main methods undergoes a logical OR operation followed by a morphological operation for obtaining the final moving object shapes. Once all operations are performed, the process enters the next cycle for real-time monitoring. Note that treatments are straightforward operations roughly executed within a $O(N)$ time complexity, with N the number of pixels, that make the global method a good candidate for real-time execution. Also, their intrinsic parallelism should allow efficient parallel implementation in GPU systems.

Algorithm 2 Proposed method.

```
Require: $Dataset(\ m_{k-1}, m_k, m_{k+1}, b_k)$
function FRAME DIFFERENCE($m_{k-1}, m_k, m_{k+1}$)
    $T_k \leftarrow T(\text{adaptive})$
    for $Input : m_{k-1}, m_k, m_{k+1}$ do
        $f_{k-1} \leftarrow n_{k-1} \leftarrow m_{k-1}$
        $f_k \leftarrow n_k \leftarrow m_k$
        $f_{k+1} \leftarrow n_{k+1} \leftarrow m_{k+1}$
        $(f_{Laplace} \leftarrow n_{GrayScale} \leftarrow m_{ColorScale})$
        $D_k \leftarrow |f_k - f_{k-1}|$
        $D_{k+1} \leftarrow |f_{k+1} - f_k|$ // Difference opration
        if $D_k or D_{k+1} < T_k$ then
            $R_k, R_{k+1} \leftarrow 0(background)$ // Binary operation
        else
            $R_k, R_{k+1} \leftarrow 1(object)$
        end if
        $R_k \cup R_{k+1} \rightarrow FD$
        $FD + Canny \rightarrow FD_c$
        $FD_c \rightarrow FD_m$ // Morphology processing
    end for
    return $FD_m$
end function
function BACKGROUND SUBTRACTION($m_k, b_k$)
    $T_k \leftarrow T(\text{adaptive})$
    for $Input : m_k, b_k$ do
        $f_k \leftarrow n_k \leftarrow m_k$
        $b_{Laplace} \leftarrow b_{GrayScale} \leftarrow b_k$
        $D'_k \leftarrow |f_k - b_{Laplace}|$
        if $D'_k < T_k$ then
            $R'_k \leftarrow 0(background)$
        else
            $R'_k \leftarrow 1(object)$
        end if
        $R'_k + Canny \rightarrow BS_c, BS_c \rightarrow BS_m$
    end for
    return $BS_m$
end function
function IFDM ADD IBSM($FD_m, BS_m$)
    $FD_m \cap BS_m \rightarrow result$
    return $result$
end function
```

Experiment and Evaluation

Datasets

Here, three benchmarks, the SABS dataset, the Wallflower dataset, and the Multivision dataset, are applied to experiments. They are used for visual demonstration, comparative experimentations, and numerical evaluation under different standard criteria.

The SABS (Stuttgart Artificial Background Subtraction) dataset[1] is an artificial benchmark for pixel evaluation of background models [21]. SABS consists of video sequences with nine different background external changes for video surveillance. It has been added global illumination and Gaussian noise. Compared to other manually ground truth datasets, the SABS dataset does not so much suffer from imperfect labels. SABS contains ground truth annotation and additional shadow annotation for detection evaluation.

[1] http://www.vis.uni-stuttgart.de/en/research/information-visualisation-and-visual-analytics/visual-analytics-of-video-data/sabs.html.

