Stereo-based Pedestrian Detection Using Two-stage Classifiers

Manabu Nishiyama, Akihito Seki, Tomoki Watanabe
Corporate Research and Development Center, Toshiba Corporation
1, Komukai-Toshiba-cho, Saiwai-ku, Kawasaki, 212-8582, Japan
manabu.nishiyama@toshiba.co.jp

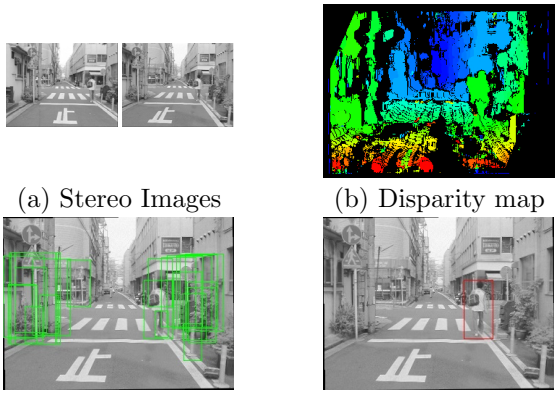
Abstract

Detecting pedestrians from a moving vehicle is a challenging problem since the essence of the task is to search non-rigid moving objects with various appearances in a dynamic and outdoor environment. In order to alleviate these difficulties, we basically use Co-occurrence Histograms of Oriented Gradients (CoHOG) as a feature descriptor. While the CoHOG feature provides high classification performance, it requires a high computational cost. In this paper we introduce another combination of a feature descriptor and a classifier for the first stage operation in order to reduce computational cost. Through experiments we show that the proposed method can reduce the calculation time while keeping the pedestrian detection capability.

1 Introduction

Human detection in images has been a central issue in computer vision and long been investigated[1, 2, 3, 4, 5]. It is difficult problem because pedestrians have many variations of size, pose and motion. In automotive area it is particularly promising because it is useful for finding pedestrians and improves traffic safety. In order to make pedestrian detection work on a practical automotive application we need to implement it on a car specific processor, where processing power and memory size are limited. Automotive application also requires that both computation time and latency is short. Thus, calculation efficiency is a crucial factor as well as pedestrian detection capability.

One of the popular approaches to pedestrian detection is classification based. In [2] Histograms of Oriented Gradients (HOG) feature descriptor and a linear SVM was used to classify pedestrian. Since the HOG and the SVM proved to be capable but time-consuming, two-staged approach was employed in [4] where the Haar-like features were used in the first stage and the HOG feature was calculated for remaining candidate regions. This paper also deals with a practical pedestrian detection based on classification and we focus on reduction of the computational cost. Our approach is similar to [4], but we use other feature descriptors in each stage, which provide better classification performance. An overview of the detection process is presented in the next section. Then, in order to reduce the calculation time we introduce another classification stage. We apply the proposed method to our original data set recorded in real environments and show that our approach can reduce computational time while pedestrian detection capability is kept.



(a) Stereo Images (b) Disparity map

(c) Detected ROIs (d) Detected pedestrian

Figure 1. Overview of our stereo-based pedestrian detection.

2 Overview of our method

Fig.1 shows an overview of our stereo-based pedestrian detection[6]. It has been compared with other methods ([3] and [5]) using the same data set and provides better performance. Our method consists of four components as follows.

Firstly, we make the disparity map of stereo input images. The input images are rectified so that the scanlines are parallel to the base line. Then, correspondence between the stereo images is calculated with a Sum of Absolute Differences (SAD) based method. Color of a disparity map in Fig.1(b) corresponds to distance from camera. Red pixel means short distance, and blue means long. Also, black pixel means unknown disparity given by cross-checking to reduce mismatching.

Secondly, we generate potential pedestrian Regions of Interest (ROIs) from the disparity map. We extract uniform disparity regions as candidates. Green rectangles in Fig.1(c) indicate detected ROIs. In this step, non-vertical regions (e.g. road regions) are eliminated and we can reduce both calculation time and the number of false detection.

Thirdly, each extracted ROI is classified whether it is a pedestrian or not by using a pattern recognition technique. Pedestrian and non-pedestrian patterns are used to learn a classifier beforehand. Pattern recognition consists of two parts: feature extraction and classification. We employ CoHOG[7] as a feature descriptor and a linear SVM as a baseline classifier. We will mention this feature descriptor in the following section. Fig.1(d) shows the classified results and red rectangles indicate detected pedestrians. Disparity information is also useful for the pattern recognition[8].

