International Journal of Innovative Computing, Information and Control
Volume **14**, Number **1**, February **2018**

ICIC International ©2018 ISSN 1349-4198
pp. 279–295

PEDESTRIAN DETECTION BASED ON MODIFIED DYNAMIC BACKGROUND USING GAUSSIAN MIXTURE MODELS AND HOG-SVM DETECTION

Jia-Qi Gui and Zhe-Ming Lu*

School of Aeronautics and Astronautics
Zhejiang University
No. 38, Zheda Road, Hangzhou 310027, P. R. China
21624031@zju.edu.cn; *Corresponding author: zheminglu@zju.edu.cn

Received May 2017; revised September 2017

Abstract. *In this paper, we propose a fast pedestrian detection method based on surveillance video clips under stationary cameras. Our purpose is to address the problem of low speed of pedestrian detection with an HOG-SVM detector. First, the background modeling using the mixture of Gaussians is used to extract the moving objects in the video. Then, three steps, i.e., shadow removing, eroding and dilating and border expanding are performed to make further alterations to the extracted foreground. At the same time, our experiments based on the INRIA dataset calculate the histogram of oriented gradients feature of the whole pedestrians and classify them by a support vector machine. Experimental results indicate that the foreground extracted by our background modeling scheme can contain all the moving objects well through shadow removing and border expanding. So the proposed methods outperform the traditional HOG+SVM method in both recognition accuracy and processing speed.*

Keywords: Gaussian mixture model, Shadow removing, Eroding and dilating, Border expanding, HOG+SVM, Pedestrian detection

1. **Introduction.** In video surveillance, the main task of pedestrian detection is to spot the dynamic pedestrians from video sequences. However, because of the diversity of pedestrians such as the differences in their appearance, clothing, shapes and gestures, as well as some uncontrollable external factors like the light changes, the camera shakes, and the tree sticks shake, the pedestrian model is easily influenced. So there is a big challenge for us to let the pedestrian be extracted from the video fast and efficiently. In this way, the pedestrian detection has been a very hot topic in the research area of computer vision. At present, the methods of pedestrian detection are mainly separated into two types: traditional pedestrian detection methods and machine-learning pedestrian detection methods.

It is difficult for conventional methods to detect pedestrian because of the above change. However, machine-learning pedestrian detection methods train the feature by learning the sample of dataset repeatedly. It has a higher robustness. It will be faster for calculating and modeling in pedestrian images when we use formula features to represent pedestrian information. Machine-learning pedestrian detection methods generally contain three parts: feature extraction, training machine learning classifier and detection. Machine learning is the mainstream method in pedestrian detection currently. It mainly uses image features such as edge, shape and color in static images to describe the pedestrian regions. Among them, some features can be used to detect pedestrians well, like the Haar wavelets feature [1], the HOG feature, the Edgelet feature [2], the Shapelet feature [3] and the shape contour template feature [4].

In recent years, a new pedestrian detection method based on deep learning [5] has been proposed. The tasks considered to be the state of art include image classification [6], face recognition [7,8] and object detection [9,10]. Deep learning is a new field in research of the machine learning. The feature named histogram of oriented gradients (HOG) is the main concern in our paper, which was proposed by Dalal and Triggs [11] in 2005. This normalized feature descriptor extracted the gradient from the overlapped image blocks to represent the objects. Dalal and Triggs designed the HOG-SVM detector and applied it in pedestrian detection. Experimental results on the MIT database set show that the HOG-SVM algorithm detected pedestrian correctly in 100% cases. HOG is currently a normalized feature descriptor widely used in pedestrian detection. However, the HOG feature needs a fixed size of detection window to scan the whole frame of the video complexly. Then HOG features were calculated in each scanning window and used to train an SVM detector, which greatly caused the high computational complexity and poor real-time performance.

According to the characteristic of the video surveillance, our paper proposes a modified algorithm to improve the speed of pedestrian detection by using HOG features. First, our paper makes full use of the background modeling method based on mixture of Gaussians to extract moving objects and then removes shadows in the foreground later. So the HOG detection area can be reduced. Thus, through border expanding, all the moving objects can be completely contained in the scanning region. The final detection in the foreground is guided by the HOG feature and SVM classifier. Experimental results demonstrate the effectiveness and veracity of our algorithm.

The paper is organized as follows. Section 2 briefly introduces the background modeling using mixture of Gaussians. Section 3 describes details of the HOG and linear SVM. The following section presents the results of our experiments on three videos. Finally, Section 5 concludes this paper.

2. **Background Modeling Using Mixture of Gaussians.** Background modeling using mixture of Gaussians is a classical algorithm of basic background subtraction [12]. It uses a Gaussian probability-density function to precisely quantify things. In this technique, each pixel of a scene is modeled independently by a mixture of at most K Gaussian distributions. With the arrival of the new images, parameters (average, mean and weight) of Gaussian distributions are continually updated [13], and each pixel has to be matched to Gaussian distributions to determine whether it is updated or not. So it can accurately characterize the background information in real time. Compared with the single Gaussian model, it can commendably deal with dynamic background which is regularly changing as well as individual and mutational background models.

