Fast Pedestrian Detection Based on Sliding Window Filtering

Feidie Liang[1], Dong Wang[2], Yang Liu[2], Youcheng Jiang[1], and Sheng Tang[1]

[1] Adavanced Computing Research Laboratory, Beijing Key Laboratory of Mobile Computing
and Pervasive Device, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
{liangfeidie,jiangyoucheng,ts}@ict.ac.cn
[2] Huawei Technologies Co., Ltd, Beijing, China
{dave.wangdong,ethan.liuyang}@huawei.com

Abstract. Pedestrian detection is a fundamental problem in video surveillance. An overwhelming majority of existing detection methods are based on sliding windows with exhaustive multi-scale scanning over the whole frame images which can achieve good accuracy but suffer from expensive computational cost. To reduce the complexity significantly while keeping high accuracy, in this paper, we propose an effective and efficient pedestrian detection method based on sliding windows with well-designed multi-scale scanning over candidate regions instead of whole frames. The candidate regions can be obtained through three main steps: (1) foreground extraction by using a fast background subtraction model to remove large number of static regions since pedestrians are usually keeping moving; (2) region merging and filtering through clustering foreground pixels to avoid over-partitioned or too large regions of non-pedestrian; (3) well-designed multi-scale scanning by exploiting the size information of current region to avoid useless scales. Therefore, through utilization of motion and size information, we can not only speed up the detection through reducing large number of windows, but also improve the accuracy of detection through eliminating many false positive regions. Our experiments on two public datasets have verified that our method outperforms the state-of-the-art methods in both speed and accuracy of detection.

Keywords: Pedestrian detection, Video surveillance, Background subtraction, Sliding window.

1 Introduction

Pedestrian detection is a fundamental problem in video surveillance since it is a key procedure for successive tracking, action recognition, personal identification and abnormal events detection. In the last decade, most researches focused on improving the detection accuracy and made significant progresses. However, an assignable byproduct of the increased detection accuracy is the quicker increased computation cost, which makes the pedestrian detection further far from real-time processing. Especially in video surveillance, embedded computers are usually adopted to perform pedestrian detection thus the computation capability is very limited. Meanwhile, many

W. Lin et al. (Eds.): PCM 2012, LNCS 7674, pp. 811–822, 2012.

applications require real-time accurate pedestrian detection in video surveillance, so a big challenge is how to improve the speed of state-of-the-art methods without sacrificing the detection accuracy.

One of the most successful approaches for pedestrian detection is the sliding window paradigm, which uses a sliding window to scan over an image exhaustively in scale-space, and classify each window individually [1, 2]. The results of the Pascal Visual Object Classes Challenge from 2005 to 2010 [3] and recent researches [4-7] show that this approach can achieve better detection accuracy than other approaches. However, due to the large number of possible target locations and pedestrian sizes in an image, enormous windows are extracted for further classification to detect all possible pedestrians. For example, the number of dense multi-scale (scale step is 1.05, sliding window size is 64×128 pixels, scanning stride is 8 pixels) sliding windows is about 25,900 on an image with resolution of 640×480 pixels. In general, the number of sliding windows grows as O(n4) for images of size n×n, which makes it computationally too expensive to exhaustively classify all of them, especially for those with high dimension features.

In recent years, many acceleration methods have been proposed to increase the detection speed by shortening the time of classification stage. For example, cascade strategy is adopted in [8, 9] so that most negative sliding windows can be rejected in the early stages of the cascade classification. Branch and bound search introduced in [10] can speed up the classification speed by trying to only focus on the image regions which have the highest possibility of containing pedestrian. However, this type of optimization can't reduce the time of other stages in sliding window paradigm (e.g. collecting features), which may become performance bottlenecks after applying these optimizing methods. Another optimizing direction is to reduce the number of sliding windows. For example, the authors of [11] combined the image pyramid and classifier pyramid to reduce the number of sliding windows with only a little reduction of detection accuracy. Coarse-to-fine detection scheme [12] can also be used to reduce the number of sliding windows, but the detection of small pedestrians is sacrificed.