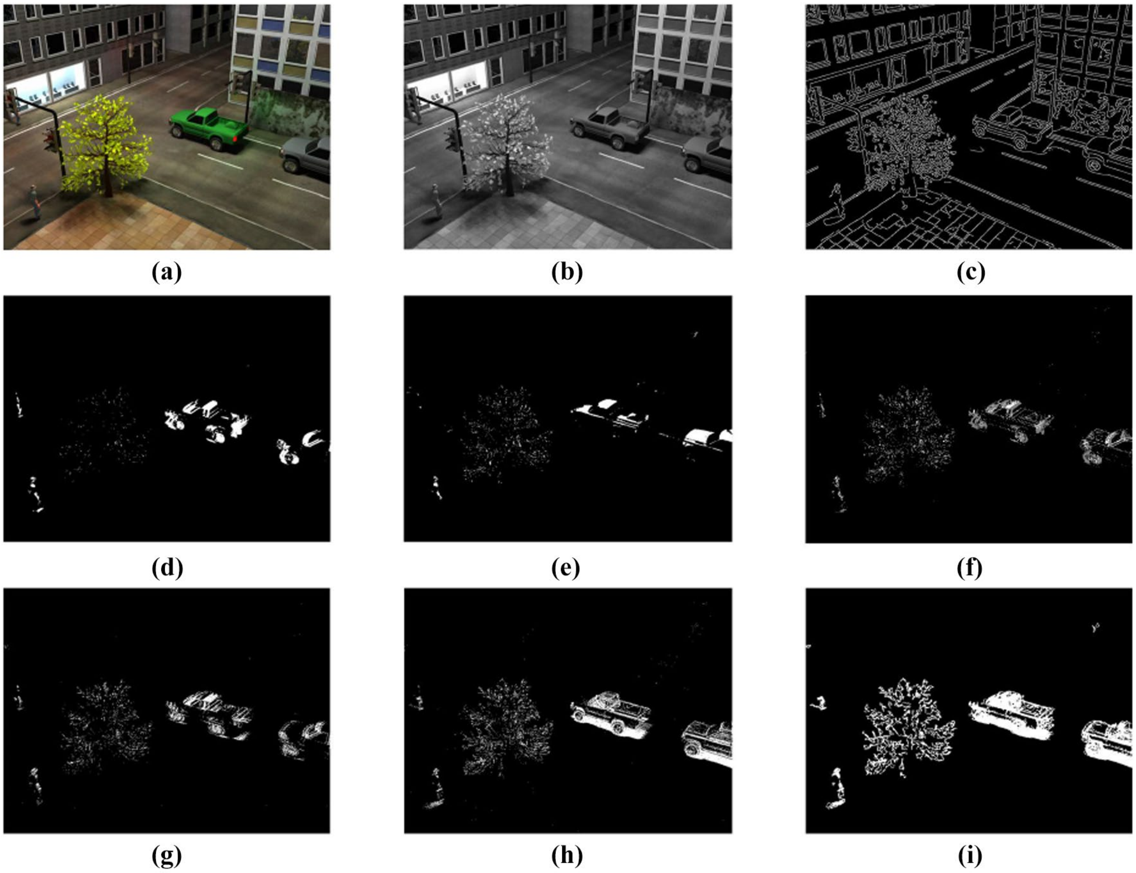


Fig. 9 From the left to the right: **a** original color scale image, **b** gray-scale image, **c** image processed by the Canny edge detector, **d** image extracted by standard three-frame difference, **e** image extracted by standard background difference, **f** the logic OR operation between (**d**) and (**e**), **g** the improved three-frame difference method after Laplace filter, **h** the improved background subtraction method after Laplace filter, and **i** the improved 3FDBS-LC method

Wallflower dataset[2] consists of seven test scenarios [22]. Each scenario represents a different, potentially problematic situation for background maintenance. When dealing with these image sequences, the output of the algorithm is divided into background image and foreground image, accompanying with their corresponding hand-segmented evaluation image. In order to deal with various problems that arise at the spatial scale, the evaluation image is segmented at pixels, regions, and frames levels. These training images, test images, and hand-segmented evaluation images are useful for training, evaluation, and comparison work.

Multivision dataset[3] is a database for evaluation of hardware/software real-time vision systems based on multiple cameras [23]. In a vision system, the goal is to translate the image into detailed information and to provide a visual solution that efficiently processes images taken from multiple cameras and complements the estimation reliably and robustly. The benchmark provides a dataset with ground truth segmentation, which enables to carry out objective evaluation of frame difference algorithms and background subtraction algorithms, as required in our study.

Evaluation Criteria

Based on ground truth assessment, some evaluation criteria are defined to assess and compare the data results between different detection methods. In pattern recognition and information retrieval, precision is an indicator for the relevance of the results, and recall is a measure of the return of real relevant results. The experimental output quality is evaluated in this experiment by using accuracy, recall, precision, and F-measure.

[2] https://www.microsoft.com/en-us/download/details.aspx?id=54965.

[3] http://atcproyectos.ugr.es/mvision/.

