International Journal of Innovative Computing, Information and Control
Volume **15**, Number **6**, December **2019**

ICIC International ©2019 ISSN 1349-4198
pp. **2305–2320**

FAST PEDESTRIAN DETECTION AND TRACKING BASED ON VIBE COMBINED HOG-SVM SCHEME

Lang Wang[1], Jiaqi Gui[2], Zhe-Ming Lu[2,*] and Cong Liu[1]

[1]School of Information Science and Engineering
Ningbo Institute of Technology, Zhejiang University
No. 1, South Qianhu Road, Ningbo 315100, P. R. China
langwang1980@aliyun.com

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

Received March 2019; revised July 2019

Abstract. *Taking into account the problem of low speed of pedestrian detection with a HOG-SVM detector, this paper proposes a modified algorithm according to the characteristic of the video surveillance. The first step is using the ViBe method to extract the foreground objects zone in the video. However, the shadows of objects can affect the efficiency of pedestrian detection, so this paper removes shadows for each frame before the ViBe method. Then, three steps, i.e., eroding and dilating, 4-neighborhood searching algorithm and border expanding, are performed to make further alterations to the extracted foreground. At the same time, the Histogram of Oriented Gradients (HOG) feature of the extracted zone is calculated and then sent into the Support Vector Machine (SVM) classifier to judge whether there are pedestrians or not. The last step is using a template matching technique to further track detected pedestrians. Experimental results indicate that the proposed method outperforms the traditional HOG+SVM and GMM+HOG+SVM algorithms in terms of both recognition accuracy and processing speed.*
Keywords: ViBe method, Eroding and dilating, Template matching, Histogram of oriented gradients, Support vector machine, Gaussian mixture model, Pedestrian detection

1. **Introduction.** The main task of pedestrian detection is to spot the dynamic pedestrians from video sequences. With the development of computer vision, pedestrian detection has been widely used in the fields such as intelligent auxiliary driving, intelligent monitoring, pedestrian analysis and intelligent robot. However, because of the complexity of the real-world background, the diversity of pedestrian postures and the diversification of shooting angles, there is a big challenge for us to extract pedestrians from the input video fast and efficiently. In this way, pedestrian detection has been a very hot topic in the research area of computer vision. At present, pedestrian detection methods are mainly classified into two types: traditional pedestrian detection methods and machine-learning based pedestrian detection methods. Machine learning is the mainstream scheme for pedestrian detection currently. It mainly uses image features such as edge, shape and color information in static images to describe the pedestrian regions. Among them, some features can be used to detect pedestrians well, like the Haar wavelets based feature [1], the HOG based feature, the Edgelet based feature [2], the Shapelet based feature [3] and the shape contour template based feature [4]. Papageorgiou and Poggio [1] described an object class based on the Haar wavelet transform. They implicitly derived a model of an

DOI: 10.24507/ijicic.15.06.2305

object class by training a support vector machine classifier using a large set of positive and negative examples. Wu and Nevatia [2] modeled an individual human as an assembly of natural body parts. They used a new type of silhouette oriented features called edgelet features to learn part detectors by a boosting method. Sabzmeydani and Mori [3] learned the shapelet features based on local regions of the image to discriminate between pedestrian and non-pedestrian classes. They used AdaBoost to create these shapelet features and train the final classifier. Gavrila [4] used a template tree to efficiently represent and match the variety of shape exemplars. He adopted a Bayesian model to estimate the a posteriori probability of the object class, after a certain match at a node of the tree. In recent years, a new kind of 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]. Deep learning is a new field in machine learning. Lecun et al. [10] were the first authors to use CNN on detecting pedestrians and proposed an unsupervised deep learning method. Felzenszwalb et al. proposed the Deformable Part Model (DPM) which combined with a bunch of generated stochastic neural networks [11]. They both proved that there is great potential for deep learning algorithm in pedestrian detection [12].

Target tracking is to locate the individual or multiple specific objects of interest in real time and get accurate motion status. The appearance, contour, position and motion state of the moving object have good stability and similarity in adjacent video frames. The target and its surrounding background have some differences in the appearance of the image. According to these basic conditions, the tracking algorithm extracts features that describe the appearance of the target, or establishes a target model that is distinct from the background. Target tracking algorithms can be mainly categorized into active contour model based tracking, feature-based tracking, region-based tracking, and model-based tracking, and so on. They have their own advantages when compared with the machine-learning-based methods: they are usually of low computation overload and do not need to collect large amount of pedestrian or non-pedestrian samples [13].

The feature named Histogram of Oriented Gradients (HOG) is the main concern in our paper, which was proposed by Dalal and Triggs [14] in 2005. And they combined the HOG feature with the SVM classifier to achieve breakthroughs in the field of pedestrian detection. The HOG feature based method densely extracts the local histogram of oriented gradients in the image window, which can fully extract pedestrian shape information and appearance information. It has excellent discrimination to distinguish between pedestrians and other objects. However, computing HOG features requires intensive and complex scans, which greatly causes the high computational complexity and poor real time performance.