In order to reduce the complexity significantly while keeping high accuracy, in this paper, we propose a novel multi-strategy filtering method to reduce the number of sliding windows. First, we apply sliding window technique merely on regions containing moving objects in the video. For video surveillance, the cameras are usually stationary most of time, so moving objects can be extracted by normal background subtraction methods. Then, besides the foreground information, we adopt clustering techniques incorporated with spatial information to form the image regions to be detected, so that the negative effects of noises on the number of windows and the fragmentation problem can be reduced to some extent. Finally, during the multi-scale scanning process, we exploit the size information of regions to avoid useless scales, hence further reduce the number of windows. Combining all these three windows filtering strategies, our method can greatly reduce the number of windows, thus achieve much faster detection speed. Meanwhile, our method can get better detection accuracy since most negative windows are filtered out before classification, hence reduces the false positive rate. Our experiments on two public datasets show that our method improves both speed and detection accuracy in comparison with the state-of-the-art methods. Most importantly, our method can achieves near real-time detection rate on both datasets, which is about 12 FPS (frames per second) on 768×576 images and 26 FPS on 384×288 images on average.

2 Related Work

As an early work, Papageorgiou et al. [13] and Viola et al. [9] have showed great success of sliding window paradigm for object detection. Based on sliding window paradigm, Dalal and Triggs [2] use HOG descriptors and SVM to build a pedestrian detector. By tuning the parameters of their HOG features, they find the best configuration for pedestrian detection through a variety of feature configurations on a challenging dataset of human figures. By combining HOG and Local Binary Pattern (LBP) as the feature set, Wang et al. [7] propose to use a global detector for whole sliding windows and part detectors for local regions in sliding windows which can well handling partial occlusion. Felzenszwalb et al. [6] build an object detection system based on mixtures of multi-scale deformable part models, which also combines the HOG and sliding window. This system achieves the best results in the PASCAL object detection challenges [3]. In summary, as a dense version of the dominating SIFT [14] feature, HOG and sliding window have shown great success in object detection and recognition [4, 5] despite its low detection speed.

In order to decrease the run-time, Qiang et al. [8] use a computationally efficient rejection chain classifier like [9] to fast filter out false alarms, with HOG based on variable size of blocks as its input feature. This method combined AdaBoost feature selection and integral image can effectively accelerate the computation speed. Lampert et al. [10] propose an efficient sub-window search (ESS) method for object localization, the method relies on a branch-and-bound scheme to find the global optimum of a quality function over all possible sub-images in the possible candidate images and can return the same object locations as the traditional sliding window approach can. Dollár et al. [11] proposed a method called FPDW which uses a step size of an entire octave to build a sparsely multi-scale image pyramid, and at each octave, multiple features introduced in [15] are approximated and a classifier pyramid is used. This approach can achieve nearly the same accuracy as using densely multi-scale image pyramids, with nearly the same speed as using a classifier pyramid applied to an image at a single scale. Wei et al. [12] propose a multi-resolution framework which uses a coarse-to-fine feature hierarchy to represent different resolutions. Then the lower resolution features are used to reject the majority of negative windows, leaving a relatively small number of windows to be processed in higher resolutions. The performance of this approach is good but small size of pedestrians will be lost. All these methods can somehow decrease the run-time of multi-scale detection, but still far from real-time processing. The reported fastest detection speed is about 6 fps for detecting pedestrians at least 100 pixels high and 3 fps for detecting pedestrians over 50 pixels on 640×480 image [11], much lower than the speed of our method as aforementioned.

Different with the existing acceleration methods, our approach mainly takes advantage of the motion information to increase the pedestrian detection speed of sliding windows based on the observation that pedestrians are usually keeping moving in video surveillance. Moreover, because most of our acceleration methods are applied in the early stages of generating sliding windows, our method can be combined with other existing approaches which accelerate classification stage to further improve the detection speed.

3 Our Method

The fast pedestrian detection based on sliding window method is illustrated in Fig. 1. In detection stage, we choose HOG as feature descriptors and Linear SVM as our classifier for their great success in object detection area. In the following, we describe the multi-strategy filtering method in details.

