OBJECT DETECTION USING EDGE HISTOGRAM OF ORIENTED GRADIENT

Haoyu Ren   *Ze-Nian Li*

Vision and Media Lab
School of Computing Science
Simon Fraser University
Vancouver, BC, Canada
{hra15, li}@sfu.ca

ABSTRACT

In this paper, we address the object detection problem by a proposed gradient feature, the Edge Histogram of Oriented Gradient (Edge-HOG). Edge-HOG consists of several blocks arranged along a line or an arc, which is designed to describe the edge pattern. In addition, we propose a new feature extraction method, which extracts the structural information based on the gravity centers as complementary to traditional gradient histograms. As a result, the proposed Edge-HOG not only reflects the local shape information of objects, but also captures more significant appearance information. Experimental results show that the proposed approach significantly improves both the detection accuracy and the convergence speed compared to the traditional HOG feature. It also achieves performance competitive with some commonly-used methods on pedestrian detection and car detection tasks.

***Index Terms*—** Object detection, HOG, Edge-HOG, local feature, gradient histogram

1. INTRODUCTION

Object detection of a particular class is a fundamental problem of computer vision. One of the major challenges in this field is that the object appearance may vary greatly due to many factors, such as different illuminations, view points, poses, etc. This has motivated inventions of various approaches. Among them, a widely used paradigm is to train a classifier on local features using algorithms of the boosting family [1][2][3][4]. For example, Viola et al. [1] built an efficient face detector using AdaBoost algorithm to train a cascade classifier based on the Haar-like feature. Tuzel et al. [2] projected the covariance matrices to Riemannian Manifolds and further utilized LogitBoost to train a pedestrian detector.

Designing local features that can reflect the intrinsic characteristics of the object appearance is an important way in the boosting framework. In general, there are two kinds of commonly-used local features; contour features and statistic features. The contour features are usually constructed along an edge to describe the local structural information. Wu et al. proposed the edgelet feature [5], which is a short segment of edge or curve with different weight on each pixel. Gao et al. [6] proposed the adaptive contour feature based method. This feature consists of a chain of a number of granules in oriented granular space and has good discrimination power for human detection and segmentation. Statistic features extract statistic information (e.g., histograms, covariance matrix) from a local region, which has strong discrimination ability on local patterns. One of the most famous features is the Histogram of Orientation Gradient (HOG) [7] proposed by Dalal and Triggs. Inspired by this work, Shen et al. [8] proposed enhanced HOG feature that using one gradient orientation to encode all pixels in a region. Su et al. [9] proposed Local-Main-Gradient-Orientation HOG, which weighting every bin of gradient orientation histogram according to their significance within a predefined area, in order to emphasize the important gradient information.

The HOG feature ignores some important structural information in each cell. If two cells contain the same edge but at different positions, the resulting feature vectors will be the same. To solve this issue, we propose a enhanced version of HOG feature named Edge-HOG. Edge-HOG arranges the cells along an edge template to gain additional structure information. In addition, we extract a complementary feature vector based on the gravity centers, so that it captures more discriminative information than the traditional HOG does. Several experiments on public datasets are used to evaluate our method. The results show that the Edge-HOG improves both the training efficiency and the performance compared to the traditional HOG. It also achieves competitive performance with the commonly-used approaches in both pedestrian and car detection tasks.

2. EDGE-HOG FEATURE

2.1. Traditional HOG feature

The essential idea behind HOG features is that local object appearance and shape within an image can be described by

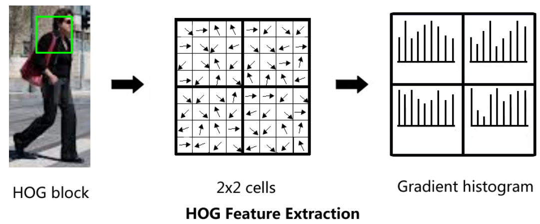


Fig. 1. Traditional HOG feature

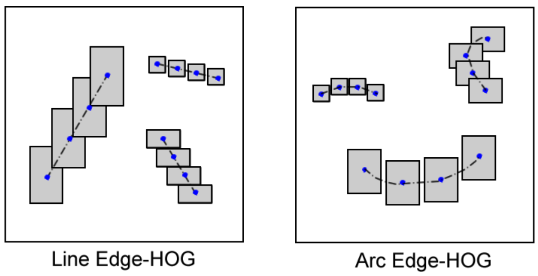


Fig. 2. Edge-HOG cell arrangement. The blue dots are the cell centers. The dot-dashed lines show the edge template concatenating these cells.

the distribution of intensity gradients or edge directions. HOG divides the image into small connected regions, called cells, and for each cell it compiles a histogram of gradient directions for the pixels within the cell. Each pixel within the cell casts a weighted vote for an orientation-based histogram channel based on the values found in the gradient computation. The histogram channels are evenly spread over 0 to 180 degrees.

