A Heuristic Deformable Pedestrian Detection Method

Yongzhen Huang, Kaiqi Huang, Tieniu Tan

National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, China
{yzhuang, kqhuang, tnt}@nlpr.ia.ac.cn

Abstract. Pedestrian detection is an important application in computer vision. Currently, most pedestrian detection methods focus on learning one or multiple fixed models. These algorithms rely heavily on training data and do not perform well in handling various pedestrian deformations. To address this problem, we analyze the cause of pedestrian deformation and propose a method to adaptively describe the state of pedestrians' parts. This is valuable to resolve the pedestrian deformation problem. Experimental results on the INRIA human dataset and our pedestrian pose database demonstrate the effectiveness of our method.

1 Introduction

Pedestrian detection plays an important role in computer vision applications, e.g., surveillance, automatic navigation and human computer interactions. Many pedestrian detection algorithms have been developed. Haar wavelet-based cascade approaches [1] constructs a degenerate decision tree of progressively complex detector layers composed of a set of Haar wavelet features. Part based methods [2] utilize the orientation and position histograms to built parts detectors, and then combine them to a whole description of pedestrians. Edge based methods [3] design a set of enhanced edge features to capture the edge structure of pedestrians. The work of histogram of oriented gradient (HOG) [4] is a milestone in pedestrian detection. It is the first work that achieves impressive performance on pedestrian detection. The HOG algorithm uses the SIFT-like descriptor in each of dense overlapping grids with a local normalization strategy to generate feature maps. Then, the sliding window scheme is adopted, where the trained model is matched with windows at all positions and scales of an image. The HOG feature is good at describing the spatial distribution of pedestrians' edge. In addition, it is insensitive to illumination change because of the local normalization and resistant to position and scale variation due to the sliding window strategy. The success of HOG spurs more researches on pedestrian detection. Many approaches have been developed, focusing on feature selection and combination [5], [6], [7], [8], [9], speed enhancement [10], [11], components learning and spatial relationship description [12], [13], combination with poses estimation [14] and machine learning applications, e.g., kernel tricks [15] and

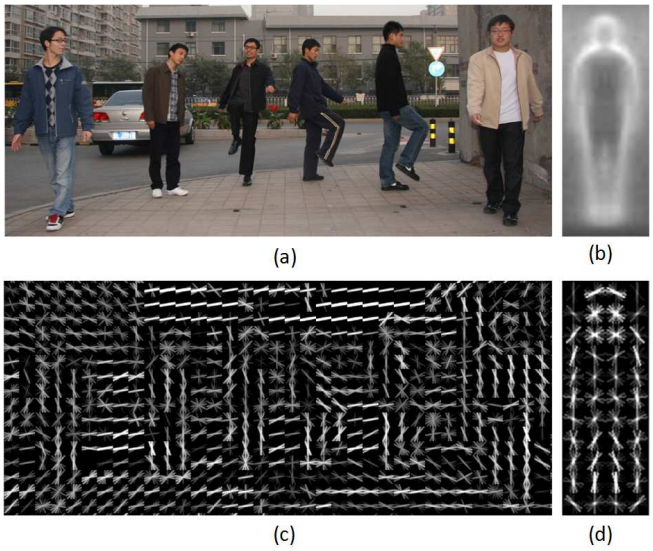

Fig. 1. A demonstration showing the limitation of HOG in handling pedestrian deformation. (a) Six pedestrians with different poses and viewpoints. (b) The average gradient image of pedestrians. (c) The HOG feature map. (d) The positive model trained by the HOG algorithm. Obviously, one model cannot describe all poses of pedestrians.

different Boosting algorithms [16]. Comprehensive studies can be found in two recent surveys [17], [18]. However, these algorithms share all or some of the following problems:

- Rely heavily on training data. It takes much time and resource to collect and label pedestrian images. Moreover, the trained model usually cannot be directly used for other scenes, e.g., the model trained by outdoor datasets may perform bad in indoor scenes.
- Use fixed models to describe deformable pedestrians. For example, the HOG algorithm generates one template that best differentiates positive and negative samples in the training set. We can consider that it describes the average feature map of pedestrians. This strategy is limited to handle pedestrian deformation. Fig. 1 shows an example. Recently, some researchers combined pedestrian's part models and their spatial relationship [13]. In this framework, each part is still described by a fix model. It ignores the deformation of parts themselves, and thus cannot describe all shapes of parts. For example, a standing person's leg is different from a running person's leg in shape.