The HOG feature combined with the SVM classifier has been widely used in image recognition, especially in pedestrian detection. There are many pedestrian detection algorithms constantly proposed, but basically based on the idea of HOG+SVM. However, the detection and tracking speed of the original HOG+SVM method is low, and the false detection rate and missed detection rate of the original HOG+SVM method are high. According to the real time requirement of the video surveillance, this paper proposes a modified algorithm to improve the speed of pedestrian detection by using HOG features. Firstly, our paper removes shadows for each frame and then makes full use of the foreground detection algorithm (ViBe) to extract moving objects. The reason why we remove shadows is that, in indoor monitoring scenes, shadows may occur due to factors such as occlusion and uneven illumination between moving human bodies. The shadow will be mistaken for the target, and will have a negative impact on subsequent tracking and behavior recognition. Therefore, in the moving target detection phase, the shadow portion

should be removed. Next, through border expanding, all the moving objects can be completely contained in the scanning region. Then the detection in the foreground is guided by the HOG feature and SVM classifier. Here, SVM is widely used in pattern recognition in various fields [15], including face recognition, text categorization, handwriting recognition, bioinformatics, and so on. Finally, the template matching method is used to track the detected pedestrians. Experimental results demonstrate the effectiveness and veracity of our algorithm.

The paper is organized as follows. Section 2 briefly introduces the ViBe method. Section 3 briefly introduces the HOG+SVM algorithm. The fast pedestrian detection and tracking algorithm based on our ViBe combined HOG-SVM scheme is presented in Section 4. Section 5 presents the results of our experiments. Finally, Section 6 concludes this paper.

2. **ViBe Method.**

In our scheme, the ViBe algorithm [16] is utilized to achieve the initial foreground region of a moving object. ViBe is a fast background modeling method for foreground detection based on probability statistics and it stores a sample set for each pixel, where the sample elements include the pixel itself and its neighboring pixels. And then each new pixel is compared with the sample set to determine whether it belongs to the background or not. The core of the method includes four modules, i.e., model initialization, foreground detection, model update and exception handling [17].

2.1. **Model initialization.**

The ViBe algorithm mainly uses the single frame of the video sequence to initialize the background model. For each pixel, the gray values of N pixels are randomly selected within its eight-neighborhood pixels and stored in N samples corresponding to the ViBe model, that is, the model is initialized only by the first frame. The second frame starts to perform foreground extraction.

2.2. **Foreground detection.**

The foreground detection process includes two steps: the first step is to compare with the N samples in the model, to see whether the current pixel matches the background model or not; the second step is to count the number of the matched samples.

Step 1. Starting from the second frame, each new pixel is compared with its corresponding N pixel samples from the previous frame. Specifically, the method compares the absolute of the gray value difference with the pre-set threshold R. If it is less than the threshold, it means finding a match.

Step 2. Count the number of matched pixels $n_t(x, y)$, and then compare it with the pre-set threshold $\#\min$ (The minimum matching number). If the cumulative matching number is less than $\#\min$, it indicates that the pixel is in the foreground; if the matching number is larger than or equal to $\#\min$, it indicates that the pixel belongs to the background, as shown in Equation (1)

$$M_t(x, y) = \begin{cases} 1, & n_t(x, y) < \#\min \\ 0, & n_t(x, y) \geq \#\min \end{cases} \tag{1}$$

where $M_t(x, y) = 1$ means that the point (x, y) is judged as the foreground point at time t, and $M_t(x, y) = 0$ means that the point is judged as the background point. Therefore, the final foreground detection result is a binarized result.

2.3. **Model update.**

For the pixels that are determined to be the background, their ViBe models and the ViBe models of their neighborhood pixels need to be updated. The ViBe algorithm proposes a model updating method. The main idea of the model updating process is described as follows.

(1) Set the updating probability ϕ, i.e., when a pixel is determined as the background, the probability of this point to update the neighbor sample points is $1/\phi$, and the probability of it to update itself is also $1/\phi$. The next step is to randomly select a sample from the model and replace the sample with the gray value of this pixel.

(2) Update the ViBe model of its neighborhood. Firstly, we randomly select a pixel within the 8-neighbor of it and replace the value of the selected neighbor pixel with its gray value. Then we should update the ViBe model of the selected neighbor pixel.

2.4. **Exception handling module.** If the number of the pixel in a certain position which is continuously determined as the foreground exceeds a specific threshold, the pixel is directly determined as the background. Finally, we should clear the count of this pixel.

The above method has been paid more and more attention because of its simple and fast characteristics.

3. **Pedestrian Detection Based on the HOG+SVM Algorithm.** Histogram of Oriented Gradients (HOG) is a feature descriptor used in computer vision and image processing for object detection. The technique counts occurrences of gradient orientation in localized portions of an image. 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 important step in the HOG algorithm is gradient calculation, which calculates the gradient direction of each pixel position by calculating the abscissa and ordinate gradient of the image. The main purpose is to weaken the interference of light any further, and capture the contour information. Gradient calculation can be conducted by following equations [18]:

$$G_x(x,y) = H(x+1,y) - H(x-1,y) \tag{2}$$

$$G_y(x,y) = H(x,y+1) - H(x,y-1) \tag{3}$$

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. The next step is to calculate the gradient magnitude and direction by using the following equations:

$$G(x,y) = \sqrt{G_x(x,y)^2 + G_y(x,y)^2} \tag{4}$$

$$\alpha(x,y) = \tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right) \tag{5}$$

Then, the orientation histogram from the orientations and magnitudes is derived at each cell. In our scheme, each cell size is of 8×8 pixels and the orientation histogram has 9 bins with each cell.

