# Multi-channel Feature for Pedestrian Detection

Zhixiang He, Meihua Xu(✉), and Aiying Guo

School of Mechatronics Engineering and Automation, Shanghai University, No. 149, Yanchang Rd, Shanghai 200072, China
mhxu@shu.edu.cn

**Abstract.** Multi-channel feature for pedestrian detection is proposed to solve problems of real-time and accuracy of pedestrian detection in this paper. Different from traditional low level feature extraction algorithm, channels such as colours, gradient magnitude and gradient histogram are combined to extract multi-channel feature for describing pedestrian. Then classifier is trained by AdaBoost algorithm. Finally the performance of the algorithm is tested in MATLAB. The result demonstrates that the algorithm has an excellent performance on both detection precision and speed.

**Keywords:** Pedestrian detection · Image channels · Multi-channel feature · AdaBoost

## 1 Introduction

Advanced Driver Assistance Systems (ADAS) [1] has already been widely used in automobiles. As a part of ADAS, pedestrian detection plays a key role in safe driving. Pedestrian is one of the main objectives in drivers' view. Drivers need to analyze the behaviors of pedestrians and make the right decisions so as to avoid traffic accidents. Therefore pedestrian detection is a direct factor influencing safe driving.

Pedestrian detection algorithms are generally divided into two types: background modeling [2, 3] and statistical learning [3]. Background modeling firstly obtains the region of interest (ROI) which contains pedestrians. Then pedestrian features are extracted from ROI and classified. It mainly includes template matching [4], frame differential [5], optical flow [6], etc. Statistical learning requires features like color, gray, gradient and texture, which is extracted from a lot of pedestrian samples. Then it trains pedestrian detection classifiers by AdaBoost [7, 8], Support Vector Machine (SVM) [9], neural network [10], etc.

A novel pedestrian detection method based on statistical learning is proposed in this paper. As the main contribution of this paper, we extract features from multi- channel like Luv color space, gradient magnitude and gradient histogram. Integral image is introduced into the process of extracting features, which reduces a lot of computation. Features from various channels are utilized to train pedestrian classifier by AdaBoost.

The rest of the paper is organized as follows. We review related work in Sect. 2. In Sect. 3 we discuss implementation details. We perform an experimental evaluation in detail in Sect. 4 and make a conclusion in Sect. 5.


M. Fei et al. (Eds.): LSMS/ICSEE 2017, Part I, CCIS 761, pp. 472–480, 2017.
DOI: 10.1007/978-981-10-6370-1_47

## 2 Related Work

Due to the influence of complex road circumstance, a real-time and accurate pedestrian detection in front of vehicles is very difficult. Pedestrian occlusion, variable light and weather factors will make a difference to the performance of pedestrian detection. A lot of researches have been down as follows.

Dalal et al. [11] extracted HOG from pedestrian dataset and trained them with SVM. Zhu et al. [12] calculated the HOG features with integral image to reduce the operation time. Two variants of Local Binary Patterns (LBP): S - LBP and F - LBP were applied to the pedestrian detection by Mu et al. [13]. Wang et al. [14] combined the advantages of HOG and LBP to detect pedestrian when sheltered.

Inspired from the researches above, we try to describe pedestrian with features extracted from different channels. The corresponding channel of a given image is an output response. Channels can be computed by a variety of linear or nonlinear transformations. There are many available image channels such as gradient, color, texture and so on. Below we present the details of the channels used in this paper.

**Luv color space:** the CIE 1976 (L*, u*, v*) color space [15], which is also known as CIELUV, is a color space adopted by the International Commission on Illumination in 1976. A Luv image can be obtained by simple coordinate transformation from RGB image, which requires two steps.