To address the above problems, we propose a heuristic deformable pedestrian detection approach. It doesn't need any training samples and can adaptively describe shapes of pedestrians' parts. This is because we use the prior knowledge on pedestrian deformation, including: 1) most pedestrian deformations are caused by variation of body parts, e.g., arms waving; 2) the components of a part, e.g.,

the upper and the lower arms, are rigid (limited by the rigidity of bones). Thus, they can only rotate around joint points. 3) the joint points can only move in a reasonable range. Therefore, each key part can be described by the combination of two lines which can rotate around joint points. We call it as the "two lines" model. Fig. 2 illustrates an example. In the next section, we will elaborate how to adaptively estimate the location of joints points and the angle of two lines.

Fig. 2. An example showing person deformation and the "two lines" model.

2 Heuristic deformable pedestrian detection

In this section, we first briefly introduce our algorithm, and then elaborate each stage. The framework is shown in Fig. 3. It consists of five stages:

1. An input image is convolved by eight Gabor filters with different orientations. The outputs are Gabor feature maps (termed as S1 units in this paper).
2. S1 units are convolved by a group of component filters and the outputs are the response map of components (S2 units).
3. Two S2 units are combined to describe a part. The choice of S2 units is adaptive according to the maximum response of component filters with respect to the orientation. The output is the response map of parts (S3 units).
4. The most possible location of a part is adaptively calculated by searching the max response of S3 units in a predefined range, c.f. spatial constraints in Fig. 3. These locations indicate key parts that cause pedestrian deformation.
5. Finally, seven maximums are combined in each sliding window. The output is the detection window and the location of joint points (red points).

2.1 S1 units

The S1 layer produces shape information of pedestrians. An input image is convolved by a group of Gabor filters with different orientations:

$$S1(\theta, x, y) = I(\theta, x, y) * G(\theta, x, y), \tag{1}$$

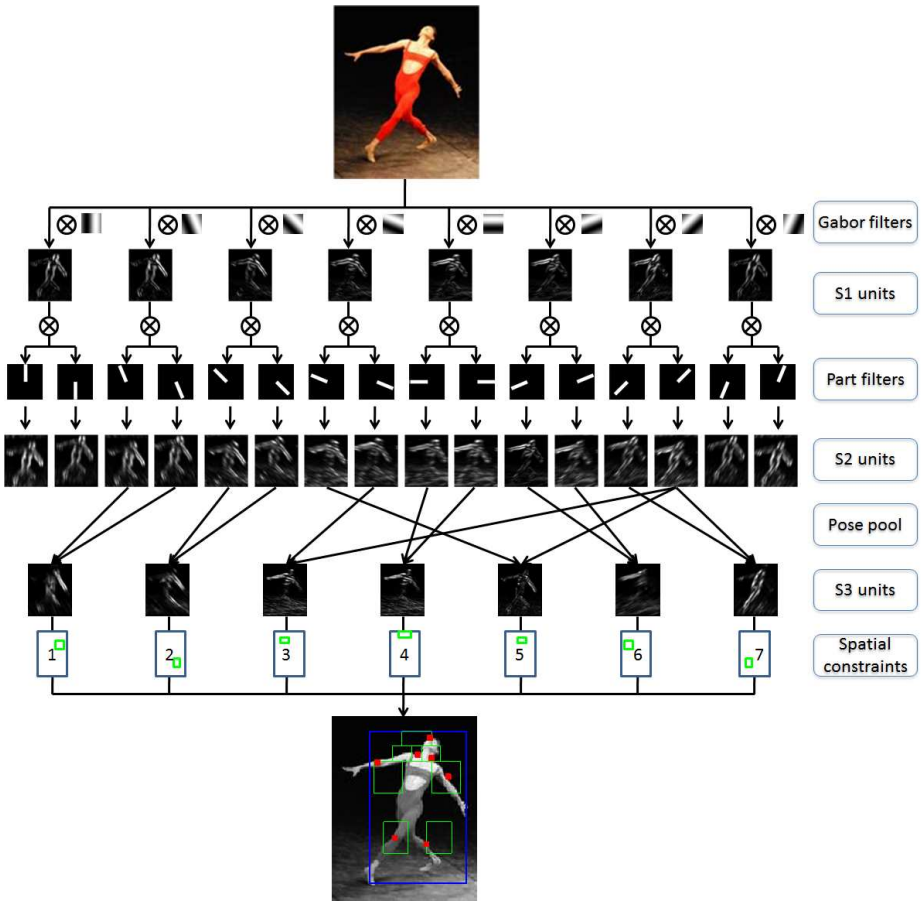

Fig. 3. The framework of our system.