Due to changes of local illumination and the contrast radio 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 the following equation:

$$v \to \frac{v}{\sqrt{\|v\|_2^{\,2} + \varepsilon^2}} \tag{6}$$

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.

Finally, each block size is of 2×2 cells (The picture is of size 64×128), so it will produce 105 blocks. Concatenating the HOG features of the 105 blocks contained in a window

forms a 3780-dimensional HOG description X of a window. X is the eigenvector of the window, which is used for the final classification.

The method of 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, as shown in Figure 1.

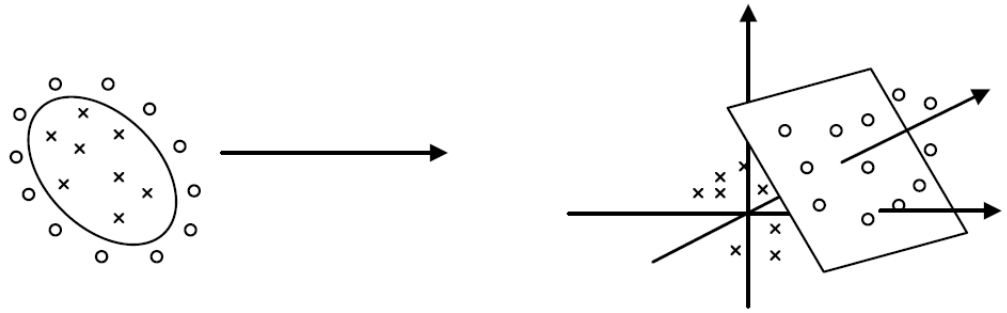


FIGURE 1. Nonlinear mapping

However, the dimension of non-linear mapping can truly be very large in practice. For example, in the case of document classification, one may wish to use as features sequences of three consecutive words, i.e., trigrams. Thus, with a vocabulary of just 100000 words, the dimension of the feature space reaches 10^{15}, so it can become very costly. A solution to this problem is to use kernel methods, which are based on kernels or kernel functions.

The kernel K is performed using the following equation:

$$K(x, x') = \langle \Phi(x), \Phi(x') \rangle \quad \forall x, x' \in \mathrm{X} \tag{7}$$

We will see several common examples where the computation of $K(x, x')$ can be achieved in $O(N)$ while that of $\langle \Phi(x), \Phi(x') \rangle$ typically requires $O(\dim(\mathrm{H}))$, with $\dim(\mathrm{H}) >> N$. So the kernel K is efficient.

The last step of 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.

As a kind of important algorithms in the field of target tracking, the target tracking algorithm based on template matching [19] has been paid more and more attention because of its accuracy and practicability. The basic idea of this algorithm is to use the target information and characteristics in the video to establish the target template. And then the image in each frame is compared against the target template to search for the target. The last step is to obtain the motion state estimation of the target. The basic steps of the whole algorithm are shown in Figure 2.

As shown in Figure 2, the target tracking algorithm based on template matching mainly includes three steps: template establishment, matching & tracking and template update. The input of the algorithm is an image in the video, and the output is the tracking result of the input image. The template establishment belongs to the initialization phase, the main body of the algorithm is matching & tracking and the template update is the link to maintain the whole target tracking process.

In this paper, we choose the normalized squared difference matching method, which determines the match degree by normalizing the square sum of the gray value difference

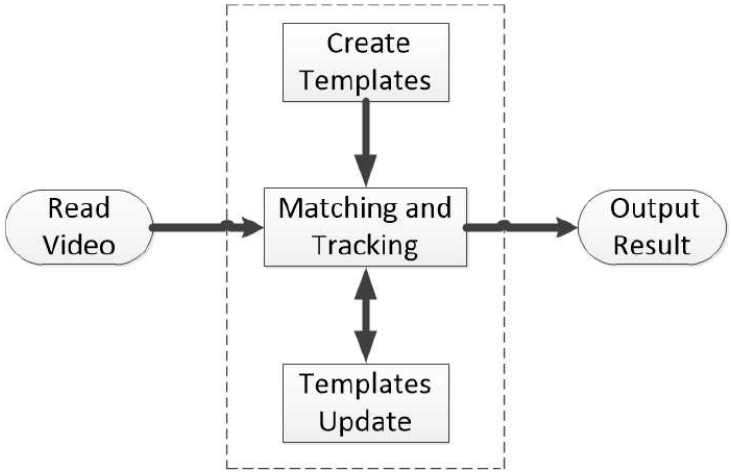


FIGURE 2. The basic steps of target tracking algorithm based on template matching

of the template image and the image to be matched. We set the size of the template M is $I_X \times I_Y$, and the size of the image to be matched P is $J_X \times J_Y$. When the algorithm is implemented, the template is translated on the image to be matched. The overlapped area under the template is set to $S(x, y)$. (x, y) is the coordinate position of the lower right corner of the S in the graph P.

