IJBSCHS
[Original Article]

Biomedical Soft Computing and Human Sciences, Vol.19, No.1, pp.33-41

(Accepted on 2014.1.17)

Detection and Tracking of a Human on a Bicycle Using HOG Feature and Particle Filter

Heewook JUNG, Joo Kooi TAN, Hyoungseop KIM, Takashi MORIE and Seiji ISHIKAWA

Department of Mechanical and Control Engineering, Kyushu Institute of Technology

Abstract: *Detection of a human on a bicycle is an important research subject in an advanced safety vehicle driving system to decrease traffic accidents. The Histograms of Oriented Gradients (HOG) feature has been proposed as useful feature for detecting a standing human in various kinds of background. So, many researchers use currently the HOG feature to detect a human. Detecting a human on a bicycle is more difficult than detecting a standing human, because the appearance of a bicycle can change dramatically according to viewpoints. In this paper, we propose a method of detecting a human on a bicycle using HOG feature and RealAdaBoost algorithm. When detecting a human on a bicycle, occlusion is a cause of decreasing detection efficiency. Occlusion is a serious problem in car vision research, because there are often occlusion in real transportation environment. In such a case, the proposed method predicts the next position of a human on a bicycle using a tracking strategy. Experimental results and their evaluation show satisfactory performance of the proposed method.*

Keywords *Bicycle detection, HOG feature, RealAdaBoost, Object tracking, Occlusion, Particle filter, Advanced safety vehicle.*

1. Introduction

The number of entire traffic accidents is currently decreasing due to various safety mechanisms in an automobile. Advanced safety systems like EyeSight of SUBARU and Pedestrian Detection with Full Auto Break System of VOLVO are deployed on a commercial scale. We call these cars as ASV (Advanced Safety Vehicle). These systems are using stereo cameras, wide angle cameras and a radar to detect pedestrians, cars, etc. But these systems have a limit to vehicle speed for stable performance of object detection. Actually the limit speed of SUBARU is 30 km/h and that of VOLVO is 35 km/h at the highest for effective operations of their safety systems. Advanced safety systems still have some problems of making miss detection and false detection. The main reason of miss detection and false detection are the existence of occlusion, cluttered background and image degradation.

To prevent traffic accidents, it is necessary to detect the objects having the risk of traffic accidents (like a human, a car, a bicycle, etc.) and to utilize those factors that contribute to safety driving (like ITS (Intelligent Transport Systems)) [12],[14],[15],[16]. Here, we keep observation upon the object detection. In this paper, we focus on detecting a bicyclist and a bicycle among various objects on the road. Currently, many researchers are intensively studying on pedestrian detection [2],[10],[13],[17]. Bicycle accidents, however, frequently occur in a transportation environment, because drivers have difficulty in predicting a bicycle's driving direction and a bicycle is sometimes more dangerous than a pedestrian because of its speed. Moreover, detecting a bicycle is more difficult than detecting a pedestrian, because a bicycle's appearance can change dramatically according to viewpoints [11],[19],[20].

1-1 Sensuicho, Tobata, Kitakyushu,
Fukuoka 804-8550 Japan
Phone and Fax: +81-93-884-3183
e-mail: jung@ss10.cntl.kyutech.ac.jp

In this paper, we propose a method of detecting a human riding a bicycle using HOG (Histograms of Oriented Gradients) feature and RealAdaBoost. The proposed method can also detect bicycles' driving directions. The reason for detecting bicycles' driving directions is to judge if a bicycle comes suddenly into a driver's sight.

We also propose a method of tracking a detected human on a bicycle using particle filter. In real transportation environment, there are many cases of occlusion. Occlusion is a fatal factor that may cause traffic accidents. We employ particle filter for the tracking of a detected human on a bicycle, because it is robust to a non-linear system. Movement of objects in car vision is sometimes non-linear. The proposed method can track objects invisible to a driver. The proposed method is described in the following and the performance of the method is shown by the experiments employing real outdoor scenes.

This paper is organized as follows. Section 2 introduces the learning algorithm of the proposed method and section 3 describes a detection algorithm of a bicycle. We mention a tracking algorithm of the bicycle in section 4. In section 5, experimental results are shown. Finally, discussion and the conclusion are given in section 6.