(1) RGB to CIE XYZ

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \frac{1}{0.17697} \begin{bmatrix} 0.49 & 0.31 & 0.20 \\ 0.17697 & 0.81240 & 0.01063 \\ 0.00 & 0.01 & 0.99 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{1}$$

(2) CIE XYZ to CIE Luv

$$L^* = \begin{cases} 116\left(\frac{Y}{Y_n}\right)^{\frac{1}{3}}, \frac{Y}{Y_n} > \left(\frac{6}{29}\right)^3 \\ \left(\frac{29}{3}\right)^3 \frac{Y}{Y_n}, \frac{Y}{Y_n} \leq \left(\frac{6}{29}\right)^3 \end{cases} \tag{2}$$

$$u^* = 13L^* \cdot \left(u' - u'_n\right) \tag{3}$$

$$v^* = 13L^* \cdot \left(v' - v'_n\right) \tag{4}$$

While $u' = \frac{4X}{X + 15Y + 3Z}$ and $v' = \frac{9Y}{X + 15Y + 3Z}$ 。

**Gradient magnitude:** The gradients of each pixel in horizontal and vertical have to be calculated before the calculation of gradient magnitude. Then gradient magnitude could be obtained by formula (5).

$$M(x,y) = \sqrt{(I(x+1,y) - I(x-1,y))^2 + (I(x,y+1) - I(x,y-1))^2} \quad (5)$$

**Gradient Histograms:** The gradient angle is calculated by the tangent function of the vertical gradient and the horizontal gradient, which is shown as formula (6).

$$\theta(x,y) = \arctan\left(\frac{I(x,y+1) - I(x,y-1)}{I(x+1,y) - I(x-1,y)}\right) \quad (6)$$

Six gradient histograms are calculated in this paper. We choose every 30° of gradient angle as a gradients histogram channel. Six new images with the same size of the given image are created. Then we fill the images with gradient magnitude according to its gradient angle so as to get six gradient histograms.

# 3 Implementation Details

## 3.1 Extraction of Features

Multi-channel features are extracted as shown in Fig. 1. We acquire the output response from original image in the channels of Luv color space, gradient magnitude and gradient histogram through a variety of image transformations. Then rectangular regions with random channels, locations and sizes are selected. We define a feature as a sum of pixels in this rectangular region. The process is repeated for several times. Finally we obtain a feature pool within features from different channels.

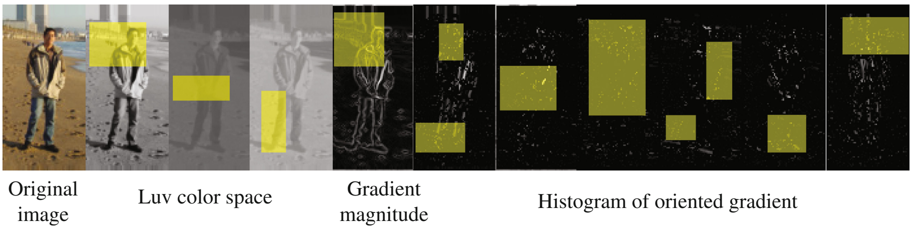


**Fig. 1.** Multi-channel features

We introduce the integral image to reduce the amount of computation in the process of extracting features. The integral image is utilized to compute rectangle features. Example of integral image is shown in Fig. 2. Firstly integral image is calculated by

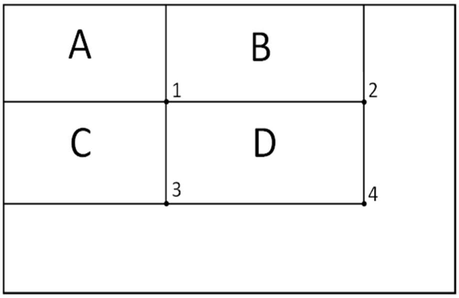


**Fig. 2.** Example of integral image

formula (7). Then we get the sum of all pixels in rectangular region D according to formula (8).

$$ii(x,y) = \sum\nolimits_{x' < x, y' < y} i(x', y') \tag{7}$$

$$S = ii(x_1, y_1) + ii(x_4, y_4) - ii(x_2, y_2) - ii(x_3, y_3) \tag{8}$$

### 3.2 AdaBoost Algorithm

As an iterative algorithm, the key idea of AdaBoost is training different weak classifiers based on the same training dataset. The classification ability of a single weak classifier is low, but several weak classifiers could be gathered as a strong classifier in high classifying ability. Then, a series of strong classifiers are concatenated to achieve an excellent performance in classification.