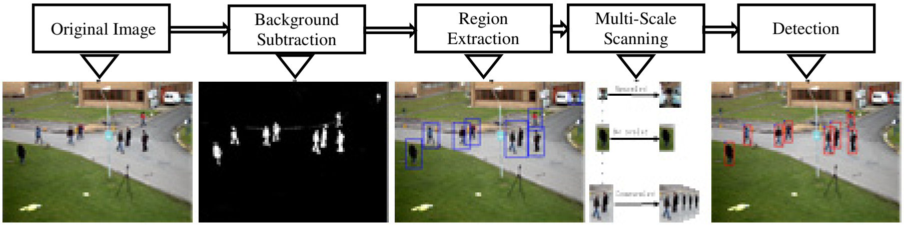


Fig. 1. Framework of our method

3.1 Background Subtraction

Pedestrians are usually walking in video surveillance. Although they may stop walking for a while, they won’t keep static for a long time. Based on this observation, we first use a fast and simple background model to filter out most non-motion pixels. We choose the ViBe [16] model to perform background subtraction. Compared to other background models, this model is appropriate for our method for four reasons: (1) It can fast separate foreground pixels from the background by combining random process to the background subtraction process. (2) It only needs a single frame to initialize the model which is necessary to start pedestrian detection without latency. (3) It uses information of neighborhood pixels to update the model which will achieve good accuracy to get foreground pixels. (4) The objects will be treated as foreground if they stop moving for a short period of time. (It is hard and not common for pedestrians to keep absolutely static for a long time especially in video surveillance, so no additional steps are needed to handle this case.)

After background subtraction, morphology operations are used to eliminate small noises like salt and pepper noises caused by sensors, compression artifacts.

3.2 Region Extraction

The foreground pixels obtained by background subtraction can't be directly used because rectangular regions are required to apply sliding window method. An intuitive solution to the generation of candidate rectangular regions is to cluster all adjacent foreground pixels and then generate different rectangular regions with the pixels in each cluster. However, this solution is not perfect because of the following issues.

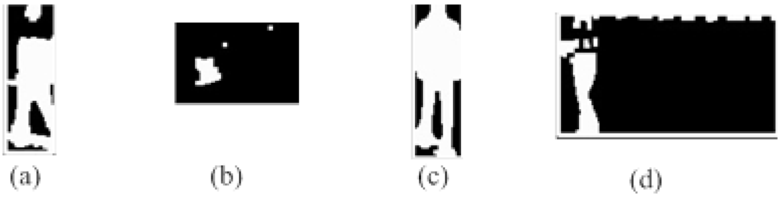


Fig. 2. Example of (a) fragments (b) noises (c) extracted region too close to the edge of pedestrian to get right HOG features (d) superfluous region caused by noises

At first, pedestrian may be fragmented into several separated elements such as the pedestrian in Fig.2 (a). For example, parts of pedestrian may temporally obstructed by the trees or fences; portions of a pedestrian may be accidentally very similar to objects in the background. In these cases, a pedestrian may not be detected or may be detected as multiple pedestrians because it is fragmented and each part is classified separately. Therefore, we introduce a clustering method which measures the similarity of two pixels in different regions. If the similarity is less than a given threshold $C_{threshold}$, these pixels need to be put into the same cluster. The similarity can be defined by distance or colors. In experiments, we find the Manhattan Distance is appropriated for our method with low computational complexity and good accuracy.

Secondly, there are some noises with relatively large areas as shown in Fig.2 (b), which cannot be removed by morphology operations. If we treat such noises as moving objects, both the size and the number of candidate regions may increase. But these regions affirmatively contain only noises other than pedestrians. Therefore, we use the size information of a pedestrian to further remove this kind of noise as follows. We set a size threshold $R_{threshold}$ for the candidate regions. All the small candidate regions are filtered out if their widths are less than $R_{threshold}$ or their heights are less than $R_{threshold}\times2$. The height threshold is chosen to double the value of the width threshold which is consistent with most of real aspect ratio of pedestrians. By choosing the suitable value for $R_{threshold}$, most candidate regions that only contain noises can be filtered out.

Thirdly, because the neighboring pixels to the pedestrians are usually necessary to extract the HOG features of the pedestrians, sliding window method can't be directly applied to the generated candidate regions since the peripheral pixels of a candidate region have not sufficient neighboring information. Fig.2 (c) shows an example of this case. Thus in our approach, each candidate regions is extended by $(\varepsilon_x,\varepsilon_y)$ before sliding window step, where $(\varepsilon_x,\varepsilon_y)$ stands for the extended width and height at both directions respectively. The extension enlarges the size of candidate regions thus increases the number of sliding windows, but the improved detection accuracy is worthy of this cost.

