Human Detection and Object Tracking Based on Histograms of Oriented Gradients

Shibo Zhang
College of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China

Xiaojie Wang
College of Computer Science
Beijing University of Posts and Telecommunications
Beijing, China

Abstract—**Feature extraction methods are widely used in the object detection procedure. In this paper, improved Histograms of Oriented Gradients features are used to represent the edge information of images. In order to track in real-time, we use background subtraction detection with Histograms of Oriented Gradients, which achieves the required accuracy and satisfies real-time demand.**

Keywords- Human detection; Histograms of Oriented Gradient (HOG); Object Tracking

I. Introduction

Human detection is an important issue for enhancing traffic safety in Intelligent Transportation Systems and advanced human-computer interaction and so on.

Feature extraction is the first fundamental step in most object detection and recognition algorithms. Researchers have done a lot of work in the human detection technology. Dalal et al. propose to use HOG as a feature descriptor, where there are 3780 feature vectors in the 128×64 image window. In addition, good results can be achieved by using a SVM classifier in INRIA library. After that, Zhu et al. improve the work based on reference. In their opinion, multi-scale HOG features are used and the Adaboost algorithm selects main features for linear SVM weak classifiers which are taken to compose strong classifiers of cascade structure. Mori use a new feature which pays more attention to the low-level radiant information of local areas of the image, and Adaboost is employed to select a subset of learned features. Recently, Wang et al. use HOG and Local Binary Pattern (LBP) as a new feature set, and a good human detector is trained by SVM. Y. Pang et al. combine HOG features and improved HOG features which are to utilize sub-cell based interpolation to efficiently compute the HOG features for each block to deal with time-consuming problem. Ali et al. use some novel features that inspired from biological vision system for human recognition. Feifei Lee et al. employ two types of histogram descriptors to get an outstanding video search algorithm for large video database.

HOG features clearly depict the edge information of images and reduce the impact of illumination. However, it is difficult to reach real-time process because of complex calculation. Therefore, it is necessary that calculation procedure should be optimized.

Though HOG feature achieves great detection rates, the high dimensionality of its feature vector poses as one of its disadvantages. The large size of the feature vector limits the number of training samples and increases the computation cost in SVM classification.

In this paper, HOG is adopted as the basic feature, takes the advantage of its highly discriminative power, and creates a much simpler feature with Adaboost. Then train a linear SVM to perform the classification with the obtained feature. Our detector achieves comparable results on the INRIA dataset compared with HOG feature, yet the feature size has been compressed. As a result the computation cost and storage requirement in the SVM classification is much lower.

When object tracking, the application scenario we are interested in is the video surveillance system, in which a static camera observing a scene is a common case. Moving objects can be segmented from the background. Therefore, it is reasonable to presume that we can get the silhouette of the moving object before the classification stage. The contour can provide not only the position, but also more reliable features for human detection when using HOG descriptors. In order to track human in real-time, we extract the front-image from each frame of video, and employ HOG descriptor to marking the human in the sub image.

A fundamental requirement for effective automated analysis of object behavior and interactions in video is that each object must be consistently identified over time. This is difficult when objects are often occluded for long periods: nearly all tracking algorithms will terminate a track with loss of identity on a long gap. The problem is further confounded by objects in close proximity, tracking failures due to shadows, etc. Recently, some work has been done to address these issues using higher level reasoning, by linking tracks from multiple objects over long gaps. However, these efforts have assumed a one-to-one correspondence between tracks on either side of the gap. This is often not true in real scenarios, where objects are closely spaced and dynamically occlude each other, causing trackers to merge objects into single tracks.

II. HOG Feature Extraction

Feature extraction is a fundamental step towards object detection and recognition. The HOG descriptor has shown the robustness and reliability for representing image local features.

978-1-4673-4714-3/13/$31.00 ©2013 IEEE

The essential idea of the descriptor is to capture or encode the local object appearance and shape as the distribution of local intensity gradients or edge directions.

Dalal et al. have proposed to represent the image information by using HOG features. Each pixel of the input image will be computed with a discrete first-order differential, and the filter template is [-1 0 1]. Next, the image is divided into 105 blocks, and the proportion of overlap between blocks is 0.5. Every block region contains 4 unit cells which are 8×8 pixels, and each cell consists of 9 orientation bins in 0-π. Moreover, tri-linear interpolation and Gaussian weighting is done to reduce calculation errors. Finally, every block has a 36-D vector which is normalized by using L2-Hys. According to the algorithm, there are 3780 features in 128×64 detection window.