AdaBoost classifier is trained by changing data distribution. It modifies the weights of each sample according to previous classifying accuracy in each training dataset. The next data with new weights are trained in next level of the classifier. All classifier are combined to be the final decision classifier.

The weak classifier has the advantages of simple structure and fast classifying speed while its accuracy is low. The classification of samples is shown in formula (9).

$$h_j(x) = \begin{cases} 1, p_j f_j(x) < p_j \theta_j \\ 0, others \end{cases} \tag{9}$$

While: x represents the type of rectangular feature and $f_j(x)$ represents value of feature and $\theta_j$ is the threshold. $p_j$ equals to 1 when inputting positive samples.

The classification error of weak classifier could be calculated according to formula (10):

$$\varepsilon_j = \sum_{i=1}^{n} w_i \left| h_j(x_i) - y_i \right| \tag{10}$$

While, $w_i$ is the sample weight, n is the number of samples and $y_i$ is the sample type (positive sample equals to 1 and negative sample equals to 0).

The training process of strong classifier is shown as following steps:

(1) normalizing sample weights:

$$w_{1,j} = \frac{1}{n}, i = 1, 2, 3\ldots, n \tag{11}$$

(2) Training T round, t = 1,2,3,…, T:
   (1) Normalizing weights of samples;

$$w_{t,i} = \frac{w_{t,i}}{\sum_{k=1}^{n} w_{t,k}}, i = 1, 2, 3, \ldots, n \tag{12}$$

   (2) Choosing the weak classifier which has a best classification of x;
   (3) Updating sample weight;

$$w_{t+1,i} = \begin{cases} w_{t,i}\beta_t, x_i \; was\ classified\ correctly \\ w_{t,i}, others \end{cases} \tag{13}$$

   While $x_i$ represents sample I and $\beta_{\mathrm{t}} = \frac{\varepsilon_{\mathrm{t}}}{1-\varepsilon_t}$.

(3) Strong classifier trained:

$$h(x) = \begin{cases} 1, \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0, others \end{cases} \tag{14}$$

While $\alpha_t = -\log \beta_i$.

### 3.3 Architecture of Pedestrian Detection

The architecture of pedestrian detection system is shown in Fig. 3. In offline training section, we extract multi-channel features samples whose size is 64 * 128. Then we train classifier with features by AdaBoost.

In the process of real-time pedestrian detection, we preprocess traffic images and extract features as the input of classifier. While pre-processing images we compute multi-channel and extract features. In online detection, we input the features to the AdaBoost classifier trained offline. Finally we mark pedestrian out with rectangle boxes.

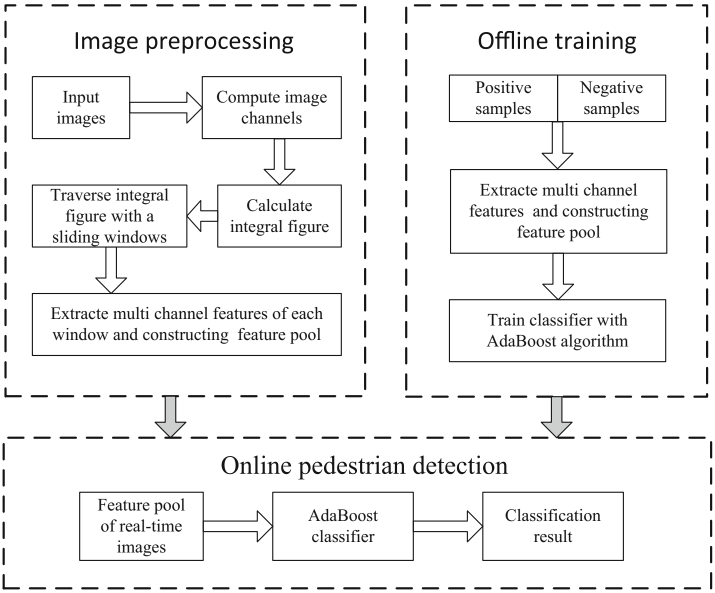


**Fig. 3.** Architecture of pedestrian detection

# 4 Experiments

## 4.1 Multi-channel

There are many kinds of multi-channel as Luv colour space, gradient magnitude and six gradient histograms could be combined in different ways. We extracted features of these multi-channel combinations from INRIA dataset and trained classifiers. We use same test samples to test each classifier and the DET curves are shown in Fig. 4.