2. Learning Algorithm

A learning algorithm is used to HOG feature and RealAdaBoost. The proposed method detects not only a bicycle but also bicycles' driving directions by the employment of three classifiers that learn its driving directions, i.e., front, left and right, by training samples. The used learning data is collected on the internet, some of which are shown in Fig.1.

A. HOG feature

The HOG feature is a feature computed from edge gradients in an image [1]. Because the HOG feature receives normalization at each area, it is robust to illumination change.

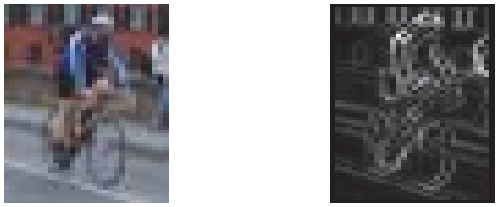
The HOG feature is extracted in the following way.

Step1: Compute the magnitude $m(u, v)$ and the orientation $\theta(u, v)$ at pixel (u, v) using Eqs.(1) and (2).

$$m(u,v) = \sqrt{f_u(u,v)^2 + f_v(u,v)^2} \ . \qquad (1)$$

(a) (b) (c) (d)

Fig. 1. Learning data : (a) Left direction, (b) right direction, (c) front direction, and (d) background.



(a) (b)

Fig. 2. A processed bicycle image: (a) An original image, and (b) its oriented gradients image.

$$\theta(u,v) = \tan^{-1}\frac{f_u(u,v)}{f_v(u,v)} \ . \qquad (2)$$

Here, $f_u(u,v)$ is a horizontal edge and $f_v(u,v)$ is a vertical edge defined by.

$$\begin{aligned} f_u(u,v) &= f(u+1,v) - f(u-1,v) \\ f_v(u,v) &= f(u,v+1) - f(u,v-1) \end{aligned} \ . \qquad (3)$$

Here, $f(u,v)$ is a brightness value. An example of an oriented gradients image is shown in Fig.2 (b).

Step2: Derive the orientation histogram from the orientations and magnitudes at each cell to decrease the amount of information that the oriented gradients image possesses.

(a) (b)

Fig. 3. Merging process: (a) Before merging, and (b) after merging.

Step3: Perform histogram's normalization with every overlapping block. Here, each block size is 3×3 cells and each cell size is 5×5 pixels.

In this paper, we use integral histogram to reduce calculation cost of features[7].

B. RealAdaBoost

The RealAdaBoost algorithm, is a method of uniting weak classifiers each of which having a simple hypothesis and generating a strong classifier [4]. The HOG feature model of a bicycle is defined and trained via RealAdaBoost to detect bicycles under various circumstances. The algorithm of RealAdaBoost is given in Appendix 1. A strong classifier for a sample $\boldsymbol{x}$ is defined as the sign of the sum of weak classifiers, as described by Eq. (4)

$$H(\boldsymbol{x}) = sign\left(\sum_{t=1}^{T} h_t(\boldsymbol{x})\right). \qquad (4)$$

In the training using RealAdaBoost, we use totally 3,600 positive training samples (1,200 samples directing front, 1,200 samples directing left and 1,200 samples directing right) and 6,000 negative samples.

3. Detection Algorithm

A bicycle, if it exists, is detected from an unknown image by applying the strong classifier defined by Eq. (4) to a window which scans the image. Once a bicycle has been detected, its driving direction is judged by using respective strong classifiers. When the detection algorithm is performed, normally several windows containing an identical bicycle are obtained on an image. Then we merge them into a single window using the mean shift clustering and the nearest neighbor algorithm [8].

A. Mean shift clustering

The first step of mean shift clustering is the assumption of density using a kernel function. Here we use a kernel function represented by

$$k(x) = \begin{cases} c(1-\|x\|) & \|x\| < 1 \\ 0 & otherwise \end{cases} . \qquad (5)$$

The second step is shifting the mean to a high density location employing a mean shift vector. A mean shift vector is given as follows;

$$m(x) = \frac{\sum_{i=1}^{n} x_i k\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} k\left(\left\|\frac{x - x_i}{h}\right\|^2\right)} - x . \qquad (6)$$

Here, *n* is the center's number of detected windows and *h* is a radius of detected windows. The first term of the mean shift vector assumes a density and the second term is a shifted mean. When the mean shift vector is 0, the density is the highest.

B. Nearest neighbor

Euclidean distance *d* is computed from the point of the highest density to each sample point.

$$d = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad . \qquad (7)$$

Merging is executed, if *d* is smaller than a predefined threshold. The result of the merging process is shown in Fig.3.