The formula of the background modeling using mixture of Gaussians is as follows:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} \times \eta(x_t, u_{i,t}, \Sigma_{i,t}) \tag{1}$$

where $K \in \{3, 4, 5\}$ is the number of Gaussian distributions in the model; $\omega_{i,t}$ is the weight parameter of the ith Gaussian distribution at time t; $u_{i,t}$ is the mean of the ith Gaussian distribution at time t; $\Sigma_{i,t}$ is the covariance of the ith Gaussian distribution at time t; η is a Gaussian probability density function defined as:

$$\eta\left(x_t, u, \Sigma\right) = \frac{1}{(2\pi)^{\frac{n}{2}} \left|\Sigma\right|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - u_t)^T \Sigma^{-1} (X_t - u_t)} \tag{2}$$

Based on the independent assumptions of color, covariance is defined as $\Sigma_{k,t} = \sigma_k^2 I$ (here, σ_k^2 is the standard deviation of the kth Gaussian distribution). If one pixel value in

a frame is satisfied with $|x_t - u_{k,t}| < D \times \sigma$, in other words, the current pixel x_t matches the kth Gaussian, so $M_{k,t} = 1$ and when unmatched, $M_{k,t} = 0$. D is a constant threshold identical to 2.5, which controls the rigorous level of foreground extraction. The smaller the value is, the more stringent the demands are. The other parameters are updated by the following formulas.

$$w_{k,t} = (1-\alpha)w_{k,t-1} + \alpha(M_{k,t}) \tag{3}$$

$$u_t = I(x,y,t) = \begin{cases} (1-\rho)u_{t-1} + \rho X_t & \text{when } M_{i,t} = 1 \\ u_{t-1} & \text{when } M_{i,t} = 0 \end{cases} \tag{4}$$

$$\sigma_t^2 = \begin{cases} (1-\rho)\sigma_{t-1}^2 + \rho(X_t - u_t)^T(X_t - u_t) & \text{when } M_{i,t} = 1 \\ \sigma_{t-1}^2 & \text{when } M_{i,t} = 0 \end{cases} \tag{5}$$

$$\rho = \alpha\eta(x_t|u_k, \sigma_k) \tag{6}$$

If no match is found, the pixel belongs to foreground, and a new Gaussian distribution is established to replace the original Gaussian distribution whose priority is the smallest. This method uses the value of the nearest pixel as the average of the new Gaussian distribution, and then it initializes a smaller weight and a larger variance. With the time going by, for an updated mixture model, if one pixel has always matched one distribution of K Gaussian distributions, i.e., $M_{k,t} = 1$, then as the time goes on, w will constantly increase and σ will constantly keep decreasing. By sorting for w/σ, w is normalized again. By setting weights and threshold, we cut out the former b Gaussian distributions with the highest weight, which are taken as the background model:

$$B = \arg\min_b \left(\sum_{k=1}^{b} w_k > T \right) \tag{7}$$

where b is a parameter from 1 to K, and T is a threshold value chosen high for multi-model distribution with repetitive motion in background. If the threshold T is small, the model is often a single Gaussian model, which is the best Gaussian distribution (the weight is the largest). If the threshold T is big, it will use multiple distributions as models, and it is stable for scenes such as shaking of leaves, and lake's ripples. The pixel is judged to the background pixel as long as it matches any one of the former b Gaussian distributions. Otherwise, the pixel belongs to foreground.

3. **Pedestrian Detection Based on HOG and Linear SVM.**

3.1. **The HOG feature.**

Histogram of oriented gradients (HOG) is a feature descriptor used in computer vision and image processing for object detection. It composes the feature by statistically calculating the HOG of the images' local area. In the decision-making process, the special feature is extracted first, and then the HOG algorithm uses it to detect whether the interested objects are in the image. HOG features improve system's performance by using standardization of local differences, which result in a high dependency of the edge segmentation and the demand of a large amount of computation. Initially, the HOG algorithm was used to detect static pedestrians, and then it was improved to detect pedestrians in video. However, because of its computational complexity, this method is not real-time. The processing pipeline of the HOG feature extraction can be illustrated as follows.

1) Gray processing: This task changes the input color image into a gray-scale image $I(x, y)$.

2) Standardization of γ and color space: In order to reduce the impact of light, the algorithm must normalize the picture first. In the textural feature of the image, the local surface exposure occupies a large proportion, so this compression can effectively reduce changes of the local shadow and lighting and can inhibit the noise interference. The compression formula can be given as follows (Here, the value of γ is taken as 1/2 [14]):

$$I(x, y) = I(x, y)^{\gamma} \tag{8}$$

3) Gradient computation: Gradient calculation is an important step in the HOG algorithm, which calculates the gradient direction of each pixel position by calculating the abscissa and ordinate gradient of the image. The main purpose is to capture the contour information, and weaken the interference of light any further, and gradient calculation can be conducted by following equations:

$$G_x(x, y) = H(x + 1, y) - H(x - 1, y) \tag{9}$$

$$G_y(x, y) = H(x, y + 1) - H(x, y - 1) \tag{10}$$

where $G_x(x, y)$, $G_y(x, y)$ and $H(x, y)$ respectively represent the horizontal gradient, the vertical gradient, and the pixel value at the location (x, y) in the input image. First, the convolution operation is performed on the original image by respectively using the two convolution kernels $[-1, 0, 1]$ and $[-1, 0, 1]^T$, and then the task obtains the gradient component $G_x(x, y)$ of the horizontal direction and the gradient component $G_y(x, y)$ of the vertical direction. And the next step is to calculate the gradient magnitude and direction by using the following equations:

$$G(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2} \tag{11}$$

$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right) \tag{12}$$

4) Formation of cell histogram: After calculating the gradient, the algorithm defines a fixed-size detection area (for example, 64×128 pixels) to scan the picture. In this window, the window image is divided into a number of small cells, such as, 6×6, and 8×8 (6×6 pixels grouping is considered the best solution for human detection). For 64×128 window images, our paper uses 8×8-sized cells to divide, it segments the window into 128 small cell units, and then the algorithm will group cells in larger spatial structures, called block. The sliding step of the block is a cell size, so it will produce 105 blocks as shown in Figure 1.