Finally, recognition results are integrated temporally[6]. Since above three steps are processed

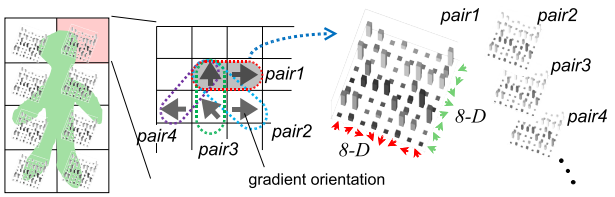


Figure 2. Overview of CoHOG feature descriptor.

frame-by-frame, it is difficult to detect pedestrians consistently through the consecutive frames. Detected pedestrians are tracked in this step, which helps the improvement of false positive and detection rate.

2.1 CoHOG feature descriptors

We use the CoHOG as a feature descriptor for pedestrian detection. The CoHOG is the extension of the HOG which is frequently selected as a feature descriptor for human detection. The calculation of the HOG is as follows. The first step is to compute a gradient orientation at every pixel $\mathbf{x} = (x, y)$. A gradient orientation θ is defined as $\theta(\mathbf{x}) = \tan^{-1} v(\mathbf{x})/h(\mathbf{x})$, where $v(\mathbf{x})$ and $h(\mathbf{x})$ are vertical and horizontal intensity gradients, respectively. The orientation $\theta(\mathbf{x})$ is then coded into eight discrete labels. We divide a ROI rectangle into several spatial blocks and create a orientation histogram for each block. Finally, all 8-D histograms are concatenated and the combined feature vector is used for classification. The orientation histogram H_i for each block is defined as

$$H_i = \#\{\mathbf{x}|\theta(\mathbf{x}) = i\}, \tag{1}$$

where $\#$ is the number of elements in the set and i is the orientation label.

In the HOG feature descriptor, we generate the histogram for gradient orientation of individual pixel. On the other hand, in the CoHOG, we handle gradient orientations *in pairs*. The CoHOG indicates the joint histogram of oriented gradients of two pixels at a certain displacement. For example, a pair of two horizontally adjacent pixels generates 8×8 dimensional histogram as shown in Fig.2. A different pair also creates another 64-D histogram. Assuming $\mathbf{d}$ to be a certain 2D displacement vector, the (i, j)-th element of the histogram H is defined as

$$H_{ij} = \#\{\mathbf{x}|\theta(\mathbf{x}) = i, \theta(\mathbf{x} + \mathbf{d}) = j\}. \tag{2}$$

In our current implementation, we use 30 pairs (30 displacement vectors) whose Chebyshev distances from each other are up to four pixels. These pairs generate a 1920 (= 64×30) dimensional feature. Combined with a HOG histogram, we get a 1928-D histogram for each block and concatenate them to generate a CoHOG descriptor. Typically we divide a candidate rectangle into 3×6 blocks. Consequently, we obtain a 34704 (=1928×3×6) dimensional feature descriptor in total. CoHOG feature descriptors have extensive vocabulary and outperform HOG features for pedestrian detection as reported in [7].

3 Adoption of Two-stage classifier

3.1 Problem of the CoHOG feature descriptor

Since the CoHOG feature descriptor can represent the shape information of the input image in detail, it provides high classification performance. However, the CoHOG is a extremely large-scale descriptor whose dimension is often much larger than the number of pixels in input image. This fact causes some problems as follows. Firstly, the large feature descriptor requires a large memory space. Required memory size increases in proportion to the number of potential pedestrian ROIs. Secondly, the CoHOG calculation requires high computational cost. In the CoHOG generation, counting gradient orientation pairs occupies the majority of calculation time. This operation takes a time in proportion to the number of the gradient orientation pairs and the size of input image. Furthermore, it is difficult to parallelize histogram creation or voting operation in general CPU architecture because voting operation includes random access to the memory. Simultaneous random memory access is not supported in well-known SIMD instruction set. In order to solve these problems, reduction of the potential pedestrian ROIs is needed.

3.2 First stage classifier to reduce ROIs

For reducing the amount of computation, the number of the CoHOG calculation needs to be cut down. We perform the first stage classification before the CoHOG calculation in order to reduce detected ROIs. The ROIs which are classified into non-pedestrian in the first stage are eliminated. We calculate the CoHOG feature descriptor and classify it using a linear SVM for the remaining ROIs. With the two-stage approach we can reduce the number of CoHOG calculation. And for reducing the total computational time, we have to employ a faster feature descriptor and a classifier than the CoHOG feature descriptor and the linear SVM classifier.

We choose joint Haar-like features and Real AdaBoost classifier[9] for the first stage. The joint Haar-like features are extension of the original Haar-like features[10]. We can use a integral image to speed up the feature extraction process. Also, the joint Haar-like features show higher classification performance than the original Haar-like features. Moreover, since joint Haar-like features are greatly different feature descriptors from the CoHOG, we can hope that the ROIs which are misclassified by the CoHOG approach may be correctly classified.

