Pedestrian Detection Based on HOG Features Optimized by Gentle AdaBoost in ROI

1Guo Lie, 2,*Ge Ping-shu, 3Zhao Yi-bing, 4Zhang Ming-heng, 5Li Lin-hui
1School of Automotive Engineering, Dalian University of Technology, Dalian 116024, China, guo_lie@dlut.edu.cn
*2College of Electromechanical & Information Engineering, Dalian Nationalities University, Dalian 116600, China, gps@dlnu.edu.cn
3,4,5School of Automotive Engineering, Dalian University of Technology, Dalian 116024, China

Abstract

For the purpose of locating pedestrian ahead of vehicle faster and more accurately, this paper presents a pedestrian detection method using boosted histograms of oriented gradients (HOG) features in the region of interest (ROI). These features are extracted in the regions where the pedestrian’s legs may exist. Then the gentle AdaBoost learning algorithm is adopted to select some discriminative features and to form a strong cascaded classifier to identify pedestrian. The week classifiers are optimized by the weighted fisher linear discriminant (WFLD) combined with look up table (LUT) gentle AdaBoost instead of the linear SVM, which is helpful to accelerate the training and detection processes. Experimental results indicate that this method can address the computational problems and achieve a good accuracy.

Keywords: Intelligent Transportation System, Pedestrian Protection, Histograms of Oriented Gradient, Region of Interest (ROI), Gentle AdaBoost

1. Introduction

In recent years, reducing road traffic accidents and casualties has become a widespread concerned social issue. Among these traffic accidents, pedestrians are the most vulnerable road users and are more likely to suffer from those accidents, especially under city transportation environments. The crashes between vehicle and pedestrian have killed many lives each year. The world health organization (WHO) finds that almost half of the estimated 1.27 million people who die in road traffic crashes every year are pedestrians, motorcyclists and cyclists [1]. As for China, the traffic states are not satisfied, according to the report of China Ministry of Public Security. There were 238,000 road traffic accidents in 2009, resulted in 16,683 pedestrians killed. Research on pedestrian detection can inform the driver of the presence of pedestrians ahead of vehicle in time, which is of great significance to reduce or avoid pedestrian collision [2].

Unlike other road users as automobiles, pedestrians are often found in heavy cluttered urban environment. Furthermore, the extensive variety of postures and clothes of pedestrians make this problem challenging [3, 4]. Therefore, it is difficult to describe the pedestrian using a single color, grayscale or texture feature. Recently, using the contours feature descriptors to express pedestrian becomes a trend when detecting pedestrian using machine vision. Viola et al. [5] focused on the Haar-like wavelets feature combined with AdaBoost algorithm to construct a cascade of classifiers to improve computation speed. Dalal and Triggs [6] presented the histograms of oriented gradients (HOG) features to capture the differences between human and non-human objects. Shapelet feature descriptor based on selected edges selected by AdaBoost was proposed by Sabzmeydani and Mori [7]. Then their works were extensively used by Yu et al. [8], they trained part detectors with shapelet and integrated them to verify the candidates. All these research have shown that abstracting effective features describing pedestrians can generate high performance in detecting pedestrians even in different poses, clothing, illumination, occlusion and background.

Among those features describing pedestrians, HOG features demonstrate remarkable effectiveness and have been used extensively. HOG describes the distribution of image gradients on different orientations and is implemented to capture shape and appearance feature in pedestrian detection. Their excellent performance provides lots of insights for later research. To speed up the detection, Zhu et al.
[9] used a cascade of rejecters and AdaBoost algorithm to select the features with variably-sized blocks. Although they demonstrated that the variable-size block method had higher detection accuracy than the fixed-size block method in their experiment results. The weak classifier in their approach was the linear SVM, which was time-consuming when evaluating each of the 5,301 possible blocks in each stage. Thus, they had to randomly sample 250 blocks in each round. However, even adopting the sampling method, the training process still spent a few days. Inspired by those contributions, this paper employs the variably-sized blocks and AdaBoost approach to train a strong classifier with discriminative HOG features. Those features were optimized by using the weighted fisher linear discriminant (WFLD) algorithm combined with look up table (LUT) gentle AdaBoost. This method does great contribute to the performance of our work with less selected features, meanwhile, the training and detection time is decreased significantly. In order to further accelerate the training and detecting process, a region of interest (ROI) is focused, where pedestrian’s legs may exist.