$$G(\theta, x, y) = \exp(-\frac{x_0^2 + \gamma^2 y_0^2}{2\sigma^2}) \times \sin(\frac{2\pi}{\lambda} x_0), \tag{2}$$

$$x_0 = x\cos\theta + y\sin\theta, y_0 = -x\sin\theta + y\cos\theta, \tag{3}$$

where I is the input image and G indicates the Gabor filter. The ranges of x and y are the abscissa and ordinate associated with the scales of Gabor filters and θ controls the orientations.

The Gabor filter is similar to the receptive field profiles in the mammalian cortical simple cells [19]. It has good orientation and frequency selectivity. From the viewpoint of computer vision, Gabor filters are more robust to noises compared with the gradient filters ([-1, 0, 1] and [-1, 0, 1]') adopted in the HOG algorithm. This is because they contain scale information and reflect the area

statistic properties similar to the edges of pedestrians' parts. An example is shown in Fig. 4.

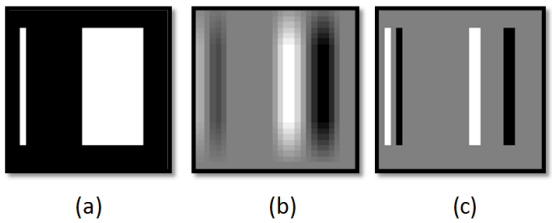

Fig. 4. Comparison between the Gabor filter and the gradient filter. (a) A test image. The left line is usually not from pedestrian. The right object is more similar to components of pedestrians which are homogeneous over a region. (b) The response of the Gabor filter. The Gabor filter produces larger response if the object's scale is similar to the size of the Gabor filter. (c) The response of the gradient filter. The gradient filter does not contain scale information, thus all vertical edges produce the same response.

2.2 S2 units

Most existing pedestrian detection algorithms train one or several fixed models. They are limited on describing pedestrian deformations. To solve this problem, we use an adaptive deformable model. As we analyze in Fig. 2, the components of a part, e.g., the upper and the lower arms, are rigid, and they can only rotate around joint points. Thus, a part can be described by the combination of two lines. In our system, we define 16 components filters (illustrated in Fig. 3). A component filter is a rectangular template defined in a binary array. The size of the component filter is double the one of the Gabor filter. One component filter reflects the orientation information of a component in a part, and the combination of two component filters can describe a part.

To generate S2 units, each Gabor feature map (S1 unit) is convolved by the corresponding[1] component filter:

$$S2(\theta, x, y) = S1(\theta, x, y) * P(\theta, x, y), \tag{4}$$

where P is the component filters, θ is the orientation of the filters and $S1$ is defined in Eq. (1).

2.3 S3 units

To describe parts' information (S3 units), we calculate the most possible orientation for each component by searching the maximum response with respect to the orientation, in the predefined range of parts:

[1] Gabor filters and component filters are with 8 $(0 - 180^o)$ and 16 $(0 - 360^o)$ orientations, respectively. Each Gabor filter corresponds to two component filters.

$$S3(k,x,y) = \max_{\theta_1}\{S2(\theta_1,x,y)\} + \max_{\theta_2,\theta_2\neq\theta_1}\{S2(\theta_2,x,y)\}, \tag{5}$$

$$x,y \in R(k), \tag{6}$$

where k is the index of parts, θ is the orientation of component filters, and R is the predefined range of parts which can be estimated by prior knowledge (c.f. the spatial constraints in Fig. 3). This maximum operation adaptively finds the orientation information of the components of a part, i.e., the angle of two components.

The final score of a searching window is the sum of maximum response of all S3 units with respect to the spatial constraints:

$$score = \sum_{k=1}^{7} \max_{x,y\in R(k)}(S3(k,x,y)), \tag{7}$$

The most possible location of a part is adaptively computed by searching the max response of $S3$ in $R(k)$.