In some cases, the noises can't be filtered out by our noise filtering methods, for example a long waving stripe around pedestrians like Fig.2 (d). This type of noises enlarges candidate regions thus more sliding windows than necessary need to be classified. To filter out these additional sliding windows, the proportion of foreground pixels in the sliding windows is taken as a factor to filter out some sliding windows. In other words, only those windows with higher foreground pixels ratio than a threshold $F_{threshold}$ will be passed to for further detection. This filter method is reasonable and many negative sliding windows can be filtered out.

3.3 Multi-scale Detection Based on Region

After subtracting the candidate rectangular regions from each frame image, the sub-images in the corresponding regions will be treated as separate images and applied traditional multi-scale sliding window methods. We also only adopt downscaling when producing the image pyramid for each sub-image to avoid the high cost of upscaling. However, there is an exception that the sub-images need to be firstly upscaled if they are too small to contain one sliding window. The scale factor ς_l can be calculated by the following equation:

$$\varsigma_l = max\left\{\frac{SW.w}{RR.w}, \frac{SW.h}{RR.h}, 1\right\} \tag{1}$$

where SW.w and SW.h are the width and height of the sliding window; RR.w and RR.h denote the width and height of the sub-images respectively. This upscaling step helps to detect out some small size pedestrians. Meanwhile, the cost of this step is not much since only small sub-images are limitedly upscaled.

Because the size and aspect ratio are different for different sub-image, we can't use fixed downscale range in downscaling process. In our approach, the downscale range ς_s is calculated for each sub-image during runtime by using the following equation:

$$\varsigma_s = max\{min\left\{\frac{RR.w}{SW.w}, \frac{RR.h}{SW.h}\right\}, 1\} \tag{2}$$

4 Experiments

Our experiments focus on PETS2009 (dataset S2, L1, view_001)[1] consisting of 794 frames and total 4893 pedestrians and CAVIAR (dataset "ShopAssistant2cor")[2] consisting of 3700 frames and total 8740 pedestrians. The video resolution of PETS2009 is 768×576 and pedestrians' height varies from 35 to 160 pixels. The video resolution of CAVIAR is 384×288 and pedestrians' height varies from 34 to 148 pixels.

[1] http://www.cvg.rdg.ac.uk/PETS2009/a.html

[2] http://groups.inf.ed.ac.uk/vision/CAVIAR/CAVIARDATA1/

Recall, precision and F-measure are used as accuracy metrics to quantitatively compare our approach with other existing ones. Taking TP as the number of true positives, FP as the number of false positives, and P is the number of positives in the test dataset, recall is defined by R=TP/P and precision is defined by Pr=TP/(TP+FP). F-measure is the harmonic mean between recall and precision and defined by F-measure=(2×R×Pr)/(R+Pr). For F-measure, larger value means better detection accuracy. We choose to use recall-precision curve (RPC, recall against 1-precision) to do the comparison in a more informative picture. After getting the detected windows reported as containing a pedestrian, PASCAL criterion is adopted to distinguish true positive windows from false positive ones. The detected window is accepted as a true positive only if the areas overlap of this window and a ground truth exceeds 50%.

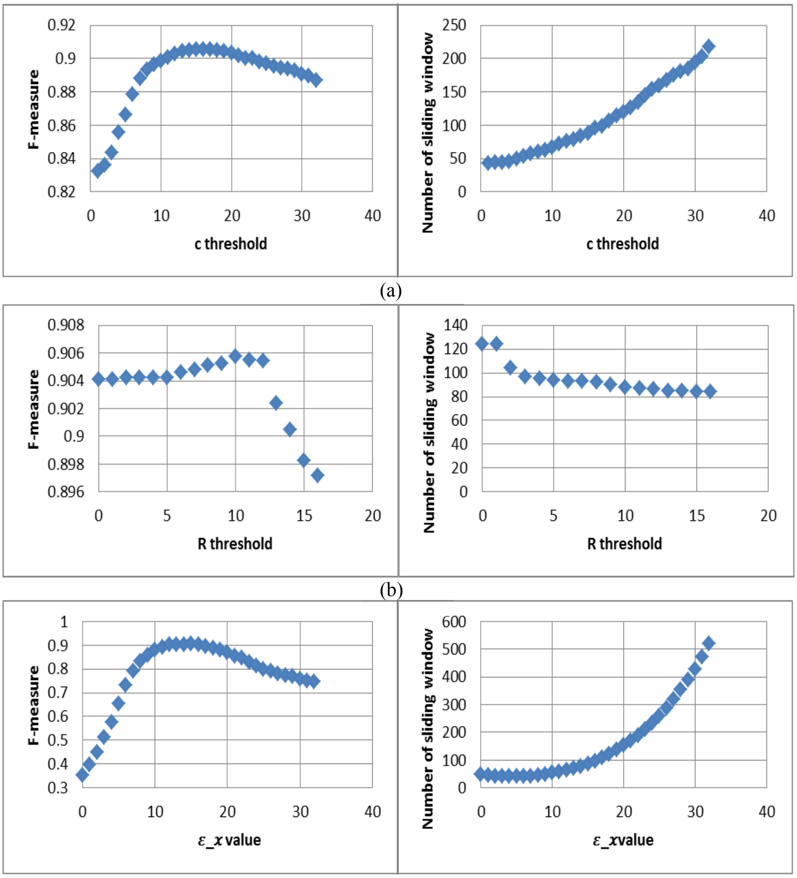
Our experiments are done on a computer with a 2.3GHz Core i5 processor and 4GB main memory. All our code is implemented in C language. The related parameters of both HOG and SVM are the same as [2]. The model is trained off-line on INRIA static image pedestrian dataset which includes 2416 positive and 1218 negative images.