The paper is structured as follows. Section 2 explains the pedestrian detection problem and methods, where the HOG features in ROI were extracted and the LUT based gentle AdaBoost algorithm was used to optimize those features. The experiments are presented and its performance is evaluated in section 3. Finally, some brief conclusions are given in section 4.

2. Problem Description and Methods

2.1 Construction of HOG Descriptor

HOG descriptor is based on the idea that local object appearance and shape within an image can be described by the distribution of intensity gradients or edge directions. To obtain these descriptors, the image can be divided into small connected regions, named cells, and for each cell compiling a histogram of gradient directions or edge orientations for the pixels within the cell. The descriptor is the combination of these histograms. The local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, named block, and then using this value to normalize all cells within the block. This normalization is serviceable to show better invariance to changes in illumination or shadowing. Since the HOG descriptor operates on localized cells, the method upholds invariance to geometric and photometric transformations, except for object orientation. Such changes would only appear in larger spatial regions, which make HOG descriptor suitable for pedestrian detection in images [10].

A briefly explain of the HOG features extraction process is shown in Figure 1. Each image to be processed is divided into small equally-sized regions called ‘cell’. Those cells are divided into groups and integrated into a bigger region in a sliding fashion, which is called ‘block’, so blocks overlap with each other. Then the distribution of gradient magnitude in all directions in cells and blocks is calculated, which is the histograms of oriented gradients.

![Figure 1. Description of HOG feature extraction](image)

In their work of Dalal and Triggs [6], they used only fixed-size blocks of 16×16 pixels to construct the HOG features. Each block consisted of 2×2 cells, and each cell was 8×8 pixels. However, the fixed-size blocks encoded very limited information. For global information, such as the whole contour of human body, the small size of the blocks can’t capture. Therefore, Zhu et al. [9] used variable-size blocks to improve the detection performance. He considered all blocks, whose size range was from 12×12 to 64×128 pixels, then the ratio between width and height was assigned by any of the following...
ratios: (1:1), (1:2), and (2:1). The three ratios were regarded as three types of blocks, as shown in Figure 2. Finally, 5031 blocks were defined in each 64×128 detection window, each of which contains a 36-D histogram vector of concatenating the 9 orientation bins in 2×2 cell.

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

$$G_x(x, y) = f(x + 1, y) - f(x - 1, y)$$
$$G_y(x, y) = f(x, y + 1) - f(x, y - 1)$$

(1)

Then the norm and orientation of each pixel can be calculated.

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

(2)

$$\alpha(x, y) = \arctan\left(\frac{G_y(x, y)}{G_x(x, y)}\right)$$

(3)

where $G(x, y)$ and $\alpha(x, y)$ represent the norm and orientation respectively.

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 cells themselves can either be rectangular or radial in shape, and the histogram channels are evenly spread over 0 to 180 degrees or 0 to 360 degrees, depending on whether the gradient is “unsigned” or “signed”. Researches show that dividing into 9 bins over 0 to 180 degrees (“unsigned” gradient) can work best. As for the vote weight, pixel contribution can either be the gradient magnitude itself, or some function of the magnitude; in actual tests the gradient magnitude itself generally produces the best results.

Due to local illumination variance and changes in background contrast, the gradient intensity range is very large. Thus, it is useful to contrast-normalize the local responses before using them. This can be done by normalizing the local histogram all of the cells in the block. The normalized descriptor blocks can be defined as HOG descriptors. Finally, the HOG descriptors in each blocks are gathered in series into a vector which forming the feature vector.

2.2 HOG Extraction in ROI

Descriptors in blocks are represented by a 36-D feature vector, for a 64×128 detection window is represented by 5031 blocks, the dimension of feature vector is over tens of thousands. Obviously, the computation is large either in the feature extraction, classifier training, or classification. It is found that the HOG features in upper body of pedestrian, especially the head and the trunk, are relatively stable. While the background of the regions where the legs existing are usually the road surface images. When legs occupy the most part of the image, the edges are more obvious and histograms will show a maximum value in certain gradient orientation. Therefore in the lower region of the image, the HOG feature between the background and pedestrian is of significant differences. Therefore, HOG is calculated in the bottom half of the detection window, with size of 64×64. Here, a total of 1386 blocks
is defined in the detection window. The significantly reduction in dimension of the feature vector improves the training and testing efficiency.