To the best of our knowledge, the proposed system is the first attempt of an adaptive (no training data) pedestrian detection system that handles deformations induced by pose and viewpoint variations. The subsequent section empirically studies the system.

3 Empirical studies

In this section, we first test our algorithm in the INRIA human dataset [20]. In this experiment, we study the influence of parameters to the performance and compare our algorithm with some popular ones. Second, considering that the INRIA human dataset contains limited pose variations, we built a new pedestrian database that includes almost all possible poses of a normal pedestrian. Details are given in Section 3.2. In this experiment, we focus on the invariance to poses variations. Finally, we analyze the speed of our algorithm.

3.1 INRIA human dataset

The INRIA human dataset contains 1805 64*128 images of pedestrians cropped from a varied set of personal photos. Most pedestrians are bystanders with a wide variety of backgrounds. Besides, it provides 454 person-free images for testing. As our algorithm doesn't need training sample, we directly test it on the testing set.

There are three parameters in our algorithm: the size of Gabor filters, the number of component filters and the predefined range of parts in Eq. (6). We empirically find that the second and third parameters are insensitive to performance in a reasonable range. In all of our experiments, we set the number of component filters to 16, and the ranges of parts are shown in Fig. 3. To save

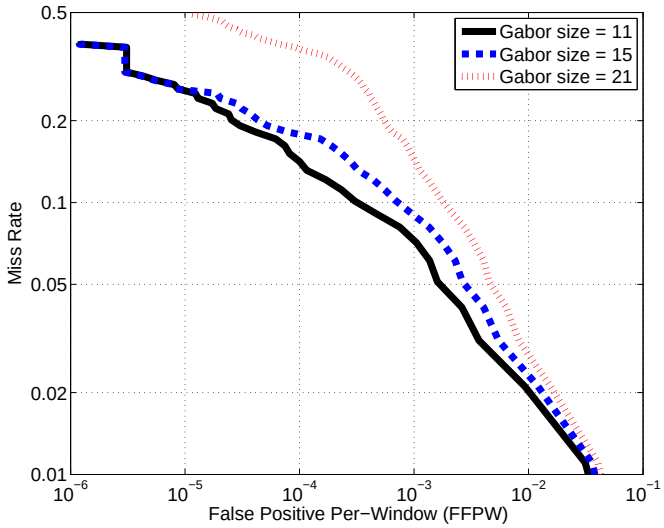

Fig. 5. Performance of our method with different Gabor sizes on INRIA human dataset.

page length and to keep the paper concise, we only analyze the influence of the Gabor size. Fig. 5 shows the performance.

Our algorithm performs well when the Gabor size is 11 and 15, which is smaller than width of pedestrians' head or limbs. The Gabor filters with this size produce desirable response in pedestrians' edge. In this case, they can better describe the edge information of pedestrians. In the later experiments, we fix the size of Gabor filters to 11.

Afterwards, we compare our algorithm with HOG [4], FtrMine [21], VJ's method [1] and LSVM [13]. All results except ours are reported in [17]. The testing strategies include per-window and per-image [17]. We strictly obey the testing rule[2] [20] adopted in the evaluation of HOG. Fig. 6 and Fig. 7 show the results.

Our algorithm does not performs best but comparable to HOG and LSVM which are two state-of-the-art pedestrian detection algorithms. Considering that our algorithm does not need any training samples, we think it performs well. In the later part of this paper, we will analyze the failure cases and discuss the potential enhancement of our method.

[2] For readers' convenience, we cite the rule here: the starting scale in scale-space pyramid is one and keep adding one more level in the pyramid till $floor(ImageWidth/Scale) > 64$ and $floor(ImageHeight/Scale) > 128$. Scale ratio between two consecutive levels in the pyramid is 1.2. Window stride (sampling distance between two consecutive windows) at any scale is 8 pixels.