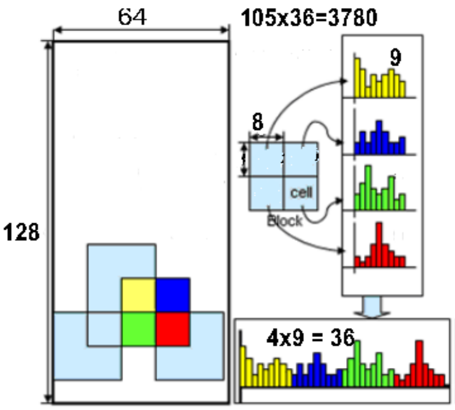

Figure 1. HOG Calculation

An overview of the HOG descriptor can be described as follows:

1. To reduce the illumination variance in different images, the gray-scale normalization is performed so that all images have the same intensity range.

2. The same centered [-1, 0, 1] mask is used to compute horizontal gradient $G_x(x,y)$ and vertical gradient $G_y(x,y)$ of every pixel.

The magnitude m and orientation θ of gradients $G_x(x,y)$ and $G_y(x,y)$ should be computed for each pixel (x,y) inside a detection window by the following equations, where $i(x,y)$ is gray scale luminance and $H(x,y)$ means the normalized luminance with power law (gamma) equalization at pixel (x,y)

$$H(x,y)=\sqrt{i(x,y)} \quad (1)$$

$$\begin{aligned} G_x(x,y) &= H(x+1,y)-H(x-1,y) \\ G_y(x,y) &= H(x,y+1)-H(x,y-1) \end{aligned} \quad (2)$$

3. Compute the norm value and orientation of each pixel. Expressions are as follows.

$$\begin{aligned} G(x,y) &= \sqrt{G_x(x,y)^2+G_y(x,y)^2} \\ \alpha(x,y) &= tan^{-1}\left(G_y(x,y)/G_x(x,y)\right) \end{aligned} \quad (3)$$

4. Split the input image into equally-sized cells and group them into bigger blocks. Before computing the HOG feature, the gradient magnitude is normalized within the block. In Dalal's paper, he uses L2-Hys normalization in the computation of HOG feature, however, during discussion he concludes that L2-Hys, L2-norm and L1-sqrt performed equally well. Since L2-norm is simpler than L2-Hys, L2-normalization is choose, as illustrated as follows:

$$v_i^* = v_i \Big/ \sqrt{\sum_{i=1}^{k} v_i^2 + \varepsilon} \quad (4)$$

v_i^* and v_i represent the original and normalized gradient magnitude of certain pixel i in the block respectively; k equals the total number of pixels in one block; ε is a small constant preventing the denominator from being zero, in our experiment, we set ε as 0.01.

5. After normalization the block is applied with a spatial Gaussian window with σ = 0.5 × block's width, as suggested by Dalal.

6. Trilinear interpolation is used to construct the HOG feature for each cell to obtain the low-level feature.

III. Implementation

A. HOG Parameter optimization

According to Dalal algorithm, the size of cells is 8×8 pixels and the size of block is 2×2 cells for 64×128 detection window.

But the detection window of our system is selected as 48×96, so we select 6×6 pixels for cell size and 2×2 cells for block size, it is shown in Fig.2.

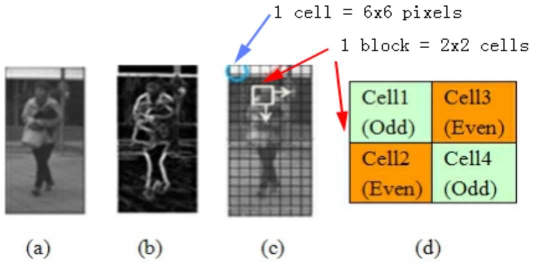

Figure 2. Cells and blocks used for HOG feature extraction : (a) input image (b) gradient image (c) cells and blocks (d) bin composition for each cell in one-block

Each cell consists of a 9-bin histogram of oriented gradients (0-180° range, 20° step size) and each block contains a concatenated vector of all 2×2 cells. In case of the original HOG, each block is thus represented by a 36-D feature vector that is normalized to an L2 unit length, each detection window is represented by 7×15 blocks with 1/2 block overlapping, giving a total of 3780 features per detection window.

In order to archive better results, different orientations are attempted. The gradient image is computed shown as Figure 3 with number of orientations equal to 3, 4, 6, 9, and 18 respectively.

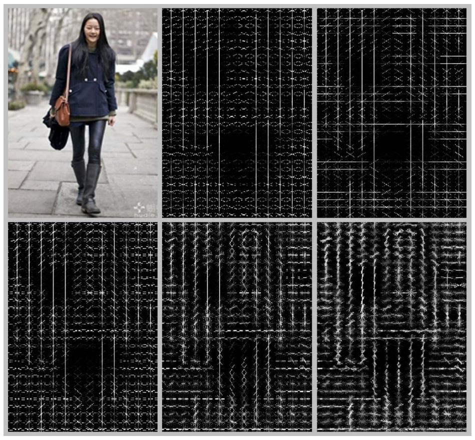

Figure 3. HOG features for number of orientations equal to 3, 4, 6, 9, and 18 repsectively.