Pedestrian’s legs ahead of vehicle usually show significant symmetry of vertical edges [11]. In order to avoid the horizontal edges’ interference of tree shadows or vehicles, the 3×3 Sobel masks are used to compute the gradients instead of the 1-D masks [-1 0 1]. Figure 3 shows the comparison of different gradient masks. As can be seen from the results, the Sobel masks can highlight the vertical edges of the pedestrian’s legs obviously.

![Images of different gradient masks](image1.jpg)

(a) Original image  (b) 1-D masks [-1 0 1] (c) 2×2 diagonal masks (d) 3×3 Sobel masks

**Figure 3.** Gradient images of different masks

### 2.3 AdaBoost Algorithm

AdaBoost is a machine learning algorithm formulated by Freund and Schapire [12], which can be used in conjunction with many other learning algorithms to improve their performance [13]. AdaBoost is adaptive in the sense that subsequent classifiers built are tweaked in favor of those instances misclassified by previous classifiers. AdaBoost calls a weak classifier repeatedly in a series of rounds from total $T$ classifiers. It starts with assigning weights to the training samples. During each iteration process, the weak classifier with the least error rate is selected, and is given a weight to determine its importance in the final classifier. Before the next iteration begins, the weights of those misclassified samples are increased so that the algorithm can focus on those hard samples. The details of the algorithm are illustrated as following.

Firstly, Given example images $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, where $y_i=0,1$ corresponds to the non-pedestrian and pedestrian respectively. Initialize the weights $w_{i,0}=D(i)$: If the sample is pedestrian, the weight $D(i)=1/k$, or else the weight $D(i)=1/l$, where $k$ and $l$ are the number of pedestrian and non-pedestrian respectively. Set the iteration number to be $T$, then for $t=1, 2, \ldots, T$, repeats the following steps:

(a) Normalize the weights $w_{t,i} = \sum_j w_{t,j,i}$, so that $w_t$ is a probability distribution.

(b) For each feature $j$, train a classifier $h_j$ which is restricted to using a single feature. The error is evaluated with respect to $w_t$. Then Choose the classifier $h_t$ with the lowest error $e_t$.

$$
e_t = \sum_i w_{t,i} \| h_j(x_i) - y_i \|$$  \hspace{1cm} (4)

(c) Update the weights according to the best simple classifier $h_j(x)$.

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_t}$$  \hspace{1cm} (5)

where $e_t=0$ if $x_i$ is classified correctly, otherwise $e_t=1$ and $\beta_t=e_t/(1-e_t)$

Finally, the strong classifier is constructed as

$$h(x) = \begin{cases} 1, & \sum \alpha_i h_i(x) \geq \sum \alpha_i / 2 \\ 0, & \text{otherwise} \end{cases}$$  \hspace{1cm} (6)

where $\alpha_t=-\log \beta_t$

The above algorithm training process shows that boosting can learn a strong classifier based on a large set of weak classifiers by re-weighting the samples. Weak classifiers are only required to be slightly better than chance. At each round of boosting, the feature-based classifier is added that best classifies the weighted training samples. With increasing of stage number, the number of weak
classifiers increases, which are needed to achieve the desired false alarm rate at the given hit rate. At last, all the weak classifiers are combined to be a strong classifier by different weight.

2.4 AdaBoost Based HOG Features Selected

Currently, more than 1386 HOG features are extracted in each 64×128 detection window. This paper intends to select a meaningful set of HOG features, which have the discriminative and distinctive properties. The gentle AdaBoost algorithm is adopted to select a small number of weighted HOG features, i.e. weak classifiers, to integrate into a strong classifier.

The performance of AdaBoost crucially depends on the choice of weak learners. Effective weak learners will increase the performance of the final classifier. The potentially large number of features prohibits the use of complex classifiers such as Genetic Algorithm [14], SVM [15] or Neural Networks. In Zhu’s work [9], the weak classifier was the separating hyperplane computed using a linear SVM. Because evaluating each of the 5,301 possible blocks in each stage was very time consuming, in practice they sampled 250 blocks, at random, in each round. Undoubtedly if the sampled 250 blocks were not well discriminated, the selected feature won’t be globally optimal, leading a decrease in detection accuracy.