The combination of these cell histograms then represents the feature vector. It will be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination or shadowing. A figure of the HOG extraction is shown in Fig. 1.

2.2. Edge-HOG feature

In this section, we introduce the proposed Edge-HOG feature. An Edge-HOG feature consists of a series of cells along an edge template, as shown in Fig. 2. The edge template (dot-dashed lines) can be lines and arcs in variable length, positions and directions. 4 pixels (blue dots) are uniformly sampled on these edges as the center pixels of Edge-HOG cells. These cells could overlap or lay far away from each other. To reduce the size of the feature pool, we add an constraint on the distance of the two neighboring center pixels d and the cell size (w, h)

$$0.75max(w, h) < d < 2min(w, h).$$

Since the cells are arranged along an edge template, the Edge-HOG not only extracts the statistical information of the local region, but also reflects the edge position and direction.

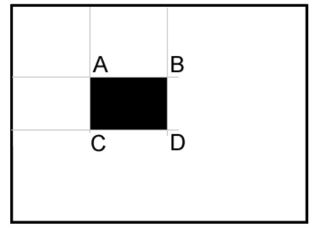


Fig. 3. Integral image

In Edge-HOG, each bin of the gradient vector is the weighted sum of all pixels with the same gradient orientation in a cell. It can be readily extracted using the integral image algorithm in [1]. The integral image is an algorithm for efficiently generating the sum of values in a rectangular subset. For an input image I, the value at point (x, y) in the corresponding integral image I' is the sum of all the pixels above and to the left of (x, y) in I. Then we could use $I'(A) + I'(D) - I'(B) - I'(C)$ to get the sum of all pixels in the rectangle (A, B, C, D) in I, as illustrated in Fig. 3. In Edge-HOG extraction, we calculate several gradient images of different gradient orientations and generate the integral images. Then we could go over all integral images to calculate the sum of the Edge-HOG cells to get the gradient feature.

Besides the cell arrangement, we induce some structure information into the feature extraction to further improve the discrimination ability of Edge-HOG. A 8-bin structure vector for each cell in the Edge-HOG feature will be calculated based on the geometric information. We first divide the cell into 8 regions according to angle of each pixel and the cell center. For each region, the gravity centers are calculated, and the distance of the gravity center and the cell center is used to generate the structure vector. As shown in Fig. 4, the cell is divided into 8 regions numbered from 0 to 7. The r_1 illustrates the euclidean distance between the gravity center of region 1 and the cell center, which will be used in the structure vector extraction. Denote the coordinate of cell center by (x_c, y_c), the ith bin of the structure vector $\mathbf{d}_{grav}$ is calculated as equation (1),

$$\mathbf{d}_{grav,i} = \frac{\sum_{(x,y)\in R_i} r_i \times grad_{x,y}}{\sum_{(x,y)\in R_i} grad_{x,y}}, \quad \ldots (1)$$

where the R_i is one of the 8 regions of each cell in Edge-HOG, $grad$ is the gradient magnitude. L1-normalization is applied on all cells after the feature extraction.

3. REALADABOOST WITH EDGE-HOG FEATURE

In RealAdaBoost [10], an image feature can be seen as a function from the image space to a real valued range $f : \mathbf{x} \rightarrow$