DET (Detection Error Tradeoff curve) is used to estimate which number of orientation is better. With INRIA dataset and different orientations, the results are shown in Figure 4. We can get that when the number of orientation is set to 9, there will be better results.

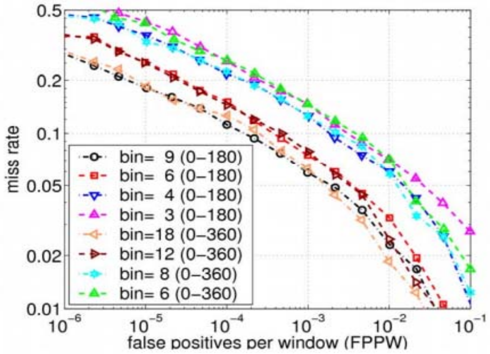

Figure 4. DET Curve with different number of orientations

B. Foreground Image

Considering how to get the moving object or region I_f in the current frame I_i, it is naturally that moving region of the object could be gotten according to the current image and the background image I_b

$$I_f = I_i - I_b$$

After getting the moving region I_f, whether human are included in the region should be recognized. We adopt the Histograms of Oriented Gradients method. However, it will cost lots of time to run HOG for each frame, so just detect each the moving region in the foreground image I_f. Each moving region is a sub-image of I_f and it is smaller than I_f, so the human in this region will be detected.

C. Multi-Human Tracking

For the purpose of counting human in the video, take human h_i for example, we must know that when h_i comes into this scene and when h_i disappears. Therefore, we track the path of h_i in this scene.

Assume that we get frame I_i and detect $|H|$ human in it, $H=\{h_1,h_2,.....h_{|H|}\}$ use $T=\{R_1,R_2,...R_{persion\_id}\}$ to save the trajectory $R=\{r_1,r_2,...r_{|R|}\}$ of each person h_{person_id}, and r_i represent the human position in I_i.

To estimate if person h_i is a new guy and which trajectory does he belong to, we must compare he to all human in T. So we need to use some features to represent a person. The most ideal features are finger print or face, however, it is impossible to get the finger print from image and it is also very difficult to read the face information.

So, we use three sample features to represent human: (1) HSV normalized histogram, (2) related position and (3) rectangle area.

Then we can compute the distance between h_i and all people in T. Here, we just use the last frame image. We define three thresholds to discriminate that if h_i is new: T_{h1}, T_{h2} and T_{h3}, then

$$\begin{cases} Exist, & if\ d_i \le Th_i, i=1,2,3. \\ New, & else. \end{cases} \quad (6)$$

If the person h_i is not a new guy, which trajectory does he belong to?

$$R^* = \arg\min (Distance(h_i, R)) \quad \forall R \in T \,. \quad (7)$$

where

$$\mathit{Distance}(\mathrm{h}_{\mathrm{i}},\mathrm{R})=$$
$$w_1\times d_1(h_i,h_{|R|})+w_2\times d_2(h_i,h_{|R|})+w_3\times d_3(h_i,h_{|R|})$$

About counting, actually, the size of T is the number of human. When implementing, we can remove the trajectory that there are no human added in. we could save the memory.

D. Experimental Results

The INRIA pedestrian dataset (available at: http://pascal.inrialpes.fr/data/human/) is used in the training and testing phase of our algorithm. The dataset contains 2416 positive training samples and 1126 positive testing samples, all the sample images have a resolution of 96 ×160.

In the experiment for static image, the detection window is set to 96×160 with number of orientation to be 9. The static picture has a resolution of 400 × 256. Average time for detection is 94ms while the time is 120ms with the algorithm shown in Reference [1]. The improved efficiency is about 22 percent.

Results that detecting human in static images are shown in Figure 5.

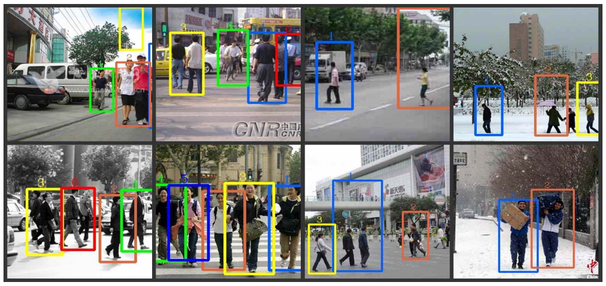

Figure 5. Some examples of our experiment results

In the experiment of object tracking, we use a video file which has a resolution of 640×480 and 30 frames per second. While processing it, 96153.7ms was used, 178ms was consuming for per frame. It is consistent to the results of static image, but for real-time requirement, it is not enough rapid, has to been improved.