   The next step is to calculate the gradient histogram within each cell. First of all, the gradient orientation of each cell is divided into nine blocks in 0°-360° (That is, every 20 degrees is a direction (bin)), as shown in Figure 2. Then, a histogram with nine orientation bins will be computed. Magnitude ($|G(x, y)|$) whose angle ($\alpha(x, y)$) belongs to the same bin will be added up as the value of this bin. In this way, the cell's histogram of gradient orientations is constructed.

5) The block normalization: Due to changes of local illumination and the contrast ratio of foreground-background, the range of gradient intensity will be very large. The algorithm needs to normalize the gradient intensity to make further compression of illumination, shadows and edges.

   The normalization step is performed using equation:

$$v \to \frac{v}{\sqrt{\|v\|_2^2 + \varepsilon^2}} \tag{13}$$

where v is the non-normalized vector containing all histograms in a given block, $\|v\|_2$ is the L_2-norm of the descriptor vector (v), and ε is a small constant which is mainly introduced to avoid possible division by zero.

6) Extracting the eigenvector: Concatenating the HOG features of the 105 blocks contained in a window forms a 3780 dimensional HOG description X of a window. The X is the eigenvector of the window, which is used for the final classification. The visualization of HOG is shown in Figure 3.

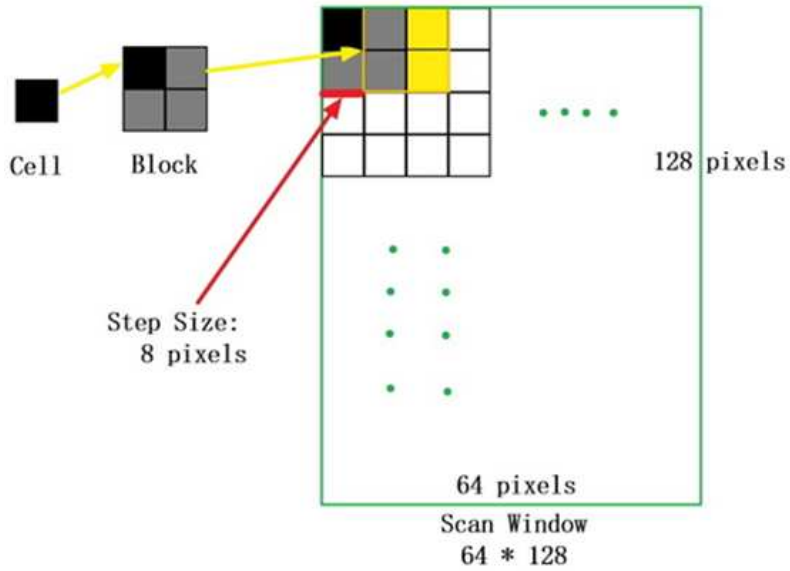

FIGURE 1. The division of small cells and large blocks

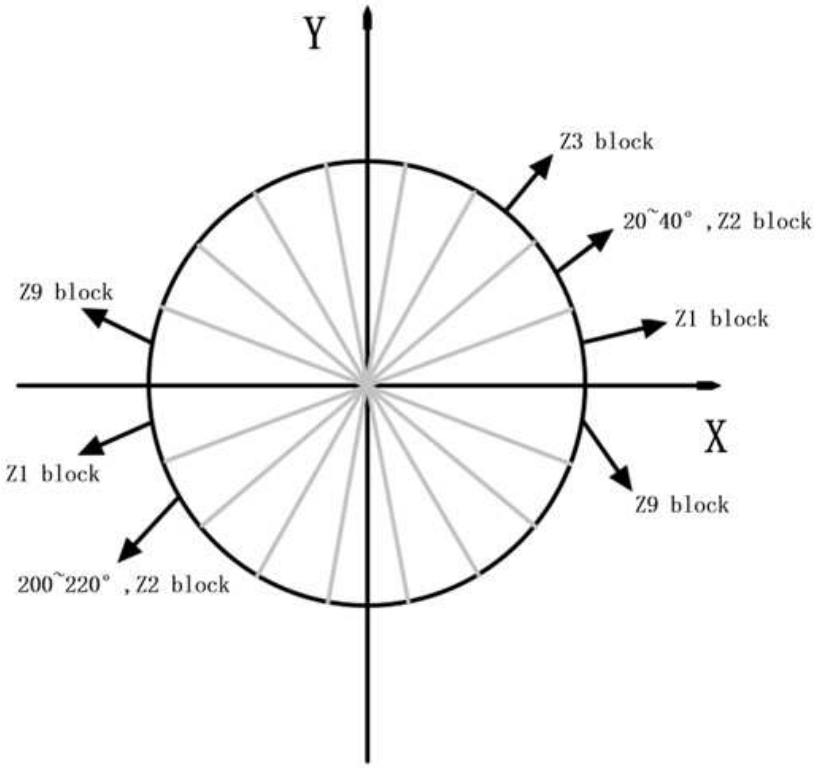

FIGURE 2. The partition of the HOG gradient orientation

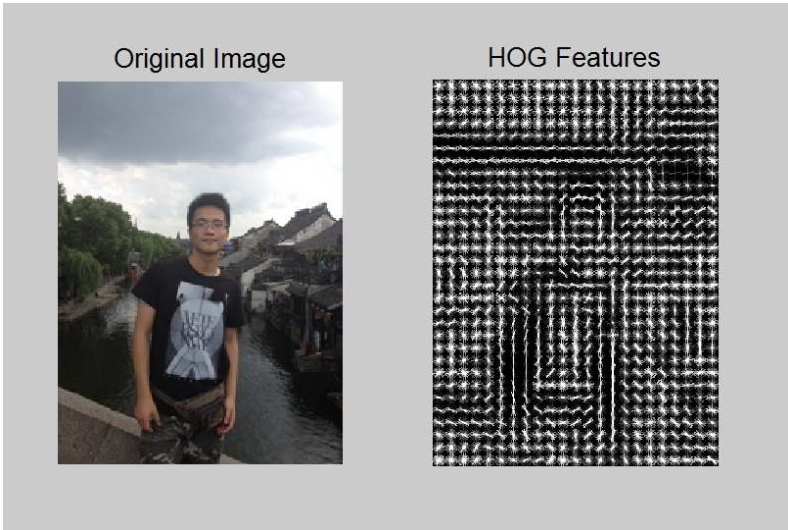

FIGURE 3. The visualization of HOG

3.2. **The method of support vector machine.** Support vector machine or simply "SVM" for short, is a binary-class model, which can map the original finite-dimensional space into a high- or infinite-dimensional space. If the sample is nonlinear in the original input space, it can be linearly separable in a higher-dimensional space by nonlinear mapping in SVM.

The concepts in SVM contain the geometric margin and the optimal separating hyperplane. Any hyperplane can be described by the linear equation:

$$\omega^T x + b = 0 \tag{14}$$

Among it, $\omega = (w_1, w_2, w_3, \ldots, w_d)^T$ is the normal vector, which decides the direction of the hyperplane; b is the term of displacement, which decides the distance between the hyperplane and the origin.

Supposing that the hyperplane (ω, b) can correctly classify training samples. For $(x_i, y_j) \in D$, if $y_j = +1$, $\omega^T x_i + b > 0$. If $y_j = -1$, $\omega^T x_i + b < 0$, i.e.:

$$\begin{cases} \omega^T x_i + b \geq +1, & y_j = +1 \\ \omega^T x_i + b \leq -1, & y_j = -1 \end{cases} \tag{15}$$

As shown in Figure 4, the input vectors that lie on the boundary of the optimal separating hyperplane are called support vectors. And the sum of the distance between two heterogeneous support vectors in the direction perpendicular to the hyperplane is

$$\gamma = \frac{2}{\|\omega\|} \tag{16}$$

In order to find an optimal separating hyperplane, we should find the normal vector ω and the displacement b, which satisfy the constraints in Equation (15), making γ the largest, i.e.:

$$\begin{aligned} \min_{\omega, b} \quad & \tfrac{1}{2} \|\omega\|^2 \\ \text{s.t.} \quad & y_i \left(\omega^T x_i + b\right) \geq +1, \ i = 1, 2, \ldots, m \end{aligned} \tag{17}$$

Above are basic formulas of the support vector machine.

There is also an important concept in the support vector machine, i.e., kernel function. The previous assumptions are based on training samples which are linearly separable, i.e., there is a hyperplane that can correctly classify training samples. However, in our real life, there are many nonlinear samples in the original space and there is no hyperplane

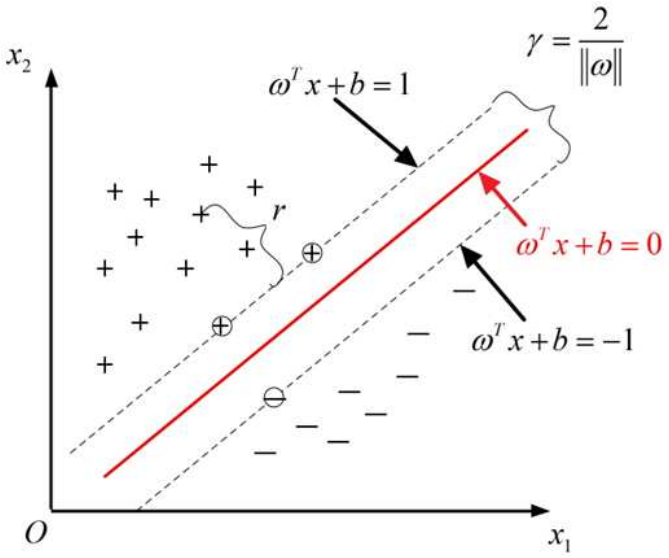

FIGURE 4. Support vector machine

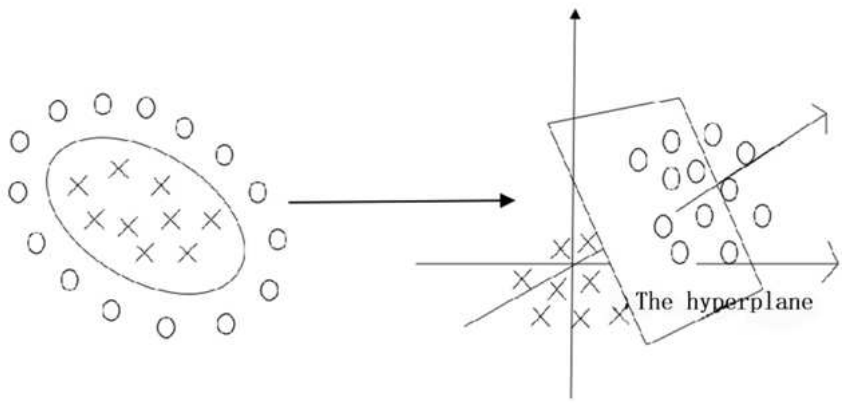

FIGURE 5. Nonlinear mapping

that can correctly classify two types of samples. As shown in Figure 5, the data itself in the plane is nonlinear, but for these data, we can map the samples in the original space into a higher dimensional feature space. In this way, samples can be linearly separable in this feature space. By introducing a kernel function $K(.,.)$ in SVM, the data can map into a high-dimensional space to solve the nonlinear problem in the original space.