The normalized squared difference for this matching method is defined as follows:

$$R(x,y) = \frac{\sum_{i=1}^{I_X}\sum_{j=1}^{I_Y}[M(i,j) - S^{x,y}(i,j)]^2}{\sqrt{\left(\sum_{i=1}^{I_X}\sum_{j=1}^{I_Y}[S^{x,y}(i,j)]^2\right)}\sqrt{\left(\sum_{i=1}^{I_X}\sum_{j=1}^{I_Y}[M(i,j)]^2\right)}} \tag{8}$$

The smaller the value of $R(x, y)$ in the above formula is, the more similar to the template. When the value of $R(x, y)$ is 0, it indicates that we find the best match at that position. However, in the actual system, it is almost impossible for the template to be exactly the same as the overlapped area under the template. So $R(x, y)$ has a very small probability to be 0, we regard the position with the minimum value of $R(x, y)$ as the target position.

4. **Fast Pedestrian Detection and Tracking Based on ViBe Combined HOG-SVM Scheme.** Aiming at the problem that the HOG+SVM algorithm of pedestrian detection is not real time and has false detection, our scheme is presented to improve the algorithm. First of all, because of the impact of shadow effects, our scheme removes shadows for the video frame and then takes advantage of the ViBe method, eroding and dilating, 4-neighborhood searching algorithm and border expanding to extract moving regions from the video frame. Therefore, pedestrian detection processes are performed only within these regions, avoiding exhaustive sliding window search across the entire frame. Next, the HOG feature of the extraction zones is calculated and then sent into the SVM classifier. Once the pedestrian is detected, the template matching method is used for tracking the detected pedestrian.

The specific idea of the HOG+SVM algorithm combined with the template matching method is as follows. Firstly, our scheme uses the HOG+SVM algorithm to detect pedestrians in the extraction zones. If there is no pedestrian, we repeat following steps:

removing shadows, the ViBe method, eroding and dilating, 4-neighborhood searching algorithm and border expanding. Once the pedestrian is found, its position is recorded and a region of interest is calculated around it. Then, we detect pedestrians in this region of interest rather than the full image, which can speed up the detection process significantly.

The above method makes the algorithm fast and easy to implement, but it is still shitty as it fails when pedestrians rotate their bodies at the certain angle. So we use the template matching method to rescue this problem. If the HOG+SVM algorithm fails, the template