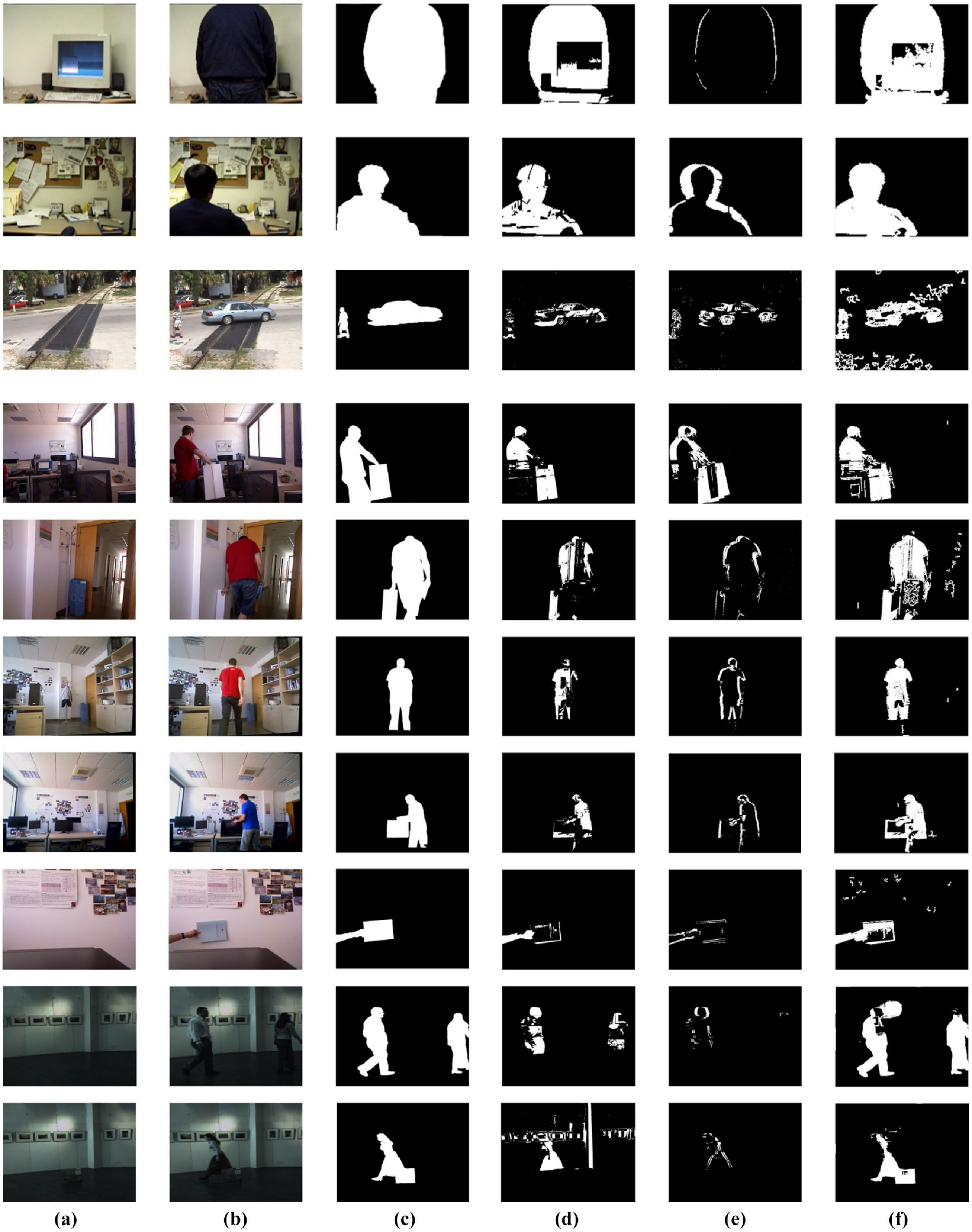


Fig. 10 From the left to the right: **a** background image, **b** frame image, **c** ground truth image, **d** background subtraction method, **e** frame difference method, and **f** the proposed 3FDBS-LC method