3.3 Joint Haar-like features and Real AdaBoost classifier

The joint Haar-like feature is based on the co-occurrence of Haar-like features as shown in Fig.3. The feature value of each Haar-like feature is quantized to a binary value, 1 or 0. This value shows whether the sum of the pixels within the white rectangles is larger than the one within black rectangles or not. The t-th joint Haar-like feature $J_t(x)$ is expressed by a F-bit binary number from F Haar-like features: $z_{t,1}, z_{t,2}, \cdots, z_{t,F}$. For example, when the values of individual Haar-like

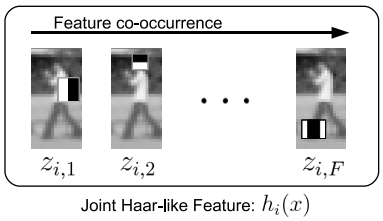


Figure 3. Example of a joint Haar-like feature.

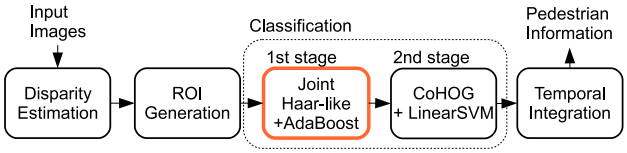
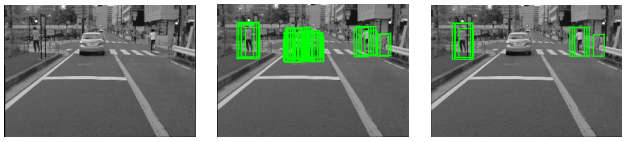


Figure 4. Pedestrian detection with two-stage classifier.



(a)Input image (b)Input ROIs of the 1st stage (c)Input ROIs of the 2nd stage

Figure 5. Potential pedestrian ROIs for the two-stage classifier.

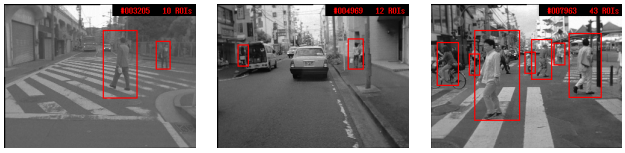


Figure 6. Example of pedestrian detection result.

features are 1, 0 and 1, the value of the joint Haar-like feature is calculated by $(101)_2 = 5$. These combined features are automatically selected by AdaBoost to capture discriminative features for the training samples. We define the t-th weak classifier based on the joint Haar-like feature as

$$h_t(x) = \frac{1}{2} \log \frac{P_t(y = +1|j) + \nu}{P_t(y = -1|j) + \nu} \tag{3}$$

where $y \in \{+1, -1\}$ is the class label (pedestrian or not), and $P_t(y = +1|j)$ and $P_t(y = -1|j)$ are class conditional probabilities of observing $J_t(x) = j$ from a sample pattern x. And ν is a small positive value, which avoids that $h_t(x)$ will be infinite. To construct the weak classifier, we need to find the best feature co-occurrence from all possible feature combinations. The conditional probability P_t is calculated by the sum of sample weights $D_t(i)$. The weight $D_t(i)$ for an i-th sample is initially set to $1/N$ where N is the number of samples. $D_t(i)$ is updated with the AdaBoost algorithm.

The final strong classifier $H(x)$ is a linear combination of T weak classifiers $h_t(x)$:

$$H(x) = sign\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{4}$$

where α is a coefficient of the weak classifier $h_t(x)$ obtained by the AdaBoost learning algorithm.

Fig.4 shows processing flow of our two-stage pedestrian detection. Joint Haar-like features extraction and Real AdaBoost classification is inserted before the CoHOG calculation. ROI reduction effect is shown in Fig.5. ROIs generated from disparity map (Fig.5(b)) are cut down by the first stage classifier (Fig.5(c)).

4 Experimental Results

In this section, we show the performance of our two-stage classifier approach through experiments. Our test sequence has 1,486 annotations of pedestrian rectangles in 20,000 frames taken with stereo cameras mounted on a vehicle. Monochrome images have been recorded at a QVGA resolution at a frame rate of 30 fps. Detection is regarded as correct if it overlaps with an annotation by more than the threshold (30%) with the intersection-over-union measure[11]. If it is less than the threshold, the detection is counted as false detection. Fig.6 shows examples of pedestrian detection result.

We have implemented the proposed method on a 3GHz Intel®Core™2 Duo processor. And we have evaluated the number of ROIs and the computational time using the test sequence. Table.1 shows the number of ROIs where the CoHOG feature was calculated and the total computational time of the whole pedestrian detection. Our previous method which use only the CoHOG feature / the linear SVM and the proposed method which has two classification stages have been compared. With the joint Haar-like features and the AdaBoost classifier the number of the ROIs input to the CoHOG calculation decreases to 9%. And the computation time of the whole processing decreases to 63%. By the two-stage classifier, we have achieved a goal of speeding up the pedestrian detection procedure.