The last step of the HOG-based pedestrian detection is to use the HOG feature vector as an input signal to SVM. In a fixed-size trial image, the trained linear SVM is used to calculate the vector descriptor, which can determine whether there are pedestrians. However, due to the large number of detection windows, once the video pixels go up, the detection speed will be very slow. It cannot achieve real-time, so we should improve it.

4. **The Proposed Scheme.** Aiming at the problem that the HOG+SVM algorithm of pedestrian detection is not real-time and there exists false detection, our scheme improves the algorithm. First of all, because of the impact of shadow effects, our scheme removes shadows for the video image and then takes advantage of the method using Gaussian mixture models to extract moving regions from video. Therefore, pedestrian detection processes are performed only within these regions, avoiding exhaustive sliding window search across the entire test image. Meanwhile, our scheme takes consideration of the

phenomenon – The extracted moving regions are not complete. Our scheme respectively performs operations of eroding and dilating and border expanding. Finally, the HOG feature of the extraction zones is calculated and then sent into the SVM classifier. The pedestrian detection algorithm can be divided into the following stages as Figure 6.

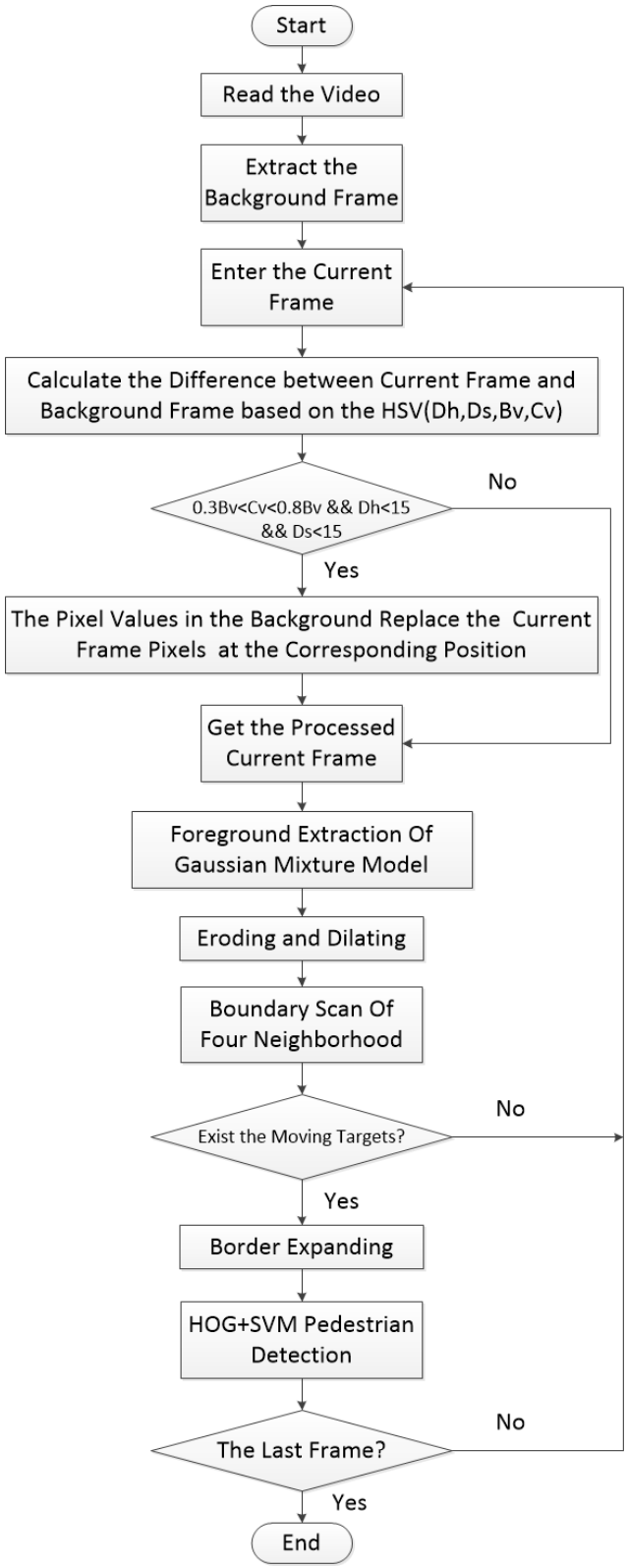

FIGURE 6. Flow chart of the algorithm framework

The shadows of objects in the video surveillance can affect the efficiency of pedestrian detection. When the shaded area is very large, the extracted moving regions using Gaussian mixture models will be larger. It will increase the detection time of the HOG+SVM algorithm. According to the characteristic that the background of the video surveillance will remain the same, our scheme adopts an algorithm based on the HSV color space to remove shadows. The HSV is more matched with human vision. It is clearer than RGB in distinguishing color brightness of the pixel. When the pixel is covered by shadows, the color brightness of the pixel will become darker and its saturation will become smaller. From this feature, we can accomplish the image processing of removing shadows. Effects of an example are listed in Figures 7-9.