4. Tracking Algorithm

In a real transportation environment, objects movement is often non-linear and the background including objects is in cluttered environments. Particle Filter (PF) is receiving attention in object tracking research, because it has been successfully applied to the tracking of moving objects in cluttered environments [5],[6]. The objective of the PF algorithm is to estimate the target location by approximating posterior distributions of a set of weighted samples (called particles). The principal steps of the PF algorithm are described in the following.

Step1: Initialization

Initialize the weighted samples (particles) set $S = \{s_0^{(n)}, w_0^{(n)} |\}_{n=1}^{N}$.

Here, $s = \{x, y, \dot{x}, \dot{y}, H_x, H_y\}$ and w is the weight of a sample, and they are initialized to *1/N*. Positions (*x*, *y*) of all the samples are initialized to the center of a detected bicycle's region. H_x and H_y are the width and the height of the detected bicycle's

region, respectively.

Step2: Selection

Select *N* samples from the set $S_t^{(n)}$ (*N* samples of S_t distributed at frame *t*) with probability $w_{t-1}^{(n)}$.

(a) Compute the normalized cumulative probability $c_{t-1}^{'(n)}$ by Eq. (8).

$$\begin{aligned} c_{t-1}^{(0)} &= 0. \\ c_{t-1}^{(n)} &= c_{t-1}^{(n-1)} + w_{t-1}^{(n)}., \\ c_{t-1}^{'(n)} &= c_{t-1}^{(n)} / c_{t-1}^{(N)} . \end{aligned} \tag{8}$$

(b) Generate a uniformly distributed random number $(r \in [0,1])$.

(c) Find the smallest *j* for which $c_{t-1}^{'(j)} \geq r$ by binary search.

(d) Set $S_{t-1}^{'(n)} = S_{t-1}^{(j)}$.

Step3: Propagation

Propagate each sample from the re-sampling set $S_{t-1}^{'}$ as follows.

$$S_t^{(n)} = AS_t^{'(n)} + G_{t-1}^{(n)}. \tag{9}$$

$$\begin{pmatrix} x_t \\ y_t \\ \dot{x}_t \\ \dot{y}_t \end{pmatrix} = \begin{pmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ \dot{x}_{t-1} \\ \dot{y}_{t-1} \end{pmatrix} + \begin{pmatrix} g_{t-1}^{x} \\ g_{t-1}^{y} \\ g_{t-1}^{\dot{x}} \\ g_{t-1}^{\dot{y}} \end{pmatrix}.$$

Here, G_{t-1} is a multivariate Gaussian random variable set.

Step4: Observation

Observe the color distributions.

(a) Compute the color density function $p_{S_t^{(n)}}^{(u)}$ for each sample of the set S_t by Eq. (10)

$$p_{S_t^{(n)}}^{(u)} = f_1 \sum_{i=1}^{I} k\left(\frac{\left\| S_t^{(n)} - X_i \right\|}{a} \right) \delta[h(X_i) - u]. \tag{10}$$

Here, f_1 is a normalized constant defined by Eq.(10), *I* is the pixels' total number of a searching area, *k* is a kernel function defined by Eq.(11), *a* is a scale factor, X_i is the pixel location of the searching area, *h* is a color histogram of *M*-bin, and *u* is the m^{th} bin.

$$f_1 = 1 / \sum_{i=1}^{I} k\left(\left\| \frac{S_t^{(n)} - X_i^*}{a} \right\|^2 \right) . \tag{11}$$

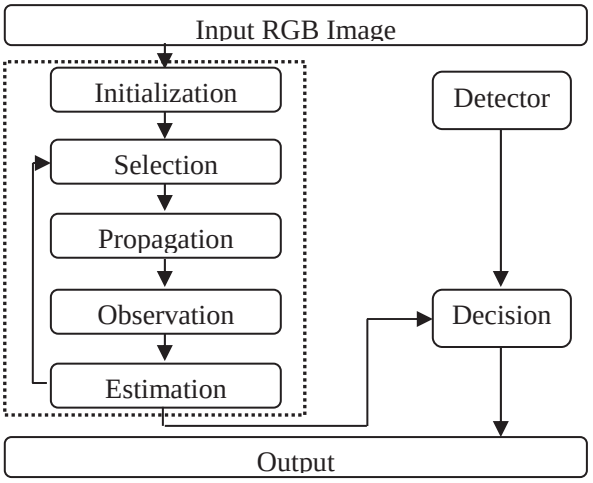


Fig. 4. The proposed tracking algorithm using a PF.