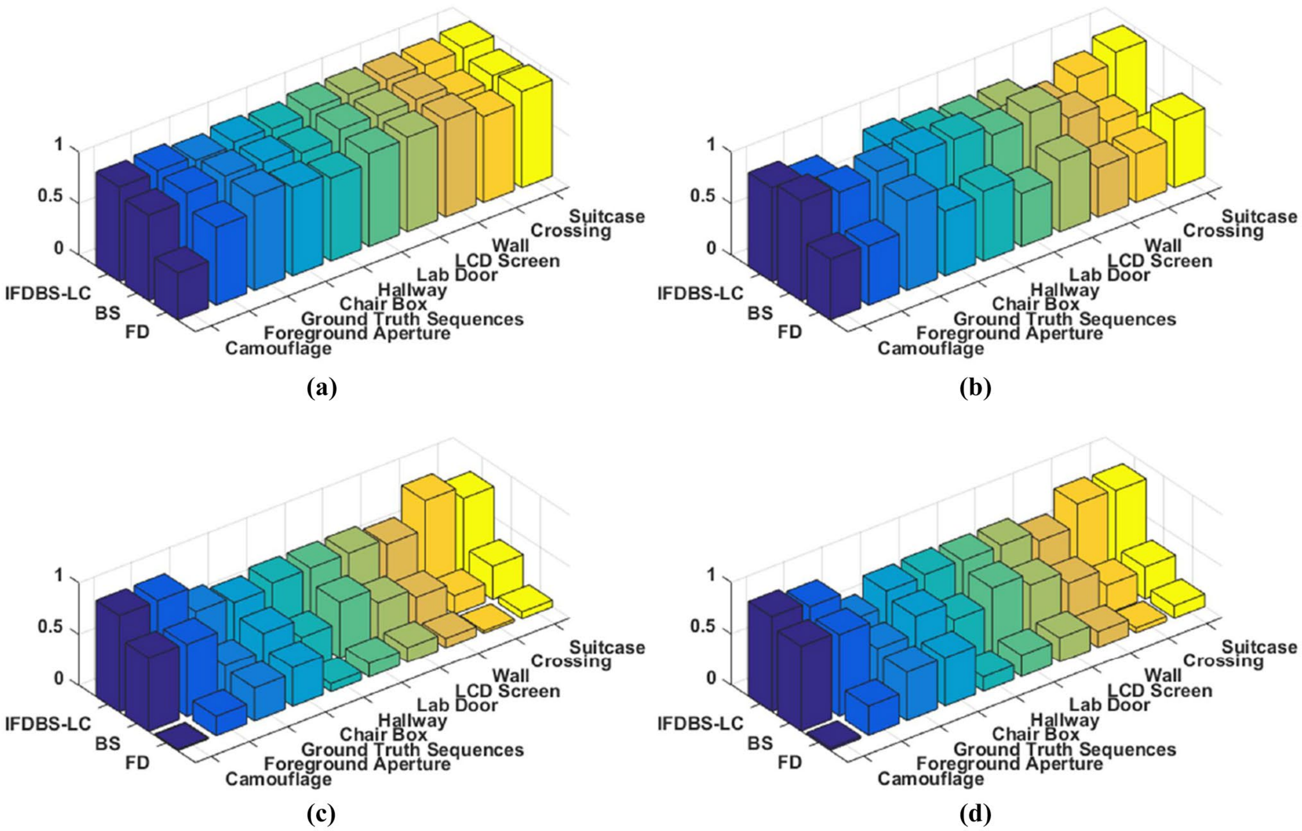


Fig. 11 Comparison histograms of three different kinds of methods in **a** accuracy, **b** precision, **c** recall, and **d** F-measure

Accuracy is defined as the number of true positives (TP) plus the number of true negatives (TN) over all of the samples. Formally,

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{FP} + \text{TN} + F\text{N}), \tag{19}$$

where TP is the number of foreground pixels that are correctly defined as foreground, TN is the number of background pixels that are correctly defined as background, FP is the number of background pixels that are mistakenly defined as foreground, and FN is the number of foreground pixels that are mistakenly defined as background.

Recall is defined as the number of true positives (TP) over the number of true positives plus the number of false negatives (FN). Then,

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}). \tag{20}$$

Precision is defined as the number of true positives (TP) over the number of true positives plus the number of false positives (FP).

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}). \tag{21}$$

F-measure is defined as the harmonic mean of precision and recall.

$$F\text{-measure} = \frac{2\text{Recall} * \text{Precision}}{(\text{Recall} + \text{Precision})}. \tag{22}$$