Figure 7. Original image in the video

Figure 8. The detected moving target using GMMs before removing shadows

Figure 9. The detected moving target using GMMs after removing shadows

Figure 10. The algorithm without eroding and dilating

Figure 11. The algorithm with eroding and dilating

In the GMM algorithm, although it can achieve foreground detection, the effects of the foreground extraction will be poor, that is, the contour of object is sparse and there are many holes in the foreground. By contrast, the algorithm adding the process of eroding and dilating can get the fuller object and suppress noise more effectively.

In consideration of a fast and accurate contour search, our paper selects 4-neighborhood searching algorithm. The 4-neighborhood searching algorithm is very effective for the perfect foreground extraction. However, the foreground extraction of GMM algorithm is not perfect, and the results are shown as the following.

From Figures 12-15, we can see that the imperfect foreground extraction using 4-neighborhood searching algorithm (excluding the interference of small pixels) causes the separation of the head and body, which results in errors in the target detection. In order to avoid such errors, our scheme combines with characteristics of the normal pedestrian posture and morphological and then formulates a set of rules in the border merging and border deletion (Rule No.1: One border is small, and the other is large. The small border is directly above the large one, and they are very close together. They will be merged; Rule No.2: One border is small, and the other is large. The large border is directly above the small one, and they are very close together. They will be merged; Rule No.3: One border is small, and the other is large. The large border completely contains the small one. The small border is deleted). In this way, our scheme can divide the border more accurately.

In moving regions, we also should consider that the foreground extraction is not complete and some small moving areas will be lost, such as the head, and the foot. It will

FIGURE 12. One example image after foreground extraction

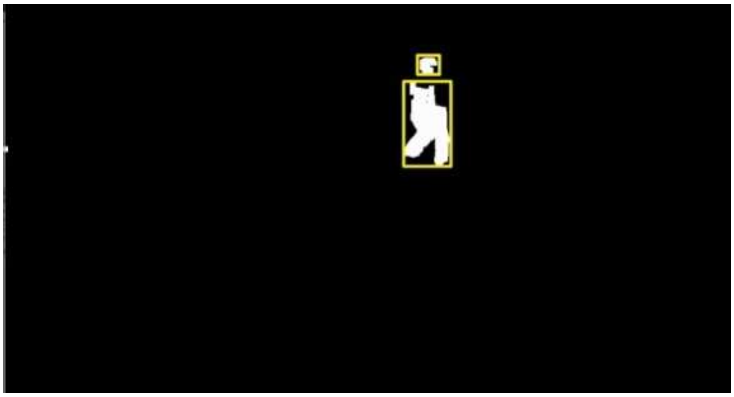

FIGURE 13. The 4-neighborhood searching result of Figure 12

FIGURE 14. Another example image after foreground extraction

subsequently cause a great interference in the HOG+SVM algorithm. The algorithm cannot effectively detect pedestrians in moving object regions. Therefore, our algorithm will make appropriate adjustments in the size of borders. The border can contain entire moving objects after this operation. As shown in Figures 16 and 17, gray borders are the extracted moving areas, and black borders are the moving areas after adjusting.

When the pedestrian detection in moving regions uses the HOG+SVM algorithm, our scheme selects the more popular INRIA pedestrian dataset as the train dataset to train HOG features. Our scheme selects 4000 images (64×128 pixels) as positive samples and

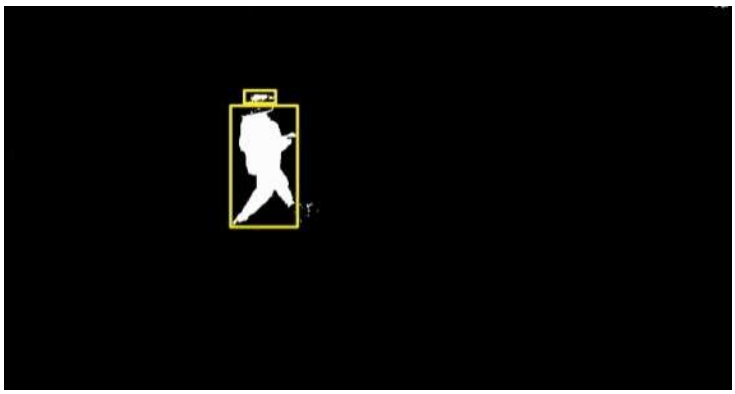

FIGURE 15. The 4-neighborhood searching result of Figure 14

FIGURE 16. Border expanding

FIGURE 17. Border expanding

2000 (64×128 pixels) images as negative samples from the INRIA dataset. Based on this operation, the HOG feature is extracted. And then in the sliding window of the moving area, the HOG feature and support vector machine are used to determine whether there are pedestrians.

Comparing Figure 18 and Figure 19, Figure 20 and Figure 21, there are some false detections and missing inspections in the pedestrian detection using HOG+SVM algorithm. Because inevitably the areas of non-pedestrian may be similar to the areas of pedestrian in the video scene, multi-scale detection needs to scan the whole picture, so the false-positive rate will be high and the detection rate will be low. Comparing Figure