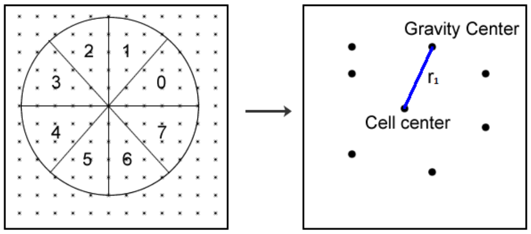


Fig. 4. Feature extraction based on structural information

$[f_{min}, f_{max}]$. The weak classifier is a function from the feature vector $\mathbf{x}$ to a real valued classification confidence space. For the binary classification problem, suppose the training data as $(\mathbf{x_1}, y_1), \ldots, (\mathbf{x_n}, y_n)$ where $\mathbf{x}_i$ is the training sample and $y \in \{-1, 1\}$ is the class label, we first divide the sample space into N_b equal sized sub-ranges B_j, the weak classifier is defined as a piecewise function

$$h(\mathbf{x}) = \frac{1}{2} ln(\frac{W_+^j + \epsilon}{W_-^j + \epsilon}), \quad \ldots (2)$$

where ϵ is the smoothing factor, $W_\pm$ is the probability distribution of the feature value for positive/negative samples, implemented as a histogram

$$W_\pm^j = P(\mathbf{x} \in X_j, y \in \{-1, 1\}), j = 1, \ldots, N_b. \quad \ldots (3)$$

The best weak classifier is selected according to the classification error Z of the piecewise function in equation (4)

$$Z = 2 \sum_j \sqrt{W_+^j W_-^j}, \quad \ldots (4)$$

In each iteration, the candidate Edge-HOG features are evaluate on all positive and negative samples. To learn the best weak classifiers, the most intuitive way is to look through the whole feature pool, which is rather time consuming. So we resort to a sampling method to speed up the feature selection process. More specifically, a random sub-sample of size $log 0.05/log 0.95 = 59$ will guarantee that we can find the best 5% features with a probability of 95%. For each candidate block, the feature vectors are extracted and further used to train a linear classify plane w* using least square. Then the final feature value used to build the probability distribution is calculated by equation

$$f(\mathbf{x}) = \mathbf{w}^* \cdot \mathbf{x} + b, \quad \ldots (5)$$

where b is the bias. Fig. 5 illustrates more details.

4. EXPERIMENTS

4.1. Experiments on INRIA pedestrian dataset

We evaluate the proposed Edge-HOG feature using the INRIA pedestrian dataset [7], which contains 1,774 human annotations and 1,671 person free images. In the experiments, we first follow the training and testing protocols [7]. Multi-scale Edge-HOG features from 8×8 to 32×32 cell sizes are utilized to train a cascade classifier for the 64×128 scanning window. 6,738 Edge-HOG features are generated following the method in section 2.2.

Parameters
N number of training samples
M number of evaluated features each iteration
T maximum number of weak classifiers

Input: Training set $\{(\mathbf{x}_i, y_i)\}, \mathbf{x}_i \in R^d, y_i \in \{-1, 1\}$

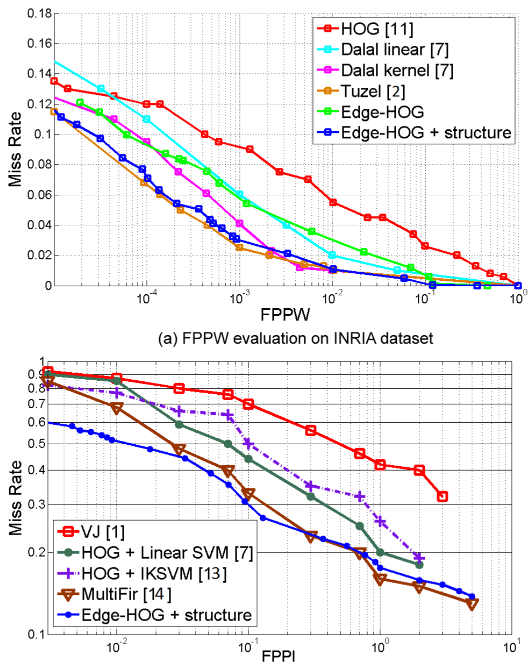
1. Initialize sample weight and classifier output
$w_i = 1/N, F(\mathbf{x}_i) = 0$
2. Repeat for $t = 1, 2, \ldots, T$
2.1 Update the sample weight w_i using the h^{th} weak classifier output $w_i = w_i e^{-y_i h_t(\mathbf{x}_i)}$
2.2 For $m = 1$ to M
2.2.1 Extract Edge-HOG features vectors
2.2.2 Train a classify plane w^* and calculate $f(\mathbf{x}_i)$ in equation (5)
2.2.3 Build the predict distribution function W_+ and W_-
2.2.4 Select the best feature which minimizes Z in equation (4)
2.3 Update weak classifier $h_t(x)$ using equation (2)
2.4 Update strong classifier $F_{t+1}(\mathbf{x}_i) = F_t(\mathbf{x}_i) + h_t(\mathbf{x}_i)$
3. Output classifier $F(\mathbf{x}) = sign[\sum_{j=1}^T h_j(\mathbf{x})]$