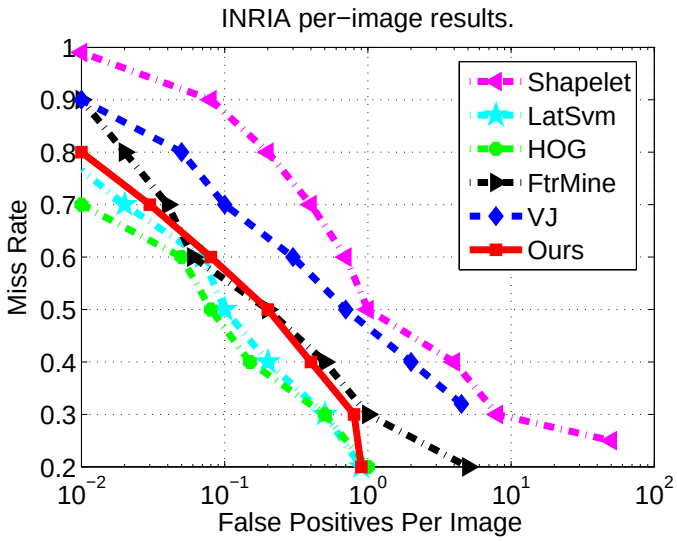

Fig. 6. Performance comparison on INRIA human dataset by the per-window testing rule.

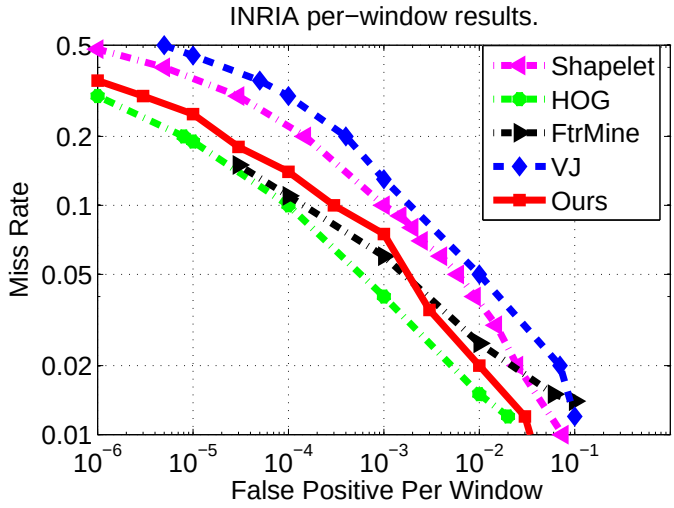

Fig. 7. Performance comparison on INRIA human dataset by the per-image testing rule.

3.2 Pedestrian poses database

The INRIA database mainly focuses on differentiating pedestrians from various backgrounds. Subsequently, we will analyze the invariability to pedestrian

deformation induced by pose and viewpoint variations. We built a large pedestrian pose database[3]. This database contains 1,440 images at three viewpoints $(0^o, 45^o, 90^o)$ and six different scenes. The database captures 30 persons, each with 48 poses. Also, we record the video when they regularly walk and run. Therefore, our database contains most poses of pedestrians.

The evaluation criterion is the "per-image" rule used in [17]: input an image and output a bounding box and score for each detection window. A detected bounding box and a ground truth bounding box are matched if their overlapped areas exceed 50% of their sum:

$$\frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} > 0.5, \tag{8}$$

where BB_{dt} is the detected bounding box and BB_{gt} is the ground truth bounding box.

Unmatched BB_{dt} is counted as false positives and unmatched BB_{gt} as false negatives. We plot the miss rate against false positives per-image curve by varying the threshold on detection confidence. In addition, we provide the performance by HOG. The curves are shown in Fig. 9.

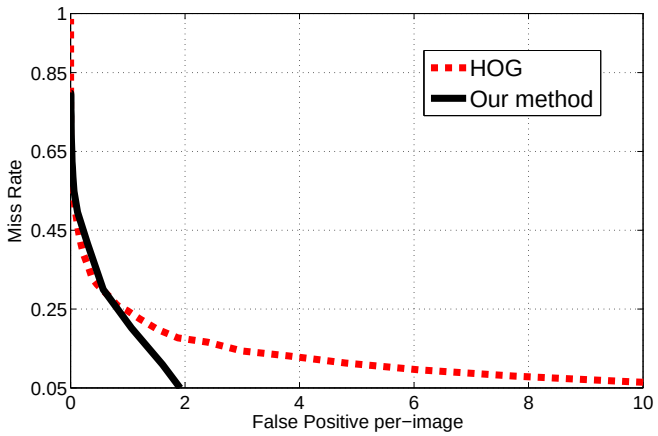

Fig. 8. Performance comparison between our method and the HOG algorithm.