Here, X_i^* is a normalized pixel location that sets the coordinate of the image to 0.

$$k(r) = \begin{cases} 1 - \|r\|^2 & \|r\| < 1 \\ 0 & otherwise \end{cases} \tag{12}$$

(b) Compute the Bhattacharyya coefficient ρ for each sample set S_t by Eq.(13)

$$\rho[p_{S_t^{(n)}}, q] = \sum_{u=1}^{m} \sqrt{p_{S_t^{(n)}}^{(u)} q^{(u)}} \tag{13}$$

Here, *q* is a color distribution density function given by Eq.(14).

$$q^{(u)} = f_2 \sum_{i=1}^{I} k(\left\| X_i^* \right\|^2 \delta[b(X_i^*) - u]) \tag{14}$$

Here, f_2 is a constant defined by

$$f_2 = 1 / \sum_{i=1}^{I} k\left(\left\| X_i^* \right\| \right) \tag{15}$$

(c) Give weight to each sample of the set S_t by Eq.(16).

$$w_t^{(n)} = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(1-\rho[p_{S_t^{(n)}}, q])}{2\sigma^2}} \tag{16}$$

Step5: Estimation

Estimate the position of the samples using weighted samples by Eq.(17).

$$E[S_t] = \sum_{n=1}^{N} w_t^{(n)} S_t^{(n)} \tag{17}$$

Tracking performance of the PF is generally affected by the background colors similar to the target. We propose a method that enhances tracking performance of color-based PF. The proposed method is given in Fig.4.

Conventional researches use only a detector when detecting objects (pedestrians, bicycles) [3], [9], [18], or, after the objects detection using a detector, they conduct object tracking using a tracker [12], [16]. The proposed method obtains a position and scale information of a bicycle detector and a bicycle tracker in every frame.

We decide bicycles' scale BI^s by combining the information on a PF and a detector by Eqs.(18) and (19), respectively.

$$BI^s = (PF_t^s + Mean\_DT_t^s)/2 \quad if \quad \left|PF_t^s - (Mean\_DT_t^s)\right| < th_1 \tag{18}$$

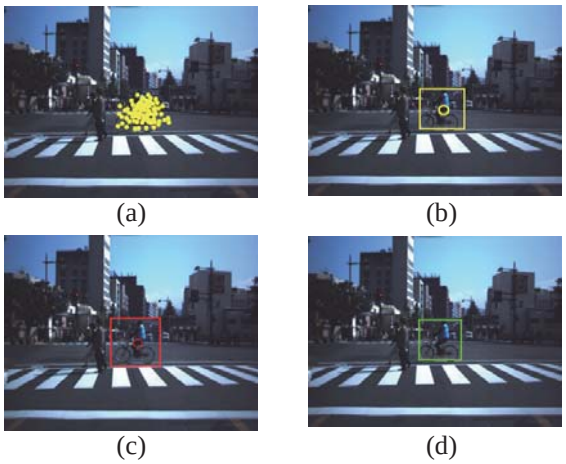
$$BI^s = Mean\_DT_t^s \quad if \quad \left|PF_t^s - (Mean\_DT_t^s)\right| > th_1 \tag{19}$$

Here, $Mean\_DT_t^s = (DT_{t-2}^s + DT_{t-1}^s + DT_t^s)/3$, PF_t^s is the scale information of PF, and DT_t^s is the scale information of a detector.

If $\left|(PF_t^p - DT_t^p)\right| < th_2$, we decide the bicycles' position as BI^p by combining the information of the PF and the detector by Eq.(20).

$$BI^p = (PF_t^p + DT_t^p)/2 \tag{20}$$
$$if \quad PF_t^p = 0, \text{ then } BI^p = DT_t^p.$$
$$if \quad DT_t^p = 0, \text{ then } BI^p = PF_t^p.$$

Here PF_t^p is the position information of a PF, and DT_t^p is the position information of a detector.



(a) (b) (c) (d)

Fig. 5. Process of the bicycle detection with tracking: (a) Particle distribution, (b) result of tracking, (c) result of detection, (d) result of detection with tracking.

Fig. 6. Setting of video cameras.