Fig. 5. Learning the Edge-HOG features using RealAdaBoost

In Fig. 6(a), we plot the miss rate tradeoff False Positive rate Per Window (FPPW) curves. We first compare the performances of the boosted classifiers with different cell arrangements and feature vectors. It can be seen that the Edge-HOG (green curve) shows better results compared to the traditional HOG [11] using the same gradient feature vector and AdaBoost training algorithm (red curve). If we use the Edge-HOG with the proposed structure vector, the accuracy (blue curve) is further improved. We also compare our method with three commonly-used methods, HOG with Linear SVM [7], HOG with Kernel SVM [7], and the covariance matrix with AdaBoost [2]. It can be seen that the proposed Edge-HOG method achieves detection rate of 92.9% at FPPW=10e-4, which is similar to the covariance matrix. But the computation cost is much lower.

Furthermore, we evaluate our method under the criteria of the detection rate versus False Positive rate Per Image (FPPI) [12]. Fig. 6(b) shows that our algorithm also achieves competitive result with Haar [1] and HOG [7] [13]. The accuracy is similar to the multi-feature combination [14].

Next, We compare the convergence speed of the training process in INRIA pedestrian dataset. Fig. 7 plots the FPPW against the number of weak classifiers for different



(a) FPPW evaluation on INRIA dataset

(b) FPPI evaluation on INRIA dataset

Fig. 6. Accuracy evaluation on INRIA pedestrian dataset

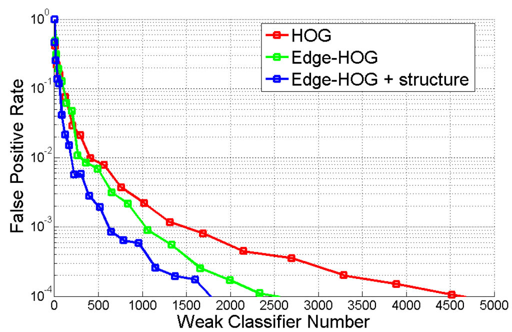


Fig. 7. Convergence speed of INRIA pedestrian dataset

methods. This figure shows that the Edge-HOG converges faster, at the rate of approximately two times faster than the HOG. When we utilize the proposed structure vector, the convergence speed is further improved. In addition, the performance of boosted classifiers is shown to be positively proportional to the convergence speed in training. This signifies that the Edge-HOG performs better on the training accuracy and speed of boosted classifiers compared to the traditional HOG.

Moreover, we investigate how the Edge-HOG captures the structure information. We list the first 5 selected features in the boosting training in Fig. 8. The series of the black rectangles in each picture represent the cells in one Edge-HOG feature. The bright background reflects the average profile of a pedestrian. From the figure, it can be seen that the first 5 selected features are basically along the profile of the pedestrian, which shows that the proposed Edge-HOG feature is very efficient to describe the structure of the human body.

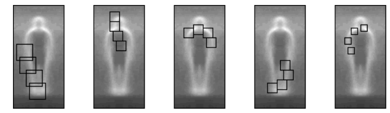


Fig. 8. The first 5 selected Edge-HOG features

Table 1. Experimental results on UIUC car dataset

Approach	single scale	multi-scale
Saberian et al. [15]	99.0%	92.1%
Xu et al. [16]	99.5%	98%
Wu et al. [5]	97.5%	93.5%
Lampert et al. [17]	98.5%	98.6%
Karlinsky et al. [18]	99.5%	98.0%
Edge-HOG	99.5%	98.6%

4.2. Experiments on UIUC car dataset

We also evaluate our algorithm on car detection task. The UIUC side view car dataset [19] is used in the experiment. This dataset contains a single scale test set (170 images with 200 cars), a multi-scale test set (108 images with 139 cars), and a training set of 550 side view car images. The car patches from the training images are resized to 64×32 pixels and horizontally flipped. We also collect 10,000 images without any cars from the internet as the negative training set. 1,764 Edge-HOG features are generated for 64×32 window.