When the miss rate is high, HOG performs slightly better than our method. When the miss rate is low, specifically lower than 0.25, our method outperforms HOG. In addition, we provide the confidence distribution of all the samples by

[3] The address of the database is now being updated. Please contact the author if you are interested in the databse.

our method and the HOG algorithm in Fig. 10. The confidence scores by HOG are more widely distributed, which demonstrates that HOG is more sensitive to pose and viewpoint variations.

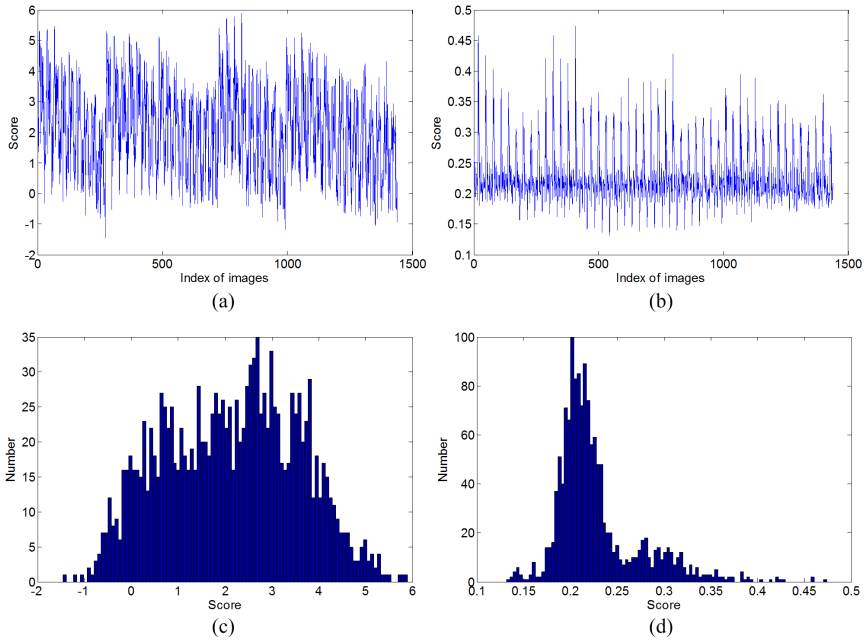

Fig. 9. Confidence distribution. (a) The score of each sample by HOG. (b) The score of each sample by our method. (c) The distribution of score by HOG. (d) The distribution of score by our method.

Fig. 11 analyzes some missed samples by our method. Our method is not sufficiently robust to large contrast variation. This is one of mains problems we would like to addressed in the future work.

Finally, we show some detection results by HOG and our method in Fig. 12.

3.3 Efficiency analysis

It takes about one second for our algorithm to handle an image with the resolution of 320×240 by Matlab on a personal computer with 2.4G CPU (Inter Core2) and 4G RAM. The main computation is the filtering operation in calculating S2 units and S3 units (more than 95% computation of the whole algorithm). According to our previous experience, if we use IPP filtering function [22] that is 20 time faster than the Matlab filtering function, the system can achieve real-time (less than 0.1 second per image).

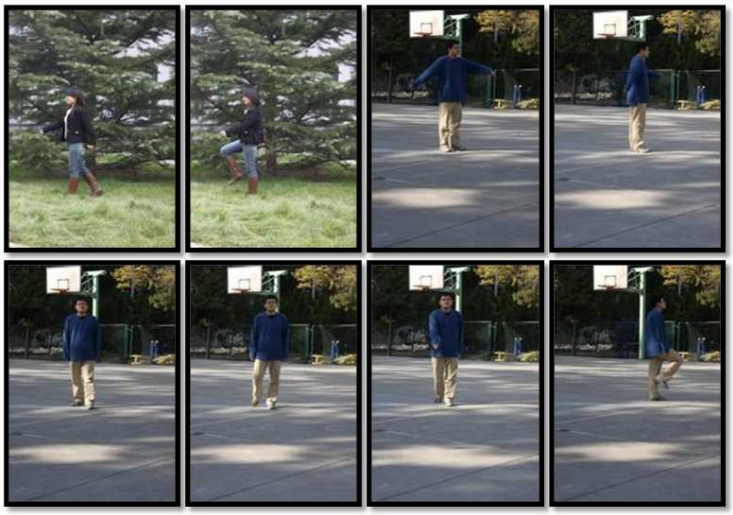

Fig. 10. Missed positive samples by our method. These samples are all in bad contrast conditions

4 Discussion

Our method can be viewed as a group of deformable models for parts, and pedestrian detection is transformed to searching local optimal description of parts. In fact, this strategy has been utilized by [23] which uses a series of positive examples to estimate appropriate weightings of features relative to one another and produce a score that is effective at estimating configurations of pedestrians. Earlier, it comes from the analysis of articulated objects [24]. The novelty of our work is that our method can adaptively describe the deformable parts. This is because we effectively use the prior knowledge on pedestrian deformation (see analysis in the introduction section).