If $\left|PF_t^p - DT_t^p)\right| > th_2$, we recalculate the bicycles' position by Eq.(20) by applying revalues that renew the information of bicycles' position by Eq.(21).

$$PF_t^p = PF_{t-1}^p + ((PF_{t-3}^p - PF_{t-2}^p) + (PF_{t-2}^p - PF_{t-1}^p))/2$$
$$DT_t^p = DT_{t-1}^p + (DT_{t-3}^p - DT_{t-2}^p) + (DT_{t-2}^p - DT_{t-1}^p))/2 \tag{21}$$

The proposed method can detect a bicycle more precisely because bicycle's information is obtained by a tracker when a detector cannot detect bicycles and by a detector when a tracker cannot track bicycles. The processes of bicycles detection with tracking of the proposed method is shown in Fig.5. Fig.5(a) shows particle distribution with a bicycle that is tracked by the tracker. Fig.5(b) is a result of tracking using the tracker and (c) is a result of detection using the detector. Final position and scale are determined using Eqs.(18) ~ (21) based on the estimation result of a bicycle's position and scale by the tracker and the estimation result of a bicycle's position and scale by the detector as shown in Fig.5(d).

5. Experimental Results

A. Experimental setup

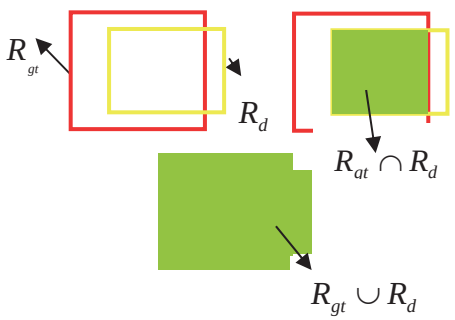


Fig. 7. Criterion for the judgment of success in the detection.

Table 1. Specification of a used PC and a camera.

OS	Microsoft Windows 7 Home Premium 64bit
CPU	Intel®Core™ i5 2.53GHz
Memory	4.0GB
Camera	SONY HDR-HC7
Software Tool	Microsoft Visual 2008 C++

In the experiment, we used real traffic scene images that we collected through a camera installed at the front seat of a car as shown in Fig.6. Specification of the camera and the PC is given in Table 1.

We did three kinds of experiments to verify the performance of the proposed method. In the first experiment, we conducted bicycle detection and tracking using only the HOG feature, that is, the HOG feature was applied to every frame of the video to detect and track a bicycle. On the other hand, in the second experiment, once a bicycle was detected in an initial frame using the HOG feature, it was tracked using the PF in successive frames. With each of these two experiments, three cases (videos) were used for examining the proposed method. The used images are 90 frames for Case 1, 100 frames for Case 2 and 350 frames for Case 3.

In the third experiment, the direction of a bicycle was judged using the HOG feature. In the experiment, 150 frames were used for the left direction, 195 frames for the right direction and 150 frames for the front direction.

B. Experimental results

Experimental results are appended below at Fig. A1 and Fig. A2 of Appendix 2. Fig. A1 of Appendix 2 is the experimental results on bicycle detection with tracking in real traffic scenes, whereas Fig. A2 of Appendix 2 is the result on the detection of bicycle's driving direction. The results are marked using colored rectangles.

Precision, *recall* and *FPR* are respectively computed by Eq. (21).

$$\begin{aligned} precision &= TP \times 100/(TP+FP)\ [\%] \\ recall &= TP \times 100/(TP+FN)\ [\%] \\ FPR &= 100 - precision\ [\%] \end{aligned} \quad , \qquad (21)$$

where *TP* is True Positive, *FP* is False Positive, *FN* is False Negative and *FPR* is False Positive Rate.

As shown in Fig. 7, we judge success or failure using the index defined by

$$\delta = \frac{R_{gt} \cap R_d}{R_{gt} \cup R_d}. \qquad (22)$$

Here, R_{gt} is a ground truth area that contains a bicycle and R_d is an area that is detected using the proposed method. If δ is 0.5 or more, we judge that the detection is successful.

6. Discussion and Conclusion

In this paper, we proposed a method of detecting a bicycle using HOG feature and RealAdaBoost algorithm in company with a tracking algorithm using PF. This method can not only detect a bicycle but also detect a bicycle's driving direction and it is applicable to the detection of various objects.