4.1 Determination of the Parameters

Several parameters are introduced in our approach. Just as discussed in Section 3.2, both the detection speed and accuracy are related to these parameters. Therefore we need carefully choose values for them. We use F-measure (stands for accuracy) and the number of generated sliding windows (stands for speed) to choose the value on PET2009 dataset. To separately study the impact of the value of a parameter on the speed and accuracy, the other parameters are chosen to be a fixed value based on our experiences as following:

- Distance threshold to cluster foreground pixels: $C_{threshold}$=15;
- Size threshold to filter out small regions: $R_{threshold}$=10;
- Extension width and height for regions: ε_x=ε_y=15;
- Foreground ratio threshold: $F_{threshold}$=0.14.

Fig.3 (a) shows the impact of $C_{threshold}$ on F-measure and number of generated sliding windows. From the figure, we can see that the number of sliding windows fast increases as the value of $C_{threshold}$ is increased. It is reasonable because more foreground pixels are clustered together thus larger candidate regions are generated. For F-measure, the relation is more complicate. When increasing the value of $C_{threshold}$, on one hand, part of the fragmented pedestrians has more chances to be clustered together thus get better detection accuracy; however on the other hand, some noises or other moving objects may also be clustered into the same clusters of pedestrians, which cause larger candidate regions along with more potential false positive results. As a result of the two factors, F-measure firstly sharply goes up when increasing the value of $C_{threshold}$ from 1, and then slowly goes down after getting the highest value when $C_{threshold}$ is 15.



(a)

(b)

(c)

Fig. 3. F-measure and number of sliding windows for different value of (a) $C_{threshold}$ (b) $R_{threshold}$ (c) ε_x (We only show the results on PETS2009, for the reason that the impact of these parameters on F-measure and number of generated sliding windows is almost the same for those 2 datasets except that the concrete values are different.)

Since using larger value for $R_{threshold}$ can filter out more small candidate regions, the number of sliding windows decreases as the increasing of $R_{threshold}$'s value, just as showed in Fig.3 (b). Although selecting larger value for $R_{threshold}$ can filter out more negative regions, true positive regions may also be filtered out. From Fig.3 (b), we can see that the highest F-measure is obtained when the value of $R_{threshold}$ is 10.

For simplification, we use the same value for ε_x and ε_y. It is not surprising that the number of sliding windows sharply increases with the increasing value of ε_x. However, it is not intuitive for the left curve in Fig.3 (c). The gradient information of the neighborhood pixels outside the pedestrian's contour is necessary to collect concrete HOG features, thus we can get higher F-measure when increasing the value of ε_x from 0. However, just as the same reason of $C_{threshold}$, larger candidate regions caused by larger ε_x also cause more potential false positive results. Therefore, F-measure slowly decreases when the value of ε_x is greater than 15. This value is reasonable because in our training dataset, the average height of pedestrians is around 34×98 and the size of sliding window is 64×128.

The value of $F_{threshold}$ is set to be 0.14 according to our experiments on the background subtraction model: the proportion of foreground pixels of a pedestrian in a window is no less than 0.14.

4.2 Comparison with Other Methods

We compare our approach with Dalal and Triggs' (D&T) [2] approach and Dollǻr's FPDW [11] approach. We choose these two approaches to do comparison because D&T is the most classical approach for pedestrian detection and FPDW is the fastest approach based on sliding window paradigm according to the best of our knowledge.