An efficient choice for a multi-dimensional classifier is fisher linear discriminant (FLD). Wang et al. [16] used the FLD as a weak learner in the context of AdaBoost for face detection. A particular advantage of using FLD as a weak learner is the possibility of re-formulating FLD to minimize a weighted classification error as required by AdaBoost. Laptev [17] improved its performance using a weak learner based on weighted fisher linear discriminant (WFLD). The introduce of WFLD as an AdaBoost weak learner can eliminate the need of re-sampling training data required by classifiers that do not make use of sample weights. The WFLD can be obtained by these functions:

\[
\mu = \frac{1}{n \sum_{i=1}^{n} w_i f_i} \sum_{i=1}^{n} w_i f_i
\]

(7)

\[
S = \frac{1}{(n-1) \sum_{i=1}^{n} w_i^2} \sum_{i=1}^{n} w_i^2 (f_i - \mu) (f_i - \mu)^T
\]

(8)

\[
w^* = (\mu^* - \mu^-)/(S^* + S^-)
\]

(9)

where \(w, f\) are defined as the sample weights and feature vectors respectively, \(\mu\) is the weighted means, \(S\) is the weighted covariance matrices, \(w^*\) is the weighted fisher linear discriminant.

In practice, the distribution of image features will mostly be non-Gaussian and multi-modal. Given a large set of features, however, it can be assumed that the distribution of samples at least for some features will be close to Gaussians. According to the Bayesian classification principle, if the means and covariance of Gaussians in two classes are \(\mu_1, \mu_2\) and \(\sigma_1, \sigma_2\), the optimal classification threshold can be determined by a function as

\[
Threshold = \frac{\sigma_1 \sigma_2}{\sigma_1 + \sigma_2} \ln\left(\frac{\sigma_1}{\sigma_2}\right) + \frac{\sigma_1 \mu_2 + \sigma_2 \mu_1}{\sigma_1 + \sigma_2}
\]

(10)

However, Laptev’s AdaBoost algorithm took simple threshold-type weak classifier as the hypothesis space, which was too weak to fit complex distributions. It was only suited for the discrete AdaBoost, which output Boolean values. Compared with discrete AdaBoost, real AdaBoost and gentle AdaBoost algorithms deal with a confidence-rated weak classifier with real-valued space \(R\) instead of Boolean predictions. They are more powerful, and comparatively much more efficient. Otherwise, the WFLD
algorithm makes the classification process faster with comparatively high performance; real-valued AdaBoost algorithm is more efficient than the Boolean values. As the gentle AdaBoost is numerically more stable than real AdaBoost, it is adopted to train the final strong classifier in this paper. In gentle AdaBoost the weak classifier can be defined as

$$f(x) = P_w(y = 1 | x) - P_w(y = -1 | x)$$  \hspace{1cm} (11)$$

In order to realize the gentle AdaBoost algorithm, the posterior probability should be estimated. A look-up-table (LUT) algorithm can be adopted to calculate the posterior probability to use real AdaBoost [18]. After using WFLD , the dimension of feature vector is changed into single. According to the LUT method, the range of features is divided evenly into $n$ sub-ranges:

$$bin_j = [(j-1)/n, j/n], \quad j = 1, 2, ..., n$$  \hspace{1cm} (12)$$

Then count the number of positive samples and negative ones whose feature belongs to this range over all training samples, the posterior probability can get through these statistics.

$$W_j^{+1} = \sum_{x_i \in bin_j, y_i = 1} w_i$$
$$W_j^{-1} = \sum_{x_i \in bin_j, y_i = -1} w_i$$  \hspace{1cm} (13)$$

where $W_j^{+1}$ and $W_j^{-1}$ represent sum of the weights of pedestrian samples and non-pedestrian samples respectively which belongs to the range $bin_j$. Then posterior probability can be calculated as

$$P_w(y = 1 | x \in bin_j) = \frac{W_j^{+1}}{W_j^{+1} + W_j^{-1}}$$
$$P_w(y = -1 | x \in bin_j) = \frac{W_j^{-1}}{W_j^{+1} + W_j^{-1}}$$  \hspace{1cm} (14)$$

Finally, the LUT gentle AdaBoost weak classifier can be obtained using the formula (11).