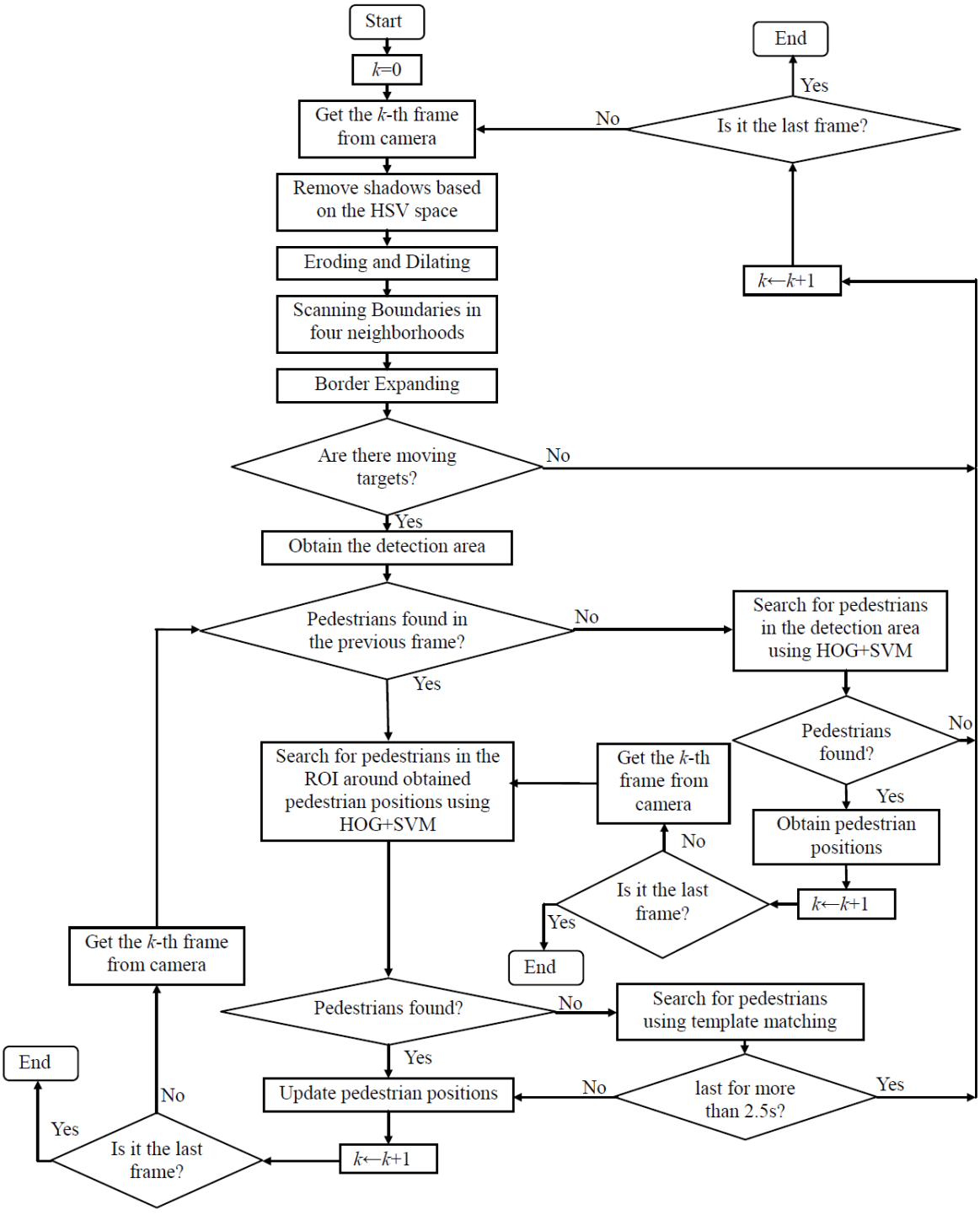


FIGURE 3. The flowchart of the proposed method

matching algorithm tracks the most likely position of pedestrians according to the last detected pedestrian templates. The template matching method makes the algorithm more reliable and tracks pedestrians quite well. However, there are two things which can make the template matching method fail. One is that we have detected pedestrians in this region of interest while the other is that the tracking window loses the pedestrian. Here, we need to introduce a timer. When the duration of the template matching is more than 2.5s, the tracking window is considered to lose the pedestrian. In the case of failure of template matching, we reinitialize pedestrian tracking with the slower ViBe+HOG+SVM algorithm over the complete frame. Figure 3 shows the flowchart of the proposed method, which can be illustrated in detail as follows:

Step 0: Set $k = 0$.

Step 1: Get the k-th frame from the camera.

Step 2: Preprocess the frame, including removing shadows based on the HSV space, eroding and dilating, scanning boundaries in four neighborhoods and border expanding. Then check whether there are moving targets in the preprocessed frame. If yes, obtain the detection area, go to Step 4. Otherwise, go to Step 3.

Step 3: k is updated by $k+1$, if k reaches the last frame, then terminate the algorithm. Otherwise, go to Step 1.

Step 4: If there are pedestrians found in the previous frame, go to Step 7. Otherwise, go to Step 5.

Step 5: Search for pedestrians in the detection area using HOG+SVM. If there are pedestrians found, obtain pedestrian positions, go to Step 6. Otherwise, go to Step 3.

Step 6: k is updated by $k+1$, if k reaches the last frame, then terminate the algorithm. Otherwise, get the k-th frame from the camera. Go to Step 7.

Step 7: Search for pedestrians in the ROI around obtained pedestrian positions using HOG+SVM. If there are pedestrians found, update pedestrian positions, go to Step 8. Otherwise, go to Step 9.

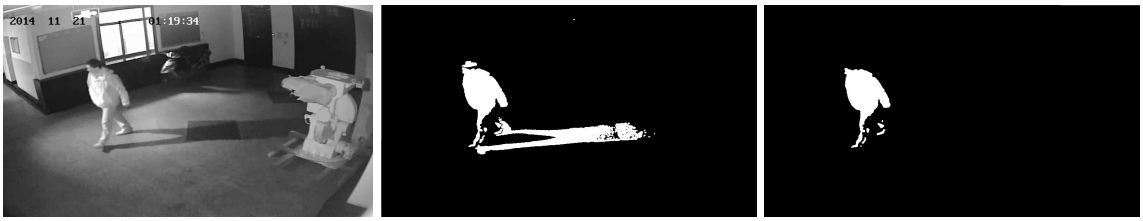
Step 8: k is updated by $k+1$, if k reaches the last frame, then terminate the algorithm. Otherwise, get the k-th frame from the camera. Go to Step 4.

Step 9: Search for pedestrians using template matching. If the template matching has lasted for more than 2.5s frames and no pedestrians are found, then go to Step 3. Otherwise, update pedestrian positions, go to Step 8.

5. **Experimental Results and Analysis.** The shadows of objects in the video surveillance can affect the efficiency of pedestrian detection. The extracted moving regions using the ViBe method will be larger when the shaded area is very large. So the detection time of the HOG+SVM algorithm will be increased. However, based on the characteristics of the video surveillance, our scheme adopts an algorithm based on the HSV color space to remove shadows. When the pixel of image is covered by shadows, the saturation of the pixel will become smaller and its color brightness will become darker. According to this feature, we can get rid of the shadows. Effects of an example are shown in Figure 4.