Next, the pedestrian detection capability of the above two methods has been evaluated on the test sequence. Fig.7 shows the performance comparison of them. The horizontal axis is a false positive number per frame and the vertical one is a detection rate. The blue line indicates a performance of the conventional method, and the red one shows a performance of the proposed two-stage method. The result shows that the proposed two-stage classifier provides better performance than the conventional single CoHOG/linear SVM classifier.

Finally, we have evaluated the performance of the classifier itself. For this experiment, we have used 33,220 positive (human) and 33,220 negative (non-human) samples for training. The test data consists of 6,877 pedestrian and 50,000 non-pedestrian image regions. Green, blue and red lines in Fig.8 indicate ROC performances of joint Haar-like/Real AdaBoost, CoHOG/linear SVM and the proposed two-stage classifier, respectively. Although the Real AdaBoost classifier using the joint Haar-like features results in lower performance than the linear SVM using the CoHOG, it makes a great deal of contribution when combined with CoHOG/linear SVM.

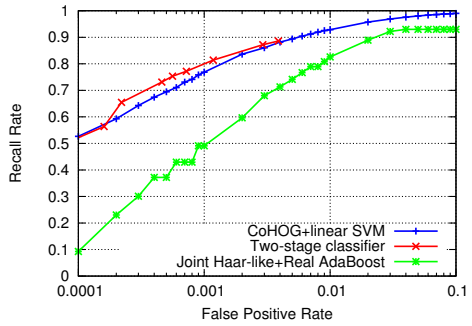


Figure 8. Comparison of the classifier performances.

Table 1. Computational time for 20,000 frames sequence.

	Total ROIs	Total time
Single CoHOG	958,331	885.9 sec
Two-stage Classifier	86,730	560.1 sec

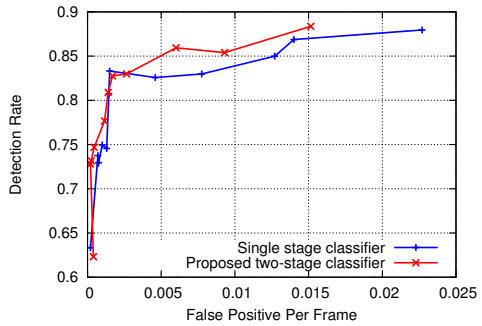


Figure 7. Comparison of the pedestrian detection performances.

5 Conclusion

In order to improve the computation efficiency of pedestrian detection, we have proposed two-stage classification approach. For the first stage we have employed the joint Haar-like features and the Real AdaBoost classifier. At the second stage, the CoHOG feature descriptor is calculated for the ROIs which have passed the first stage.

We have evaluated the efficiency of our method through the experiment using a long image sequence acquired in urban scenes. The result shows that the our method reduces the number of CoHOG calculation to about 9% and the computational time to about 63%. Furthermore, our approach improves the capability for the pedestrian detection. In future work, we will improve the memory consumption to implement the proposed method on car specific processors.

References

[1] D. M. Gavrila and V.Philomin, "Real-time object detection for "smart" vehicles," in *ICCV*, 1999, pp. 87–93.

[2] N.Dalal and B.Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005, vol. 2, pp. 886–893.

[3] A. Ess, B. Leibe, and L.van Gool, "Depth and appearance for mobile scene analysis," in *ICCV*, 2007, pp. 1–8.

[4] P.Geismann and G.Schneider, "A two-staged approach to vision-based pedestrian detection using haar and hog features," in *IV*, 2008, pp. 554–559.

[5] W. Schwartz, A. Kembhavi, D. Harwood, and L. S. Davis, "Human detection using partial least squares analysis," in *ICCV*, 2009.

[6] A. Seki, H. Hattori, M. Nishiyama, and T. Watanabe, "Temporal integration for on-board stereo-based pedestrian detection," in *WACV*, 2009, pp. 238–243.

[7] T. Watanabe, S. Ito, and K. Yokoi, "Co-occurrence histograms of oriented gradients for pedestrian detection," in *PSIVT*, 2009, pp. 37–47.

[8] H. Hattori, A. Seki, M. Nishiyama, and T. Watanabe, "Stereo-based pedestrian detection using multiple patterns," in *BMVC*, 2009.

[9] T. Mita, T. Kaneko, B. Stenger, and O. Hori, "Discriminative feature co-occurrence selection for object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 7, pp. 1257–1269, 2008.

[10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR*, 2001, vol. 1, pp. 511–518.

[11] M. Everingham and others (34 authors), "The 2005 pascal visual object class challenge," in *PASCAL Challenges Workshop*, 2006.