3. Experiments and Discussion

3.1 Database Description

To evaluate the performance of the proposed method, the MIT pedestrian image dataset is handled. The MIT pedestrian dataset consists of 924 pedestrian samples with no negative. Another 500 positive samples and 1588 negative samples are manually captured from the video sequences. In our experiments, the MIT dataset and 1088 negative samples are selected for training and the others are for testing. The sample size is 64×128 pixels. Each sample is cut and only reserve the half lower part with its size of 64×64. So the positive samples are images only included pedestrian’s legs. The negative set is about some background of roads or vehicles.

3.2 Performance Evaluate

In the experiment, the performances of the following three weak classifiers were compared: the liner SVM, the WFLD with threshold-type weak classifier, and the WFLD combined with LUT Gentle AdaBoost. A set of features is selected to compose the final strong classifier with optimizing classification performance. Some features selected at first rounds of AdaBoost are shown in Figure 4,
which illustrates the emphasis of the final strong classifier on image regions with prominent appearance such as the regions of pedestrian’s left leg, right leg and feet. Both the first three selected features as shown in Figure 4(a) and Figure 4(b) contain partial prominent appearance, which make some useful information lost. But the selected features in Figure 4(c) almost contain the entire region where the pedestrian’s legs and feet exist.

Figure 4. Features selected at first rounds of AdaBoost by (a) liner SVM (b) WFLD with threshold-type weak classifier(c) WFLD combined with LUT Gentle AdaBoost

Figure 5 compares the performance of the final strong classifier. The strong classifier with liner SVM requires 47 weak hypotheses compared with WFLD requires 10. With the LUT Gentle method, the number of week classifiers decreases to 9 and the performance of the strong classifier get better. It demonstrates that the feature selected in each round is not globally optimal with sampling method of Zhu et al. [9]. Thus, the WFLD algorithm can improve this issue, by exploiting the total blocks reach the same detection rate with much fewer features, which also speed up training and testing process.

Figure 5. Performances of the resulting strong classifiers

The ROC curve for pedestrian’s legs classification with the LUT Gentle and threshold AdaBoost on our testing set is shown in Figure 6. The results indicate that the WFLD with LUT Gentle AdaBoost algorithm has better detection performance than the WFLD with threshold AdaBoost.

Figure 6. The ROC curves for the pedestrian’s legs detection on the testing set

Table.1 gives the cost time of different week classifiers in training and detection. Compared with the Liner SVM, the training times of WFLD with threshold and LUT Gentle are decreased over 77%
and 88% respectively. In addition, the detection times of the two latter methods are both significantly decreased, which highlight the great advantage of WFLD in speed. Independently analyzing the LUT method, it is found that the training time with it is twice bigger than the method without it. The reason is that in the LUT method, not only the coefficient of each region need to be calculated, but also do the maximum, minimum of features. These will take up more storage space and increase the computing. However, the convergence speed of LUT Gentle is faster than the threshold one’s, it get fewer week classifiers which can overcome the shortcoming in computing. So there is no great difference in the detection time.

<table>
<thead>
<tr>
<th>Method</th>
<th>Detection time(s)</th>
<th>Training time(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liner SVM</td>
<td>7.719</td>
<td>4979.6</td>
</tr>
<tr>
<td>WFLD with threshold</td>
<td>0.023</td>
<td>592.62</td>
</tr>
<tr>
<td>WFLD with LUT Gentle</td>
<td>0.025</td>
<td>1141.4</td>
</tr>
</tbody>
</table>

4. Conclusions

An algorithm based on HOG of ROI and LUT Gentle AdaBoost is proposed for pedestrian detection. The features are extracted in the ROI where the pedestrian's legs may exist, which is helpful to decrease the dimension of feature vector and simplify the calculation. Then the discriminative HOG features are automatically selected by the AdaBoost approach. The WFLD combined with LUT Gentle AdaBoost is implemented as the week classifiers in this work. The experiment results show that our approach has great dominants in speed of training and detection process. In addition, the LUT Gentle method can improve the performance of the strong classifier compared with the threshold-type ones. However, our works still have some deficiencies, the detection efficiency should be further improved and the situation of occluded pedestrian must be resolved. The future work will focus on incorporating motion information of pedestrian’s legs with the HOG feature to improve the detection results. A component-based approach may be implemented to detect the occluded pedestrian.

5. Acknowledgments

This research was financed by the National Natural Science Foundation of China under grant 61104165 and the Fundamental Research Funds for the Central Universities under grant DUT12JR08.

6. References