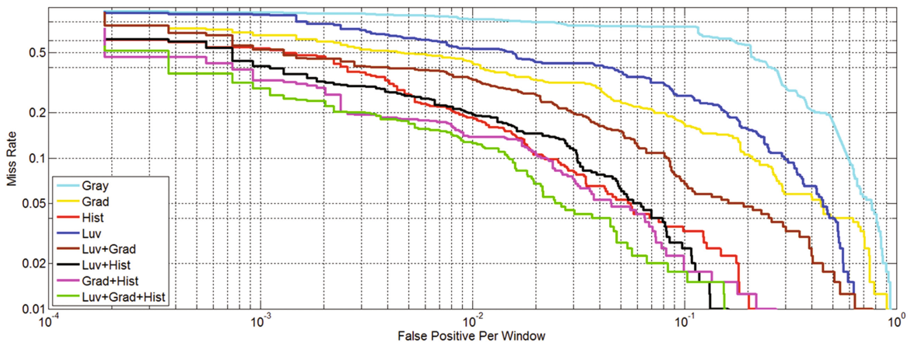


**Fig. 4.** DET curves of multi-channel feature classifiers

The test result shows that the classifier, which is trained with the features extracted from Luv, gradient magnitude and six gradient histograms, behaves the best in detection rate. Therefore, we choose these 10 channels as an optimal multi-channel compounding.

### 4.2 Experimental Data Analysis

We implement the pedestrian detection function with MATLAB on PC. Main configuration parameters of our PC are Intel 3.3 GHz processor and 8.0 GB memory. We compare the performance between HOG-SVM and this paper, which is shown in Table 1.

**Table 1.** Performance comparison between HOG-SVM and this paper.

| Algorithm | Detection rate | Detection time |
|---|---|---|
| HOG-SVM | 89% | 7.5 s |
| This paper | 93.75% | 1.97 s |

It could be found from the data in Table 1 that the algorithm has an obvious improvement on detection rate while compared with the classical pedestrian detection algorithm HOG-SVM. Furthermore the detection speed is 4 times more than HOG-SVM. The improvement of speed and accuracy in this paper mainly comes from feature extraction. Although HOG is an important feature describing pedestrian, a single feature can hardly describe pedestrian completely. Multi-channel feature used in the paper covered multiple aspects of pedestrian, such as like color, gradient and so on, which has a better performance on detection rate. On the other hand, HOG needs to traverse the entire detection window for exhaustive extraction. Large amount of computation causes a long detection time. But in this paper we randomly select a rectangular area in detection window and calculate features with integral image, which greatly reduces the amount of calculation. So it achieves a remarkable performance on detection time.

We design a pedestrian detection classifier on MATLAB and input traffic images. The detection results are shown in Fig. 5.

As it shows in Fig. 5, the algorithm has a nice detection effect on the main pedestrian targets in front of the vehicle. But it is difficult to detect the small pedestrian targets in the distance. As those pedestrians are far away from the vehicle, they make little influence on the safe driving. So the algorithm is feasible in practical application.