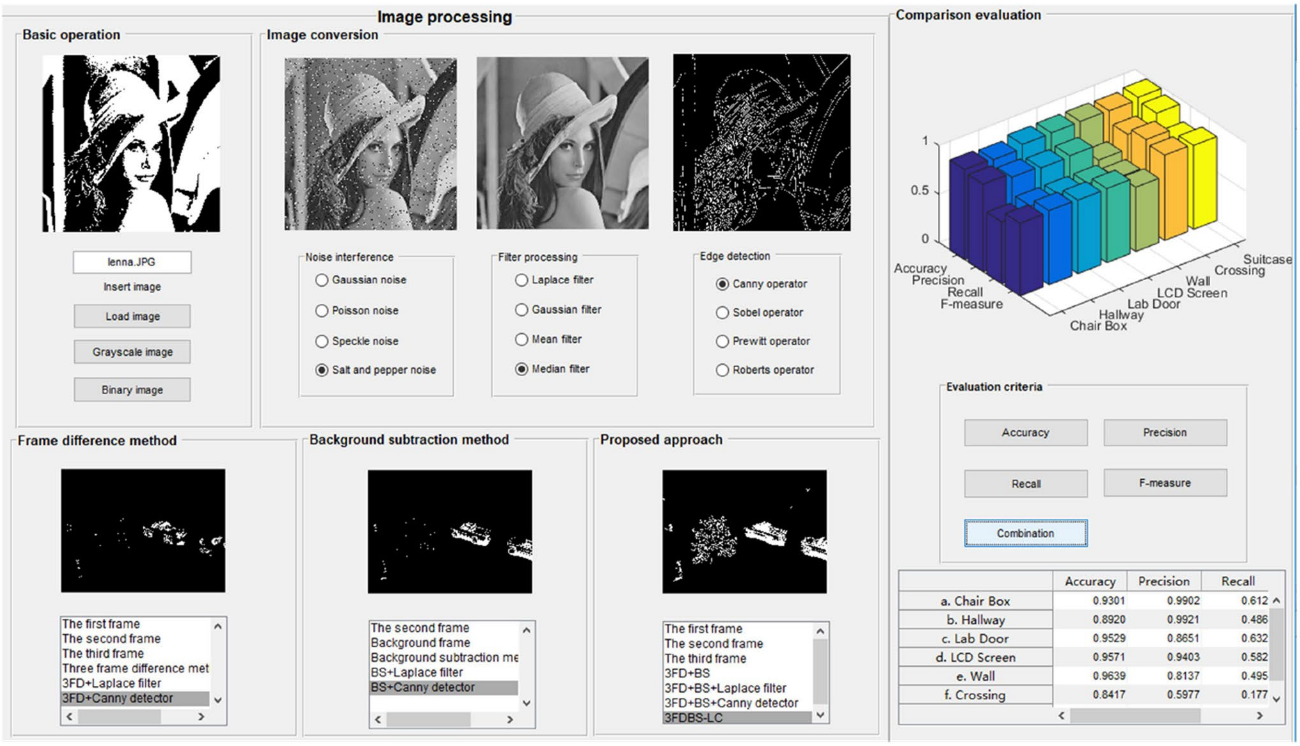
High scores for F-measure show that the classifier is returning accurate results (high precision), as well as returning a majority of all positive results (high recall).

Qualitative Evaluation of the Sequence of Treatments

The SABS-Bootstrap sequences with 352 × 288 images are used to demonstrate the results after different treatments, as shown in Fig. 9. On these images, we can find the results about the standard BS method, the FD method, their combination with or without the Laplace filter [as shown in (d)–(h)] and the proposed method (i). We can visually check that the proposed method (i) can more clearly point out moving objects: running cars, walking pedestrians, and swinging trees that are blown by the wind.

Table 1 Different kinds of metrics of their corresponding datasets

	Accuracy			Precision		
Datasets	3FDBS-LC	BS	FD	3FDBS-LC	BS	FD
Camouflage	0.9153	0.8648	0.4522	0.9090	0.9480	0.5882
F-A	0.9319	0.9260	0.7525	0.8230	0.9095	0.5746
GT-S	0.8889	0.9420	0.9144	0.5207	0.9467	0.8702
Chair Box	0.9244	0.9301	0.8534	0.8980	0.9902	0.6264
Hallway	0.9119	0.8920	0.8022	0.8552	0.9921	0.6764
Lab Door	0.9547	0.9529	0.8996	0.8369	0.8651	0.5137
LCD Screen	0.9601	0.9571	0.9146	0.8484	0.9403	0.6844
Wall	0.9625	0.9639	0.9405	0.6904	0.8137	0.4795
Crossing	0.9579	0.8417	0.8311	0.8399	0.5977	0.4760
Suitcase	0.9822	0.8997	0.9318	0.9438	0.2978	0.6573
Datasets	Recall			*F*-measure		
Camouflage	0.9380	0.9480	0.0115	0.9232	0.8673	0.0226
F-A	0.9405	0.8021	0.1871	0.8778	0.8524	0.2823
GT-S	0.6922	0.5372	0.3195	0.5943	0.6854	0.4674
Chair Box	0.6363	0.6128	0.3736	0.7449	0.7571	0.4680
Hallway	0.6842	0.4866	0.0767	0.7602	0.6529	0.1379
Lab Door	0.6858	0.6320	0.1307	0.7539	0.7304	0.2084
LCD Screen	0.6864	0.5822	0.1417	0.7589	0.7191	0.2348
Wall	0.6313	0.4957	0.0908	0.6595	0.6161	0.1527
Crossing	0.9104	0.1773	0.0251	0.8737	0.2735	0.0477
Suitcase	0.7897	0.3194	0.0657	0.8599	0.3082	0.1195



lenna.JPG

Noise interference
○ Gaussian noise
○ Poisson noise
○ Speckle noise
● Salt and pepper noise