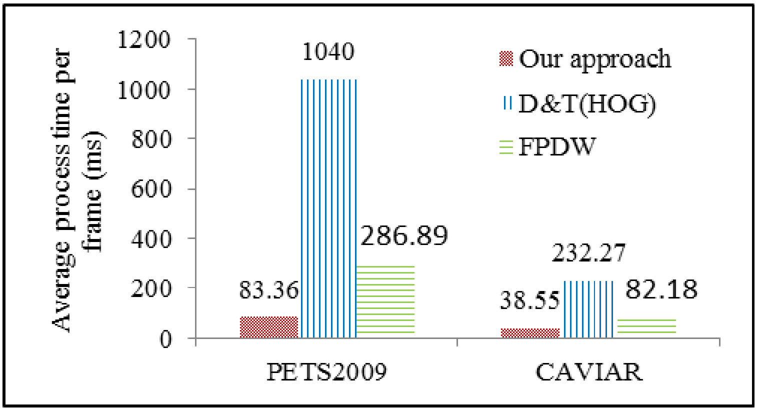


Fig. 4. Average process time per frame of our approach, D&T and FPDW on PETS2009 and CAVIAR

Fig.4 compares the average process time per frame of our approach, D&T and FPDW on PETS2009 and CAVIAR datasets. We can see that our approach is the fastest one. Our approach can achieve about 12 FPS on PETS2009, while D&T and FPDW can only achieve about 1 FPS and 3 FPS respectively. Because the video resolution of CAVIAR is lower than PETS2009, the processing time is much shorter. On CAVIAR, the detection speed of our approach is about 26 FPS, much faster than that of D&T and FPDW which are about 4 FPS and 12 FPS respectively. The speed advantage of our method mainly comes from the reduced number of windows needed to be detected through our sliding window filtering strategies. For example, the average number of sliding windows per frame on PETS2009 can be reduced from about 42,000 to 88 by our filtering strategies. Thus the cost of both features collecting and classifying of the removed windows is saved. Although background subtraction and region extraction introduce a bit of overhead, this overhead is much less than the saved time. For example, the average overhead of background subtraction and region extraction on CAVIAR is about 8.84ms and 5.52ms per frame respectively, whereas the HOG detection time is reduced from 232.27ms to 23.89ms, about 9.72 times of improvement.