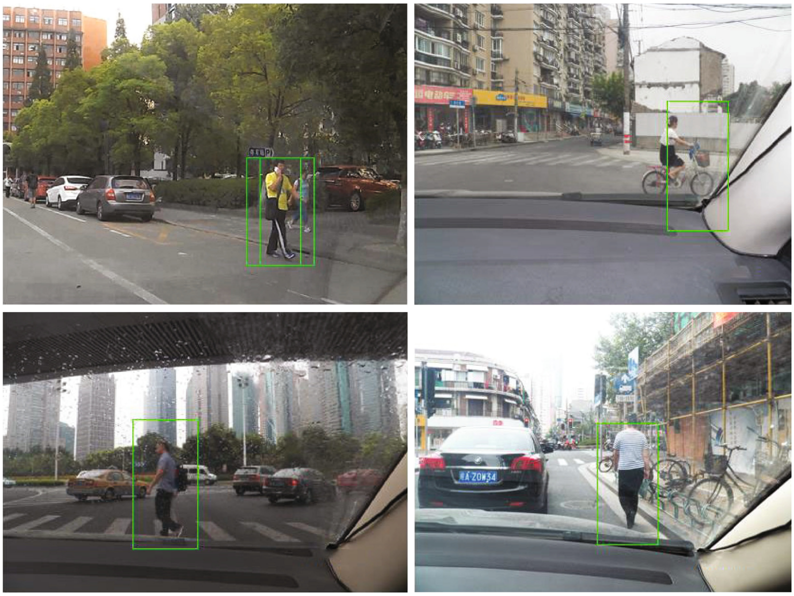

**Fig. 5.** Detection results

## 5 Conclusion

We optimize feature extraction of pedestrian and study a pedestrian detection method based on multi-channel feature. The pedestrian detection function is implemented in MATLAB. The multi-channel features of real time traffic image are extracted with integral image. Features are input to AdaBoost classifier for fast classification. The experimental result shows that the algorithm used in the paper has improved significantly in both detection speed and detection accuracy when compared with the classical algorithm HOG+SVM. It contains practical application value in pedestrian detection.

**Acknowledgement.** This work was financially supported by the national natural science foundation of China under Grant (No. 61376028) and (No. 61674100).

## References

1. Gietelink, O.J., Ploeg, J., De Schutter, B., et al.: Development of advanced driver assistance systems with vehicle hardware-in-the-loop simulations. Veh. Syst. Dyn. **44**(7), 569–590 (2006)
2. Zhen, L.I., Wei, Z.Q., Xiao-Peng, J.I., et al.: Pedestrian detection based on adaptive background modeling. J. Syst. Simul. (2009)
3. Yand, T., Li, J., Pan, Q., et al.: Scene modeling and statistical learning based robust pedestrian detection algorithm. Acta Autom. Sin. **36**(4), 499–508 (2010)

4. Hao, Z., Wang, B., Teng, J.: Fast pedestrian detection Based on Adaboost and probability template matching. In: International Conference on Advanced Computer Control, pp. 390–394. IEEE (2010)
5. Meng, X.Z., Xing, J.P., Wang, Y.Z., et al.: Pedestrian detection using frame differential method and improved HOG feature. Adv. Mater. Res. **461**, 7–12 (2012)
6. Hariyono, J., Hoang, V., Jo, K.: Moving object localization using optical flow for pedestrian detection from a moving vehicle. Sci. World J. Article id. 8196415 (2014)
7. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision Pattern Recognition, pp. 511–518 (2001)
8. Viola, P., Jones, M.: Robust real-time face detection. Int. J. Comput. Vis. **57**(2), 137–154 (2004)
9. Adankon, M.M., Cheriet, M.: Support Vector Machine. Comput. Sci. **1**(4), 1–28 (2002)
10. Zhao, L., Thorpe, C.E.: Stereo- and neural network-based pedestrian detection. IEEE Trans. Intell. Transp. Syst. **1**(3), 148–154 (2000)
11. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Computer Vision and Pattern Recognition, pp. 886–893 (2005)
12. Zhu, Q., Yeh, M., Cheng, K., et al.: Fast human detection using a cascade of histograms of oriented gradients. In: Computer Vision and Pattern Recognition, pp. 1491–1498 (2006)
13. Mu, Y., Yan, S., Liu, Y., et al.: Discriminative local binary patterns for human detection in personal album. In: Computer Vision and Pattern Recognition, pp. 1–8 (2008)
14. Wang, X., Han, T.X., Yan, S., et al.: An HOG-LBP human detector with partial occlusion handling. In: International Conference on Computer Vision, pp. 32–39 (2009)
15. CIELUV. https://en.wikipedia.org/wiki/CIELUV Accesed 30 March 2017
16. Wang, Z., Yoon, S., Xie, S.J.: A high accuracy pedestrian detection system combining a cascade adaboost detector and random vector functional-link net. Sci. World J. Article id. 7105089 (2014)
17. Zhang, S., Benenson, R., Schiele, B., et al.: Filtered feature channels for pedestrian detection. In: Computer Vision and Pattern Recognition, pp. 1751–1760 (2015)