IV. Conclusions

In this paper, we use the improved HOG feature to represent image, which depicts the edge features of image and reduces the impact of illumination. The improved method largely reduces computation consumption and accelerates detection speed.

According to the proposed approach which is described above, the assumption, that there is some region less informative than the others, should be considered. Depend on each object and dataset; we can decide which region is informative in image window. This approach employs non-uniform grid of point's perspective which concentrates more point into informative regions, and not concentrate on less informative ones. This does not mean that we totally ignore the background surroundings object, but we just make it less important than the regions contain object's contour and shape.

References

[1] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, In Proceedings of IEEE Conference Computer Vision and PatternRecognition. IEEE, pp. 886-893, 2005.

[2] Q. Zhu, M. Yeh, K. Cheng, and S. Avidan. Fast human detection using a cas-cade of histograms of oriented gradients. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2,IEEE, pp. 1491-1498, 2006.

[3] Yang, Sheng, Wu, Jiefa. Zhang, Lingling. A fast pedestrian detection method based on simplified HOG descriptor. International Journal of Digital Content Technology and its Applications, v6, pp. 114-122, March 2012.

[4] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multipleobject tracking using k-shortest paths optimization. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2011.

[5] Wu Jie fa, Yang Sheng, Zhang Lingling. Pedestrian detection based on improved HOG feature and robust adaptive boosting algorithm. Proceedings - 4th International Congress on Image and Signal Processing, CISP 2011

[6] Wang Bing-Bing, Chen Zhi-Xin, Wang Jia, Zhang Liquan. Pedestrian detection based on the combination of HOG and background subtraction method. 2011 International Conference on Transportation, Mechanical, and Electrical Engineering, TMEE 2011

[7] Wang Zhen-Rui, Jia Yu-Lan, Hua Huang, Tang Shu-Ming. Pedestrian detection using boosted HOG features. IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC, pp. 1155-1160.

[8] Chen Wei-Gang, Simultaneous object tracking and pedestrian detection using HOGs on contour. Signal Processing (ICSP), 2010 IEEE 10th International Conference on Date of Conference: 24-28 Oct. 2010

[9] Son Haengseon Lee, Seonyoung, Choi Jongchan. Efficient pedestrian detection by bin-interleaved Histogram of Oriented Gradients. IEEE Region 10 Annual International Conference, Proceedings/TENCON, 2010, TENCON 2010 - 2010 IEEE Region 10 Conference. pp. 2322-2325

[10] Nakashima Yuuki, Tan Joo Kooi, Ishikawa Seiji, Morie Takashi. On detecting a human and its body direction from a video. Artificial Life and Robotics, v15, pp. 455-458, 2010.

[11] Han, T.X. Shuicheng Yan. An HOG-LBP human detector with partial occlusion handling; Computer Vision, 2009 IEEE 12th International Conference; pp. 32- 39

[12] Alper Yilmaz, Omar Javed, Mubarak Shah. Object tracking: A survey; Journal ACM Computing Surveys (CSUR). Volume 38 Issue 4, 2006. Article No. 13

[13] Comaniciu, D. Mean shift and optimal prediction for efficient object tracking; Image Processing, 2000. International Conference; pp. 70-73

[14] Perera, A.G.A, Multi-Object Tracking Through Simultaneous Long Occlusions and Split-Merge Conditions; Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference; Volume: 1, pp. 666-673

[15] Robert E. Schapire, Yoram Singer, Improved boosting algorithms using confidence-rated predictions, Machine Learning 37 (3) , pp. 297–336, 1999.

[16] David Geronimo, Antonio Lopez, Daniel Ponsa, Angel D. Sappa, Haar wavelets and edge orientation histograms for on-board pedestrian detection, Lecture Notes in Computer Science 4477, pp. 418–425, 2007.

[17] David G. Lowe, Distinctive image features for scale-invariant key points, International Journal of Computer Vision 60 (2), pp. 91–110, 2004.

[18] Oncel Tuzel, Fatih Porikli, Peter Meer, Pedestrian detection via classification on riemannian manifolds, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) , pp. 1713–1727, 2008.

[19] Chistopher J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery 2, pp. 21–167, 1998.

[20] Navneet Dalal, Finding people in images and videos, Ph.D. thesis, INRIA Rhone-Alpes, 2006.

[21] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In ICCV, 2005.

[22] A. Mohan, C. Papageorgiou, and T Poggio, Example-based object detection in images by components, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, pp. 349–361, 2001.

[23] O. Tuzel, F. Porikli, and P. Meer, Pedestrian detection via classifiation on riemannian manifolds, IEEE Trans. Pattern Anal. Mach. Intell. , vol. 30, no. 10, pp. 1713–1727, 2008.