Filter processing
○ Laplace filter
○ Gaussian filter
○ Mean filter
● Median filter

Edge detection
● Canny operator
○ Sobel operator
○ Prewitt operator
○ Roberts operator

The first frame
The second frame
The third frame
Three frame difference met
3FD+Laplace filter
3FD+Canny detector

The second frame
Background frame
Background subtraction me
BS+Laplace filter
BS+Canny detector

The first frame
The second frame
The third frame
3FD+BS
3FD+BS+Laplace filter
3FD+BS+Canny detector
3FDBS-LC

	Accuracy	Precision	Recall
a. Chair Box	0.9301	0.9902	0.612
b. Hallway	0.8920	0.9921	0.486
c. Lab Door	0.9529	0.8651	0.632
d. LCD Screen	0.9571	0.9403	0.582
e. Wall	0.9639	0.8137	0.495
f. Crossing	0.8417	0.5977	0.177

Fig. 12 Display for object detection system

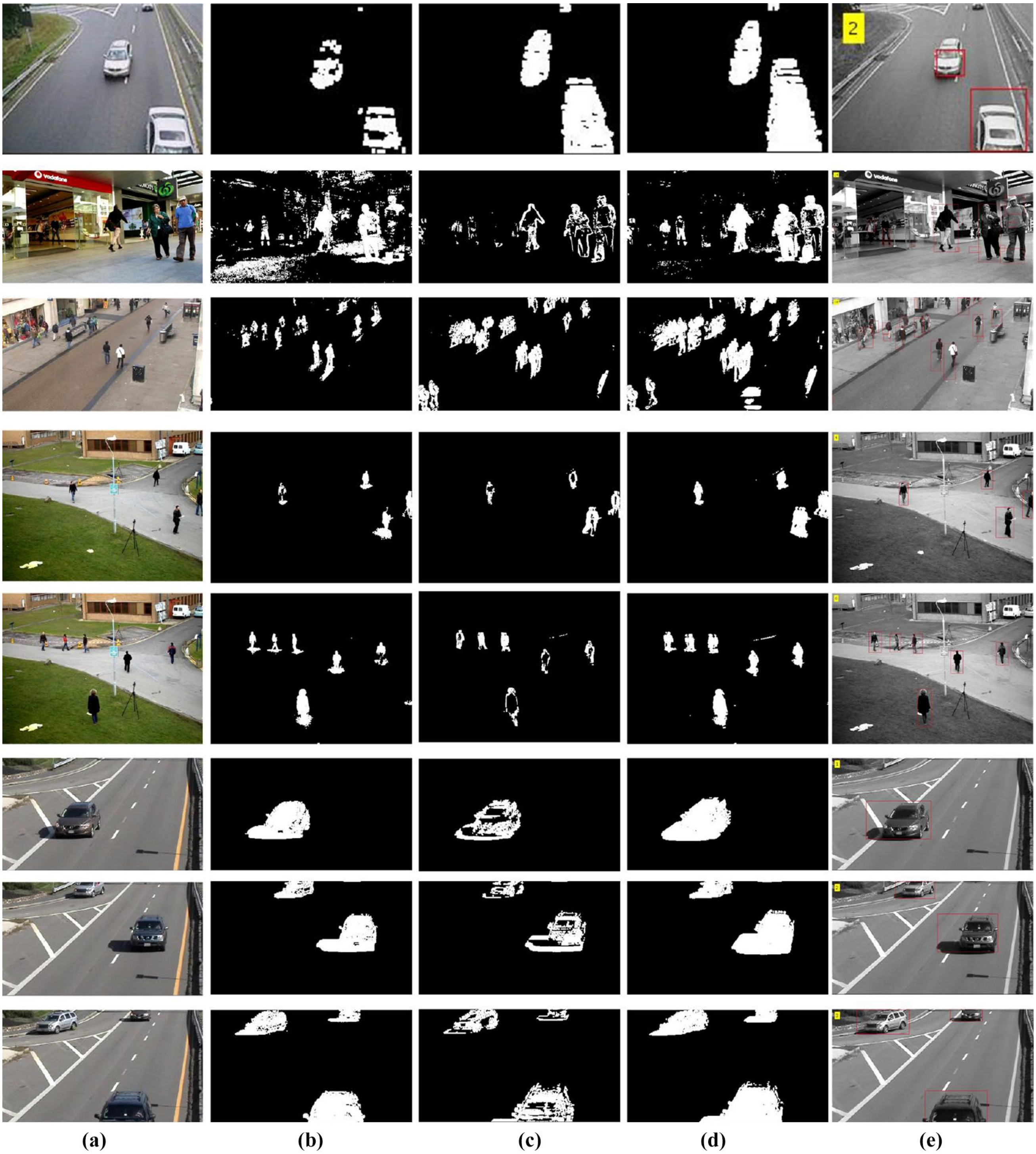


Fig. 13 Applied to actual scene in real-time surveillance: **a** original single frame, **b** BS, **c** FD, **d** IFDBS-LC, **e** realized by rectangle box