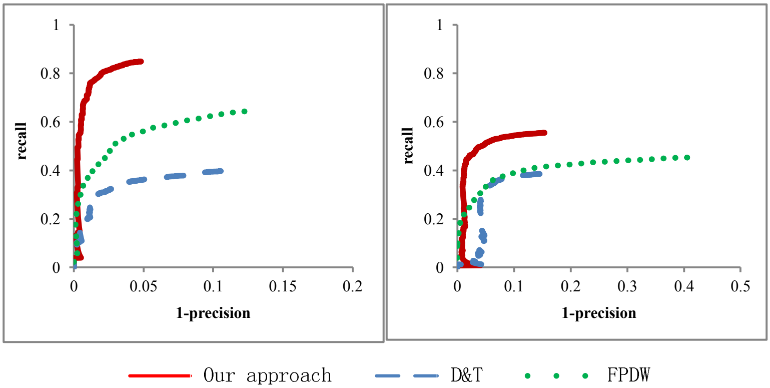


Fig. 5. RPC for (a) PETS2009 (b) CAVIAR

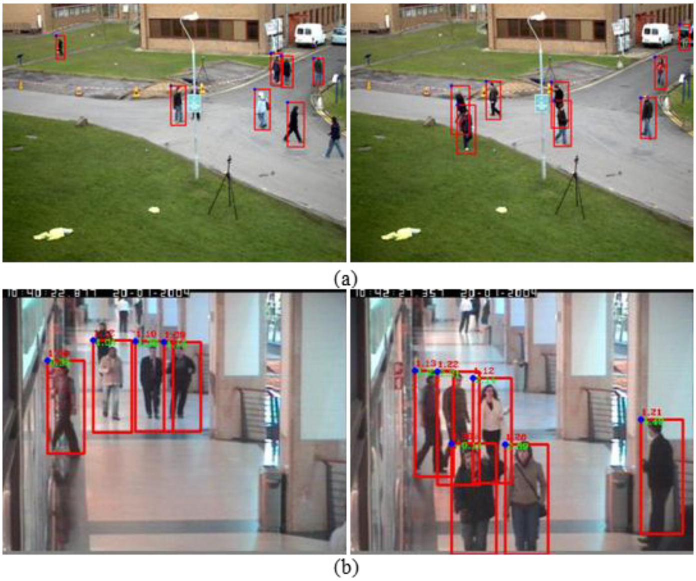


Fig. 6. Detection results of our approach on (a) PETS2009 (b) CAVIAR

From the RPC showed in Fig.5, we can see that our approach can get better detection accuracy than D&T and FPDW on both datasets. For example, the highest F-measure got by our approach is 0.906 on PETS2009, which is improved by 22% and 62% against FPDW (0.742) and D&T (0.559) respectively. The improvement of accuracy comes from two sides: (1) there are less false positive windows since most negative windows have been filtered out by our filtering strategies; (2) we can detected out some small size pedestrians as we have upscaling step during the multi-scale detection process. If we don't use the upscaling step, the F-measure will decrease from 0.906 to 0.43 on PETS09 and from 0.669 to 0.34 on CAVIAR. We also notice that all the three approaches are not good on CAVIAR dataset because the training dataset (mostly outdoors and seldom occlusions) is very different from CAVIAR dataset (all indoors and many occlusions). Fig.6 shows some detection results of our approach on the two datasets.

5 Conclusion

In this paper, we propose a fast pedestrian detection method by applying sliding window only on the regions containing moving objects. By reducing the number of sliding windows to be classified by combining filtering strategies based on motion and size information of the video, our method overcomes the speed drawback of sliding window paradigm without sacrificing the accuracy of detection. Our experimental results on two public datasets show that our proposed method can achieve near real-time detection rate, which is much faster than the state-of-the-art methods. Meanwhile, the accuracy of detection is also improved.

In the future, we plan to integrate other acceleration orthogonal methods (e.g. cascade method) to our system to further speed up the whole process to satisfy the real-time requirements for applications with higher video quality. Besides, how to achieve high accuracy of detection on diverse datasets of different domains through domain adaption technique is also our future work.

Acknowledgement. This work was supported in part by the National Nature Science Foundation of China (61173054, 61271428); and Co-building Program of Beijing Municipal Education Commission.

References

1. Dalal, N.: Finding people in images and videos. Institute National Polytechnique de Grenoble (2006)
2. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 886–893 (2005)
3. Everingham, M., et al.: The PASCAL Visual Object Classes Challenge 2010 (VOC 2010) Results, http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html
4. Dollar, P., et al.: Pedestrian detection: A benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 304–311 (2009)
5. Enzweiler, M., Gavrila, D.M.: Monocular Pedestrian Detection: Survey and Experiments. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 31(12), 2179–2195 (2009)
6. Felzenszwalb, P.F., et al.: Object Detection with Discriminatively Trained Part-Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 32(9), 1627–1645 (2010)
7. Wang, X., Han, T.X., Yan, S.: An HOG-LBP human detector with partial occlusion handling. In: IEEE International Conference on Computer Vision, ICCV, pp. 32–39 (2009)

8. Qiang, Z., et al.: Fast Human Detection Using a Cascade of Histograms of Oriented Gradients. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1491–1498 (2006)
9. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR (2001)
10. Lampert, C.H., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: Object localization by efficient subwindow search. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR, pp. 1–8 (2008)
11. Dollár, P., Belongie, S., Perona, P.: The Fastest Pedestrian Detector in the West. In: British Machine Vision Conference, BMVC (2010)
12. Wei, Z., Zelinsky, G., Samaras, D.: Real-time Accurate Object Detection using Multiple Resolutions. In: IEEE International Conference on Computer Vision, ICCV, pp. 1–8. IEEE (2007)
13. Papageorgiou, C., Poggio, T.: A Trainable System for Object Detection. International Journal of Computer Vision, IJCV 38 (2000)
14. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision, IJCV 60(2), 91–110 (2004)
15. Dollár, P., et al.: Integral Channel Features. In: British Machine Vision Conference, BMVC (2009)
16. Barnich, O., Van Droogenbroeck, M.: ViBe: a universal background subtraction algorithm for video sequences. IEEE Transactions on Image Process., ITIP 20(6), 1709–1724 (2011)