We compare our approach with previous approaches following the Equal Precision and Recall (EPR) rate method. The results are listed in Table 1. It can be seen that our algorithm achieves competitive performance to other state-of-the-art methods on both single scale and multi-scale test sets.

5. CONCLUSION

In this paper, we proposed a novel Edge-HOG feature, which arranges the blocks along an edge to reflect the shape information. In addition, we proposed a new feature extraction method based on the local structural information as complementary to the traditional gradient histogram. Experimental results show that the convergence speed of Edge-HOG is 2 times accelerated compared to the traditional HOG. It also achieves performance competitive with the state-of-the-art methods in both pedestrian and car detections.

6. ACKNOWLEGDMENT

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under the Grant RGP36726.

7. REFERENCES

[1] Paul Viola and Michael Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*. IEEE, 2001, vol. 1, pp. I–511.

[2] Oncel Tuzel, Fatih Porikli, and Peter Meer, "Pedestrian detection via classification on riemannian manifolds," *IEEE Trans on PAMI*, vol. 30, no. 10, pp. 1713–1727, 2008.

[3] Yingdong Ma and Liang Deng, "Human detection with eoh-olbp-based multi-level features," in *Proceedings of the 3rd International Conference on Multimedia Technology (ICMT 2013)*. Springer, 2014, pp. 193–203.

[4] Charles Dubout and François Fleuret, "Object classification and detection in high dimensional feature space," 2014.

[5] Bo Wu and Ramakant Nevatia, "Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors," *IJCV*, vol. 75, no. 2, pp. 247–266, 2007.

[6] Wei Gao, Haizhou Ai, and Shihong Lao, "Adaptive contour features in oriented granular space for human detection and segmentation," in *CVPR*. IEEE, 2009, pp. 1786–1793.

[7] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *CVPR*. IEEE, 2005, vol. 1, pp. 886–893.

[8] Jifeng Shen, Changyin Sun, Wankou Yang, and Zhongxi Sun, "Fast human detection based on enhanced variable size hog features," in *Advances in Neural Networks–ISNN 2011*, pp. 342–349. Springer, 2011.

[9] Xiaoqiong Su, Weiyao Lin, Xiaozhen Zheng, Xintong Han, Hang Chu, and Xiaoyun Zhang, "A new local-main-gradient-orientation hog and contour differences based algorithm for object classification," in *ISCAS*. IEEE, 2013, pp. 2892–2895.

[10] Robert E Schapire and Yoram Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine learning*, vol. 37, no. 3, pp. 297–336, 1999.

[11] Qiang Zhu, Mei-Chen Yeh, Kwang-Ting Cheng, and Shai Avidan, "Fast human detection using a cascade of histograms of oriented gradients," in *CVPR*. IEEE, 2006, vol. 2, pp. 1491–1498.

[12] Xiaoyu Wang, Tony X Han, and Shuicheng Yan, "An hog-lbp human detector with partial occlusion handling," in *ICCV*. IEEE, 2009, pp. 32–39.

[13] Subhransu Maji, Alexander C Berg, and Jitendra Malik, "Classification using intersection kernel support vector machines is efficient," in *CVPR*. IEEE, 2008, pp. 1–8.

[14] Christian Wojek and Bernt Schiele, "A performance evaluation of single and multi-feature people detection," in *Pattern Recognition*, pp. 82–91. Springer, 2008.

[15] Mohammad Javad Saberian and Nuno Vasconcelos, "Learning optimal embedded cascades," *IEEE Trans on PAMI*, vol. 34, no. 10, pp. 2005–2018, 2012.

[16] Jingsong Xu, Qiang Wu, Jian Zhang, and Zhenmin Tang, "Object detection based on co-occurrence gmulbp features," in *ICME*. IEEE, 2012, pp. 943–948.

[17] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *CVPR*. IEEE, 2008, pp. 1–8.

[18] Leonid Karlinsky, Michael Dinerstein, Daniel Harari, and Shimon Ullman, "The chains model for detecting parts by their context," in *CVPR*. IEEE, 2010, pp. 25–32.

[19] Shivani Agarwal, Aatif Awan, and Dan Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans on PAMI*, vol. 26, no. 11, pp. 1475–1490, 2004.