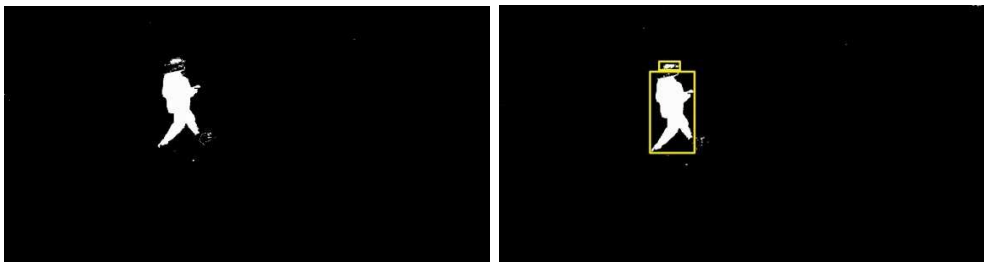
Our scheme can achieve foreground extraction using the ViBe method, but the effects will be poor, i.e., there are many holes in the foreground and the contour of the object is sparse. By contrast, the algorithm adding the process of eroding and dilating can get the fuller object and suppress noise more effectively.

In consideration of a fast and accurate contour search, our scheme selects 4-neighborhood searching algorithm. The 4-neighborhood searching algorithm is very effective for the perfect foreground extraction. However, the foreground extraction of the ViBe method is not perfect, and the results are shown in Figure 5.



(a) (b) (c)

FIGURE 4. Removing shadows: (a) original image; (b) the detected moving target before removing shadows; (c) the detected moving target after removing shadows



(a) (b)

FIGURE 5. The 4-neighborhood searching algorithm: (a) foreground extraction; (b) the 4-neighborhood searching result

From Figure 5, we can know 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 is harmful to target detection. In order to prevent the situation from getting worse, our scheme combines with characteristics of the normal pedestrian posture and morphological and then formulates a set of rules to merge or delete borders (Rule No.1: One border is small, 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.2: One border is small, 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.3: One border is small, 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 should also consider that the foreground extraction does not contain the whole person and some small moving areas will be lost, such as the head, and the foot. For this reason, the HOG+SVM algorithm cannot effectively detect pedestrians in moving object regions. Therefore, the size of borders will be adjusted appropriately in our algorithm. The entire moving objects will be within the border after this operation. As shown in Figure 6, white borders are the extracted moving areas, and gray borders are the moving areas after adjusting.

Our scheme selects six videos with three different scenes for experimental testing, of which five for the single-person video, one for the double-person video. At the same time, we make comparison among our algorithm, HOG+SVM algorithm and GMM+HOG+SVM algorithm. Here, GMM+HOG+SVM algorithm refers to taking advantage of the method using Gaussian Mixture Models to extract moving regions from video and then detecting pedestrians using HOG+SVM algorithm in the extraction zones. The difference between our algorithm and GMM+HOG+SVM algorithm is that our scheme extracts the

FIGURE 6. Border expanding

moving objects using the ViBe method and then tracks the detected pedestrians using the template matching. Here, with regard to SVM, we use *C_SVC* which means C-support vector classifier, the parameter C is the penalty coefficient, the larger C means the greater the penalty for misclassification, the appropriate parameter C is critical for classification accuracy, and we use $C = 0.01$ in this paper. We use the linear kernel, i.e., $K(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{x} \cdot \boldsymbol{z}$, which is the ordinary inner product. The termination condition is based on the maximum number of iterations 1000.

Part of the experimental comparison between the effects of our algorithm, the HOG+SVM algorithm and the GMM+HOG+SVM algorithm is given in Figures 7-18. From these results we can see that there are some false detections and missing inspections in the pedestrian detection using HOG+SVM algorithm. For example, for the first video, from the results shown in Figure 7 by the HOG+SVM algorithm, we can see that the machine in the right-bottom corner is falsely detected. From the results shown in Figure 8 by the GMM+HOG+SVM method, we can see that the pedestrian in some frames cannot be detected. From the results shown in Figure 9 by our method, we can see that the pedestrian in all frames are correctly detected. The main reason for the false detection

FIGURE 7. The detection results of the first video scene by using HOG+SVM algorithm

FIGURE 8. Detection results of the first video scene by GMM+HOG+SVM algorithm

FIGURE 9. The detection results of the first video scene by using our algorithm

is that inevitably the areas of non-pedestrian may be similar to the areas of pedestrian in the video scene. And the detection of the HOG+SVM algorithm needs to scan the whole picture, so the false-positive rate is high. The GMM+HOG+SVM algorithm basically solves the false detection of the HOG+SVM algorithm, but it still has the phenomenon of missing inspection. Our algorithm has good detecting effects on the front, back and sides of pedestrians in the pedestrian detection. It is much lower than the original HOG+SVM and GMM+HOG+SVM algorithms in the false-positive rate and false-negative rate.

Three different algorithms are performed on six videos and we use the accuracy and false alarm rate as evaluation indexes of the system as shown in Table 1. From Table 1, we can see that the average accuracy rate of our algorithm is 90.84%, while the HOG+SVM and the GMM+HOG+SVM algorithms are only about 80%. Meanwhile, as far as the false alarm rate is concerned, our algorithm is 0%, the GMM+HOG+SVM algorithm is 5.39%, while the HOG+SVM algorithm is up to 48.08%. So compared with the HOG+SVM and the GMM+HOG+SVM methods, our algorithm has greatly improved the stability and accuracy.