Comparative Evaluation

In the following comparative evaluation, all of the algorithm parameters were set as detailed in the previous sections and remained fixed for all the experiments. Based on Wallflower and Multivision datasets, ten image sequences are used: Camouflage, Foreground Aperture, Ground Truth Sequences, Chair Box, Hallway, Lab Door, LCD Screen, Wall, Crossing, and Suitcase. A visual presentation of the results obtained by two standard algorithms and by the proposed 3FDBS-LC method is given in Fig. 10. The first column presents background images, the second column demonstrates every sample frame per sequences, ground truth images are shown in the third column, the fourth and fifth columns display the detected foreground under standard background subtraction and frame difference method, respectively, and moreover, the last column is the result of proposed 3FDBS-LC method. The improvement in shape detection should qualitatively be appreciated in the figure.

The quantitative numerical evaluations based on ground truth are reported in Table 1. A comparative evaluation of accuracy, precision, recall, and *F*-measure for three different detection methods is included under the ten different image sequences. Figure 11 shows their corresponding histograms. From these results, it can be found that this proposed 3FDBS-LC algorithm can obtain good detection results superior to standard BS and FD methods.

Application to Real-Time Video Processing

In this section, we present implementation for real-time video monitoring. Our systematic detection algorithm is realized as a set of MATLAB functions, embedded in a real-time video rate-driven loop, managed on a Graphical User Interface (GUI) platform for a convenient visualization and analysis, in a way similar to Andreatos and Zagorianos [24]. As can be seen from Fig. 12, this GUI interface [25] mainly contains necessary image processing and experimental evaluation modules. Rectangle box and silhouette-based representation methods mainly realize the actual scene view implementation.

We test the proposed method by using moving sequence for moving target detection with application in real-time surveillance video. Our proposed method aims to detect all of the targets which are moving over an entire video sequence. This detection process is primarily shooting different consecutive images or frames at different time intervals. In our experiment, we use a CCTV (Closed-Circuit TeleVision) video sequence of automobile traffic presenting moving cars.

The multiple Object Tracking Benchmark,[4] and Active Vision Laboratory Benchmark[5] are used for qualitative visual evaluation and comparison. As depicted in Fig. 13, we have demonstrated through qualitative evaluation that the system can provide accurate position estimation for a large number of vehicles or pedestrians in real time. According to real-time validation, the actual implementation allows to deal with standard video rate of 24 frames by second. Also, because of the parallel nature of most of the treatments, the fastest video rate processing is envisaged by GPU implementation. This combined algorithm could then be used to track an unknown number of mobile topics with higher accuracy of the observed target shapes and in real time.

Conclusion

In this paper, an improved object detection algorithm is proposed by systematically combining important features of background subtraction and frame difference methods usually employed in real-time surveillance detection. The method mainly contains Laplace filter, frame difference method, background subtraction method, and Canny edge detector, which have real-time implementation available. The proposed algorithm was tested on standard datasets with the evaluation criteria of accuracy, recall, precision, and *F*-measure and was compared, based on ground truth evaluation, to the standard BS and FD methods. Results demonstrated an improvement in accuracy over the standard methods, and computation time of the overall method remains compatible with standard video rate on a personal computer. Also, since these procedures are parallel by nature, the design of software in relation to GPU system is a matter of current investigation to further accelerate treatments.

Acknowledgements This work was supported by China Scholarship Council (201604490109).