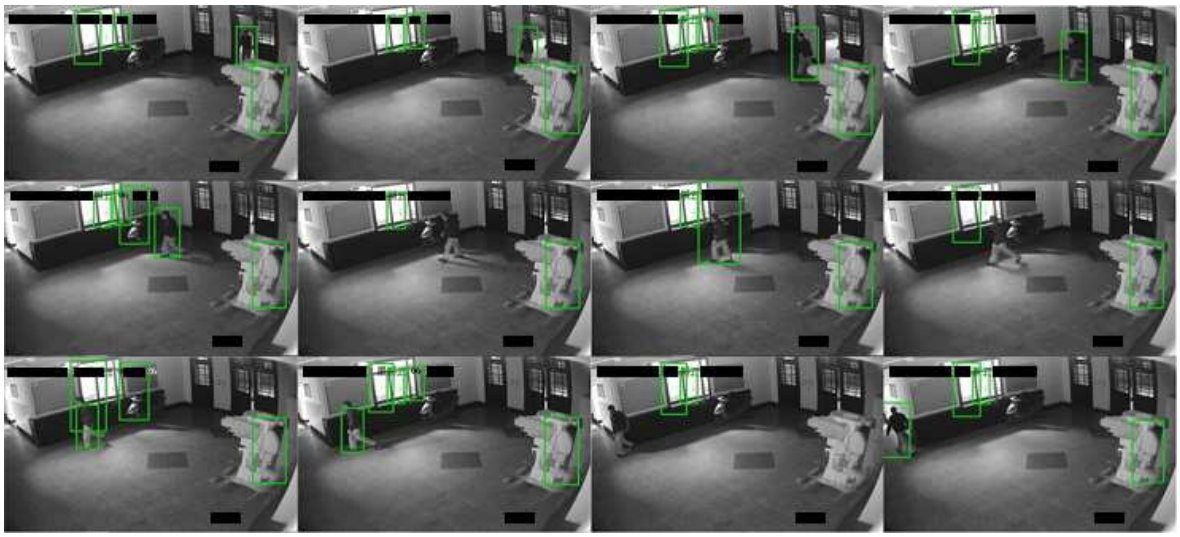

FIGURE 18. The detection result of the first video with a single person by using the HOG+SVM algorithm

FIGURE 19. The detection result of the first video with a single person by using our algorithm

22 with Figure 23, the effect of the new algorithm is very good in the complex pedestrian detection. It is much lower than the original HOG+SVM algorithm in the false-positive rate. Our algorithm has good detecting effects on the front, back and sides of pedestrians in the pedestrian detection.

Our scheme use the accuracy and false positive rate as evaluation indexs of the system as shown in Table 1. The accuracy rates between HOG+SVM algorithm and our algorithm are approximately equal. However, the false positive rates of our algorithm are much lower than those of the HOG+SVM algorithm. So compared with the HOG+SVM, our algorithm has been greatly improved in the stability and accuracy.

The biggest characteristic of our algorithm is to enhance the speed of the pedestrian detection using the HOG+SVM algorithm and enhance the real-time performance. Figure 24 shows the time-consuming case of processing three different videos by using our algorithm and the HOG+SVM algorithm respectively. From Figure 24, we can see that the

TABLE 1. The comparison of experimental data between HOG+SVM algorithm and our algorithm

V_n	N_r	HOG+SVM				Our algorithm			
		Test results	A_r (%)	False results	F_r (%)	Test results	A_r (%)	False results	F_r (%)
1	194	137	70.61	437	76.13	131	67.53	0	0
2	186	145	77.96	352	70.82	159	85.48	3	1.85
3	431	295	68.45	163	35.59	290	67.29	8	2.68
Sum	811	577	71.15	952	62.26	580	71.52	11	1.86

Note: V_n: Video number, N_r: The actual number of pedestrian, A_r: Accuracy rate, F_r: False positive rate

FIGURE 20. The detection results of the second video with a single person by using the HOG+SVM algorithm

FIGURE 21. The detection results of the second video with a single person by using our algorithm

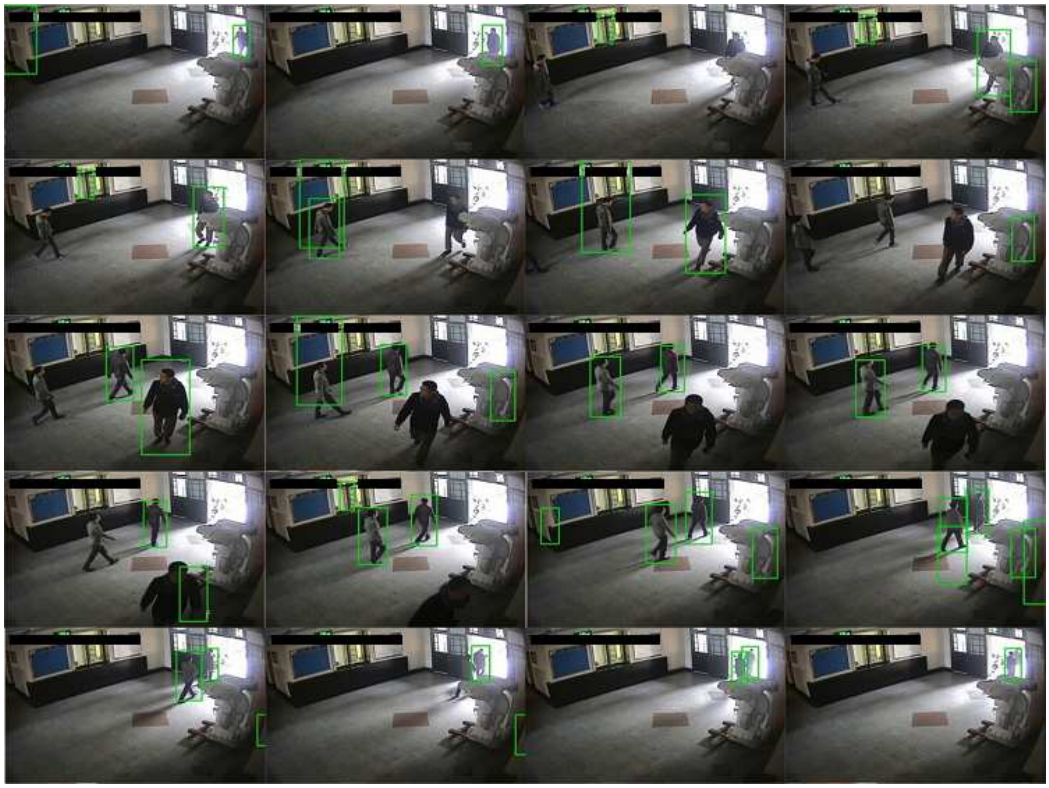

FIGURE 22. The detection results of the third video with multiple persons by using the HOG+SVM algorithm