5 Conclusion

In this paper, we have analyzed the problems of current pedestrian detection algorithms, and presented a heuristic deformable pedestrian detection method. It can handle pedestrian deformation induced by pose and viewpoint variations without training samples. Experiments on the INRIA human dataset and our pedestrian pose database have demonstrated that our method is comparable to some state-of-the-art algorithms but does not require training samples.

In the future, we will focus on enhancing the accuracy of our algorithm. We consider that the accuracy may be enhanced by further exploring the form of deformable part filters, the discrimination of different key parts and their spatial relationship. In addition, we intend to integrate the proposed algorithm with appearance based methods to improve the performance under poor contrast conditions.

Fig. 11. (best view in color) 1st row: missed positive samples by HOG and their scores. These samples are with large pose variation. 2nd row: detection results by our method. 3rd row: detection results on multi pedestrians images by HOG. 4th row: detection results on multi pedestrians images by our method.

References

1. Viola, P., Jones, M., Snow, D.: Detecting pedestrians using patterns of motion and appearance. ICCV (2003)
2. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. ECCV (2004)
3. Wu, B., Nevatia, R.: Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors. International Journal of Computer Vision **75** (2007) 247–266
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. CVPR (2005)
5. Sabzmeydani, P., Mori, G.: Detecting pedestrians by learning shapelet features. CVPR (2007)
6. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on riemannian manifolds. CVPR (2007)
7. Chen, Y.T., Chen, C.S.: Fast human detection using a novel boosted cascading structure with meta stages. IEEE Trans. on Image Processing **17** (2008) 1452–1464

8. Wu, B., Nevatia, R.: Optimizing discrimination-efficiency tradeoff in integrating heterogeneous local features for object detection. CVPR (2008)
9. Schwartz, W.R., Kembhavi, A., Harwood, D., Davis, L.S.: Human detection using partial least squares analysis. ICCV (2009)
10. Zhu, Q., Yeh, M.C., Cheng, K.T., Avidan, S.: Fast human detection using a cascade of histograms of oriented gradients. CVPR (2006)
11. Zhang, W., Zelinsky, G., Samaras, D.: Real-time accurate object detection using multiple resolutions. CVPR (2007)
12. Dollar, P., Schiele, B., Belongie, S., Perona, P., Tu, Z.: Multiple component learning for object detection. ECCV (2008)
13. Felzenszwalb, P., Mcallester, D., Ramanan, D.: A discriminatively trained, multiscale, deformable part model. CVPR (2008)
14. Andriluk, M., Roth, S., Schiele, B.: Pictorial structures revisited: People detection and articulated pose estimation. CVPR (2009)
15. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. CVPR (2008)
16. Begard, J., Allezard, N., Sayd, P.: Real-time human detection in urban scenes: Local descriptors and classifiers selection with adaboost-like algorithms. CVPR Workshops (2008)
17. Dollar, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. CVPR (2009)
18. Enzweiler, M., Gavrila, D.M.: Monocular pedestrian detection: Survey and experiments. IEEE Trans. on Pattern Analysis and Machine Intelligence **31** (2009) 2179–2195
19. Jones, J.P., Palmer, L.A.: An evaluation of the two-dimensional gabor filter model of simple receptive fields in cat striate cortex. Journal of Neurophysiology **58** (1987) 1233–1258
20. http://pascal.inrialpes.fr/data/human/.
21. Dollar, P., Tu, Z., Tao, H., Belongie, S.: Feature mining for image classification. CVPR (2007)
22. http://www.intel.com/cd/software/products/asmo-na/eng/302910.htm.
23. Tran, D., Forsyth, D.: Configuration estimates improve pedestrian finding. NIPS (2008)
24. Ramanan, D.: Learning to parse images of articulated objects. NIPS (2006)