FIGURE 10. Detection results of the second video scene by using HOG+SVM algorithm

FIGURE 11. Detection results of the second video scene by using GMM +HOG+SVM algorithm

FIGURE 12. Detection results of the second video scene by using our algorithm

FIGURE 13. Detection results of the third video scene by using HOG+SVM algorithm

FIGURE 14. Detection results of the third video scene by using GMM +HOG+SVM algorithm

FIGURE 15. Detection results of the third video scene by using our algorithm

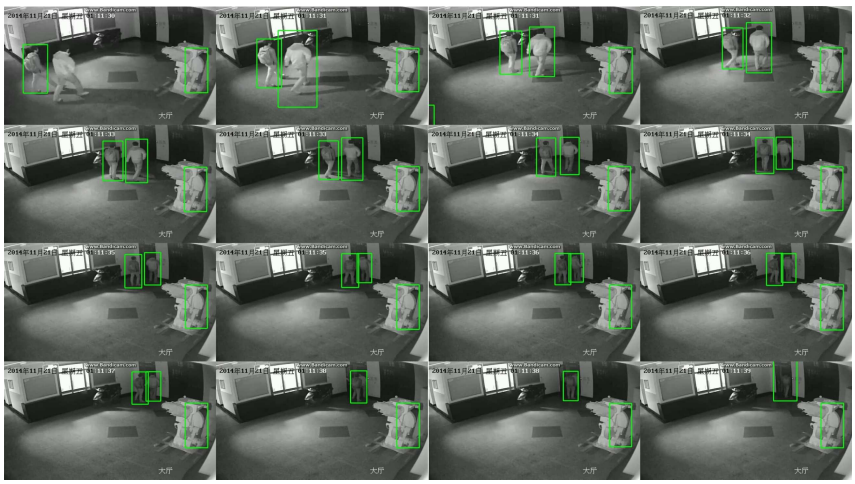


FIGURE 16. Detection results of the double-pedestrian video by HOG+SVM algorithm

Another advantage of our algorithm is that it can enhance the speed of the pedestrian detection, and some simple single-pedestrian videos can detect pedestrians in real time. Figure 19 shows the time-consuming performance of processing six different videos by using our algorithm, the HOG+SVM algorithm and the GMM+HOG+SVM algorithm respectively. From Figure 19, we can see that the speed of pedestrian detection using our algorithm is faster than the HOG+SVM algorithm and the GMM+HOG+SVM algorithm. However, when the number of the pedestrian increases, e.g., the fourth and sixth videos with two pedestrians, the speed will decrease.

6. **Conclusion and Future Work.** In this paper, we propose a fast pedestrian detection and tracking algorithm based on ViBe combined HOG+SVM scheme, which provides detection accuracy 10% higher than the HOG+SVM algorithm and the GMM+HOG+SVM

FIGURE 17. Detection results of the double-pedestrian video by using GMM+HOG+SVM

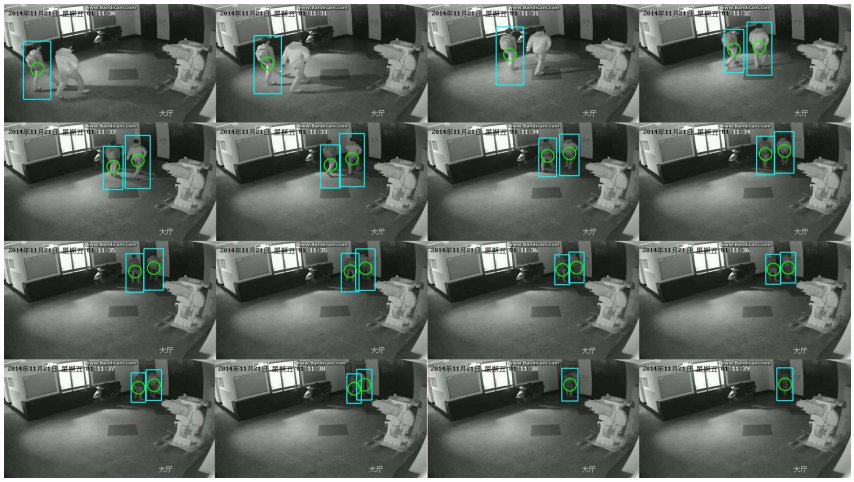


FIGURE 18. Detection results of the double-pedestrian video by using our algorithm

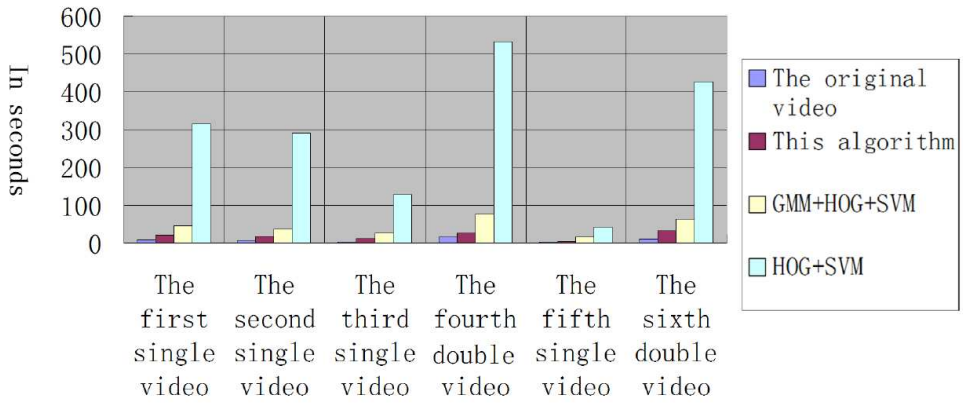


FIGURE 19. Time-consuming comparison of the three algorithms

algorithm. The method utilizes the techniques of removing shadows and eroding and dilating to further perfect the foreground extraction using the ViBe method. Then the border of moving regions will be extracted by using the 4-neighborhood searching algorithm and border expanding. Finally, pedestrians are positioned in the border of moving regions