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Luo W, Xing J, Milan A, Zhang X, Liu W, Zhao X, Kim TK. Multiple object tracking: a literature review. ArXiv Journal; 2014. arXiv:1409.7618.
2. Hu WC, Chen CH, Chen TY, Huang DY, Wu ZC. Moving object detection and tracking from video captured by moving camera. J Vis Commun Image Represent. 2015;30:164–80.
3. Yazdi M, Bouwmans T. New trends on moving object detection in video images captured by a moving camera: a survey. Comput Sci Rev. 2018;28:157–77.
4. Baek I, Davies A, Yan G, Rajkumar RR. Real-time detection, tracking, and classification of moving and stationary objects using multiple fisheye images. In: 2018 IEEE intelligent vehicles symposium (IV). IEEE; 2018. p. 447–52.
5. Lee C, Moon JH. Robust lane detection and tracking for real-time applications. IEEE Trans Intell Transp Syst. 2018;19(12):4043–8.
6. Hu HN, Cai QZ, Wang D, Lin J, Sun M, Krahenbuhl P, Yu F. Joint monocular 3D vehicle detection and tracking. In: Proceedings of the IEEE international conference on computer vision; 2019. p. 5390–9.
7. Henschel R, Leal-Taixe L, Cremers D, Rosenhahn B. Fusion of head and full-body detectors for multi-object tracking. In:

[4] https://motchallenge.net/vis/PETS09-S2L1.

[5] http://www.robots.ox.ac.uk/ActiveVision/index.html.

Proceedings of the IEEE conference on computer vision and pattern recognition workshops; 2018. p. 1428–37.

8. Shotton J, Blake A, Cipolla R. Contour-based learning for object detection. In: Tenth IEEE international conference on computer vision (ICCV’05), vol. 1. IEEE; 2005. p. 503–10.
9. Tu Z, Xie W, Zhang D, Poppe R, Veltkamp RC, Li B, Yuan J. A survey of variational and CNN-based optical flow techniques. Signal Process Image Commun. 2019;72:9–24.
10. Zhang L, Liang Y. Motion human detection based on background subtraction. In: 2010 Second international workshop on education technology and computer science, vol. 1. IEEE; 2010. p. 284–7.
11. Zhong Z, Zhang B, Lu G, Zhao Y, Xu Y. An adaptive background modeling method for foreground segmentation. IEEE Trans Intell Transp Syst. 2016;18(5):1109–21.
12. Zhang Y, Wang X, Qu B. Three-frame difference algorithm research based on mathematical morphology. Proc Eng. 2012;29:2705–9.
13. Wang Q, Gao J, Yuan Y. Embedding structured contour and location prior in siamesed fully convolutional networks for road detection. IEEE Trans Intell Transp Syst. 2017;19(1):230–41.
14. Dosovitskiy A, Fischer P, Ilg E, Hausser P, Hazirbas C, Golkov V, Van Der Smagt P, Cremers D, Brox T. Flownet: learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 2758–66.
15. Niu L, Jiang N. A moving objects detection algorithm based on improved background subtraction. In: 2008 eighth international conference on intelligent systems design and applications, vol. 3. IEEE; 2008. p. 604–7.
16. Zhan C, Duan X, Xu S, Song Z, Luo M. An improved moving object detection algorithm based on frame difference and edge detection. In: Fourth international conference on image and graphics (ICIG 2007). IEEE; 2007. p. 519–23.
17. Weng M, Huang G, Da X. A new interframe difference algorithm for moving target detection. In: 2010 3rd international congress on image and signal processing, vol. 1. IEEE; 2010. p. 285–9.
18. Gang L, Shangkun N, Yugan Y, Guanglei W, Siguo Z. An improved moving objects detection algorithm. In: 2013 International conference on wavelet analysis and pattern recognition. IEEE; 2013. p. 96–102.
19. Liu H, Dai J, Wang R, Zheng H, Zheng B. Combining background subtraction and three-frame difference to detect moving object from underwater video. In: OCEANS 2016-Shanghai. IEEE; 2016. p. 1–5.
20. Yuan Y, Xiong Z, Wang Q. VSSA-NET: vertical spatial sequence attention network for traffic sign detection. IEEE Trans Image Process. 2019;28(7):3423–34.
21. Brutzer S, Höferlin B, Heidemann G. Evaluation of background subtraction techniques for video surveillance. In: CVPR 2011. IEEE; 2011. p. 1937–44.
22. Toyama K, Krumm J, Brumitt B, Meyers B. Wallflower: principles and practice of background maintenance. In: Proceedings of the seventh IEEE international conference on computer vision, vol. 1. IEEE; 1999. p. 255–61.
23. Fernandez-Sanchez EJ, Rubio L, Diaz J, Ros E. Background subtraction model based on color and depth cues. Mach Vis Appl. 2014;25(5):1211–25.
24. Andreatos AS, Zagorianos A. Matlab GUI application for teaching control systems. In: Proceedings of the 6th WSEAS international conference on engineering education; 2009. p. 208.
25. Cui B, Creput JC. Matlab GUI application for moving object detection and tracking. In: International symposium on distributed computing and artificial intelligence. Cham: Springer; 2018. p. 353–6.

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