FIGURE 23. The detection results of the third video with multiple persons by using our algorithm

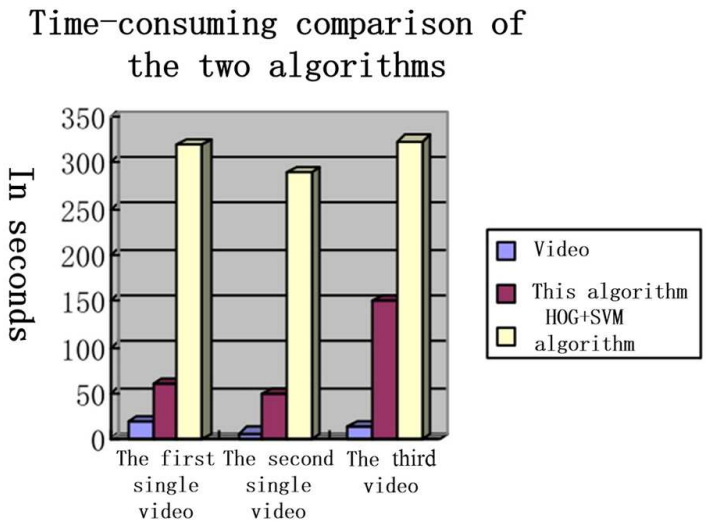

FIGURE 24. Time-consuming comparison of the two algorithms

number of pedestrians in the video is less, and the effect of pedestrian detection is more excellent by our algorithm. When the number of the pedestrians is increasing, the speed of pedestrian detection using our algorithm is still faster than HOG+SVM algorithm. However, compared to the single video, the speed will decline.

5. **Conclusions.** In order to improve the detection efficiency and accuracy of traditional HOG+SVM algorithm, this paper proposes a pedestrian detection method – a combination of modified dynamic background using Gaussian mixture models and the HOG-SVM detection algorithm. The method utilizes the techniques of removing shadows and eroding and dilating to further perfect the foreground extraction using Gaussian mixture models. Then the border of moving regions will be extracted by using the 4-neighborhood searching algorithm and border expanding. Finally, pedestrians are located in the border of moving regions by using the HOG+SVM algorithm. Compared with the traditional pedestrian detection, our algorithm greatly reduces the computational complexity, false-positive rate and the real-time performance has also been improved.

Acknowledgment. This work was partly supported by the Natural Science Foundation of Guangdong Province (2015A030310172), the Science & Technology Plan Projects of Shenzhen (JCYJ20150324140036830, GJHZ20160226202520268), and Zhejiang Provincial Natural Science Foundation of China under Grants No. LY15F010003 and No. LY17F030008. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] C. Papageorgiou and T. Poggio, A trainable system for object detection, *International Journal of Computer Vision*, vol.38, no.1, pp.15-33, 2000.

[2] B. Wu and R. Nevatia, Detection of multiple, partially occluded humans in a single image by Bayesian combination of edgelet part detectors, *The 10th IEEE International Conference on Computer Vision*, vol.1, pp.90-97, 2005.

[3] P. Sabzmeydani and G. Mori, Detecting pedestrians by learning Shapelet features, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8, 2007.

[4] D. M. Gavrila, A Bayesian, exemplar-based approach to hierarchical shape matching, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.29, no.8, pp.1408-1421, 2007.

[5] W. Ouyang and X. Wang, Joint deep learning for pedestrian detection, *IEEE International Conference on Computer Vision*, pp.2056-2063, 2013.

[6] C. Szegedy, W. Liu, Y. Jia et al., Going deeper with convolutions, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-9, 2014.

[7] Z. Wu, Z. Yu, J. Yuan and J. Zhang, A twice face recognition algorithm, *Soft Computing*, vol.20, no.3, pp.1007-1019, 2016.

[8] Z. Wu, J. Yuan, J. Zhang and H. Huang, A hierarchical face recognition algorithm based on humanoid nonlinear least-squares computation, *Journal of Ambient Intelligence and Humanized Computing*, vol.7, no.2, pp.229-238, 2016.

[9] P. F. Felzenszwalb, R. B. Girshick, D. Mcallester and D. Ramanan, Object detection with discriminatively trained part-based models, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.32, no.9, pp.1627-1645, 2010.

[10] C. Szegedy, S. Reed, D. Erhan, D. Anguelov and S. Ioffe, Scalable, high-quality object detection, *Computer Science*, 2015.

[11] N. Dalal and B. Triggs, Histograms of oriented gradients for human detection, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.1, no.12, pp.886-893, 2005.

[12] C. Stauffer and W. E. L. Grimson, Adaptive background mixture models for real-time tracking, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol.2, 1999.

[13] T. Bouwmans, F. El Baf and B. Vachon, Background modeling using mixture of Gaussians for foreground detection: A survey, *Recent Patents on Computer Science*, vol.1, no.3, pp.219-237, 2008.

[14] W. Li, H. Su, P. Feng, Q. Gao and P. Quan, A fast pedestrian detection via modified HOG feature, *Control Conference*, pp.3870-3873, 2015.