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

Video number		1	2	3	4	5	6	Sum
NA		186	163	107	457	58	535	1506
HOG+SVM	ND	145	151	102	378	24	428	1228
	ACC (%)	77.96	92.64	95.35	82.71	41.38	80.00	81.54
	NFP	352	165	123	127	9	361	1137
	RFA (%)	70.82	52.22	54.67	25.15	27.27	45.75	48.08
GMM+HOG+SVM	ND	159	147	95	376	26	407	1210
	ACC (%)	85.48	90.18	88.79	82.28	44.83	76.07	80.35
	NFP	3	3	1	53	5	4	69
	RFA (%)	1.85	2.00	1.04	12.35	16.13	0.97	5.39
Our algorithm	ND	171	146	95	435	56	465	1368
	ACC (%)	91.94	89.57	88.79	95.19	96.55	86.92	90.84
	NFP	0	0	0	0	0	0	0
	RFA (%)	0	0	0	0	0	0	0

Note: NA: The number of annotated pedestrians, ND: The number of correctly detected pedestrians, ACC: Accuracy rate, NFP: The number of false alarms, RFA: False alarm rate, ACC = ND/NA, RFA = NFP/(NFP + ND)

by using the HOG+SVM combined with the template matching method. The algorithm can achieve pedestrian detection and tracking in real time. However, this algorithm has its defect, the accuracy of algorithm relies too much on the pedestrian detection in the extraction zone. If the pedestrian detection is wrong, a non-pedestrian template will be tracked in video sequence. It will be the future work about how to ensure that a correct pedestrian template is tracked.

Acknowledgements. This work was supported partially by the Zhejiang Provincial Natural Science Foundation of China under grant No. LY17F030008 and Natural Science Foundation of Ningbo under grant No. 2018A610165. This work was also partially supported by the financial support from Ningbo Public Welfare Science and Technology Plan Project under grant No. 2019C50026.

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, *Proc. of the 10th IEEE International Conference on Computer Vision*, vol.1, no.1, pp.90-97, 2005.

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

[4] D. M. Gavrila, A Bayesian exemplar-based approach to hierarchical shape matching, *IEEE Transactions on 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, *Proc. of IEEE International Conference on Computer Vision*, pp.2056-2063, 2014.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, Going deeper with convolutions, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp.1-9, 2015.

[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] C. Szegedy, S. Reed, D. Erhan, D. Anguelov and S. Ioffe, Scalable, high-quality object detection, *Computer Science*, arXiv: 1412.1441, 2015.

[10] P. Sermanet, K. Kavukcuoglu, S. Chintala and Y. Lecun, Pedestrian detection with unsupervised multi-stage feature learning, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, Portland, OR, USA, pp.3626-3633, 2013.

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

[12] H. Li, Z. Wu and J. Zhang, Pedestrian detection based on deep learning model, *Proc. of the 9th International Congress on Image and Signal Processing, Biomedical Engineering and Informatics*, Datong, China, pp.796-800, 2017.

[13] G. Wang, Q. Liu, Y. Zheng and S. Peng, Far-infrared pedestrians detection based on adaptive template matching and heterogeneous-feature-based classification, *Proc. of IEEE International Instrumentation and Measurement Technology Conference*, pp.1-6, 2016.

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

[15] C. Li, H. Zhang, H. Zhang and Y. Liu, Short-term traffic flow prediction algorithm by support vector regression based on artificial bee colony optimization, *ICIC Express Letters*, vol.13, no.6, pp.475-482, 2019.

[16] O. Barnich and M. Droogenbroeck, ViBe: A universal background subtraction algorithm for video sequences, *IEEE Transactions on Image Processing*, vol.20, no.6, pp.1709-1724, 2011.

[17] X. Li, S. Zhu, L. Chen and J. Liu, Target detection via improved ViBe algorithm, *Proc. of the 27th Chinese Control and Decision Conference*, pp.5930-5935, 2015.

[18] B. Leng, Q. He, H. Xiao, B. Li, H. Wang, Y. Hu, W. Wu, G. Guan, H. Zou and L. Liang, An improved pedestrians detection algorithm using HOG and ViBe, *Proc. of IEEE International Conference on Robotics and Biomimetics*, pp.240-244, 2013.

[19] J. S. Bae and T. L. Song, Image tracking algorithm using template matching and PSNF-m, *International Journal of Control Automation and System*, vol.6, no.3, pp.413-423, 2008.