Table 2 is the result (Exp.1) of bicycles detection employing only HOG feature and RealAdaBoost classifier in real outdoor scenes. In real traffic scenes, bicycle's detection is not simple because real outdoor scenes often include occlusion of bicycles and many objects exist which have similar HOG feature to a bicycle.

Table 2. Evaluation on the bicycle detection without PF tracking (Exp.1).

	precision [%]	*recall* [%]	*FPR* [%]
Case 1	96.0	91.0	4.0
Case 2	81.7	76.8	18.3
Case 3	88.1	53.3	11.9

Table 3. Evaluation on bicycle's detection with PF tracking (Exp.2).

	precision [%]	*recall* [%]	*FPR* [%]
Case 1	96.0	86.2	4.0
Case 2	87.6	81.3	12.4
Case 3	89.0	70.6	11.0

Table 4. Evaluation on the detection of bicycle's driving direction (Exp.3).

	precision [%]	*recall* [%]	*FPR* [%]
Left	93.0	80.0	7.0
Right	94.1	79.7	5.9
Front	93.5	79.1	6.5

Therefore, the recall of Case 2 and Case 3 is not high. Especially, the recall of Case 3 in Table 2 is low because occlusion often occurred in Case 3.

Table 3 is the result (Exp.2) of bicycles detection using HOG feature together with a PF tracker. The proposed method achieved a better result than the case using only HOG feature (Exp.1). As you see in Table 3, the recall of Case 3 has been improved largely in comparison with Case 3 in Table 2, because the PF tracker successfully tracked the bicycles that the detector based on the HOG feature cannot detect when occlusion exists as shown in Fig. A1.

Table 4 is the result of the detection of bicycle's driving direction (Exp.3). The proposed method can detect three directions, because we use training data of the three directions. The recall value of front direction detection is low, because a bicycle of too small size is difficult to detect its direction.

In future, we plan to enhance the performance of the detection and tracking of a bicycle, and to develop a method which detects various objects that might become risky factors for traffics.

ACKNOWLEDGMENT

This study was supported by JSPS KAKENHI Grant Number 25350477, which is greatly acknowledged.

References

[1] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detections," Proc. of Conf. on Computer Vision and Pattern Recognition, pp. 886-893, 2005.

[2] Q. Zhu, S. Avidan, M. C. Yeh, K. T. Cheng, "Fast human detection using a cascade of histograms of oriented gradients," Proc. of Conf. on Computer Vision & Pattern Recognition, pp. 1491-1498, 2006.

[3] Y. Nakashima, J. K. Tan, S. Ishikawa, T. Morie, "On detecting a human and its body direction from a video", Proc. of 15th Int. Symposium on Artificial Life and Robotics, pp. 294-297, 2010.

[4] R. E. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions", Machine Learning, Vol. 37, no. 3, pp. 297-336, 1999.

[5] P. Li, T. Zhang, "Visual contour tracking based on particle filters," Proceedings of Generative-Model-Based Vision, Copenhagen, pp. 61-70, 2002.

[6] K. Nummiaro, E. Koller-Meier and L. Van Gool., "A color-based particle filter," Proc. of European Conf. on Computer Vision, pp.53-60, 2002.

[7] F. Porikli, "Integral histogram: A fast way to extract histograms in Cartesian spaces," Proc. of Conf. on Computer Vision and Pattern Recognition, pp. 829-836, 2005.

[8] D. Comaniciu, P. Meer, "Mean shift: A robust approach toward feature space analysis," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, no. 5, pp. 603-619, 2002.

[9] C. R. Wang, J. Wu, and J. J. Lien, "Pedestrian detection system using cascaded boosting with invariance of oriented gradients," International Journal of Pattern Recognition and Artificial Intelligence, vol. 23, no. 4, pp. 801-823, 2009.

[10] X. Wang, T. X. Han, S. Yan, "An HOG-LBP human detector with partial occlusion handling," Proc. of Int. Conf. on Computer Vision, pp. 32-39, 2009.

[11] K. Takahashi, Y. Kuriya, T. Morie, "Bicycle detection using pedaling movement by spatiotemporal Gabor filtering", Int. J. of Innovative Computing Information and Control, Vol. 8, No. 6, pp. 4059-4070, June 2012.

[12] H. Cho, et al., "Vision-based bicyclist detection and tracking for intelligent vehicles", tech. report CMU-RI-TR-10-11, Robotics Institute, Carnegie Mellon University, 2010.

[13] B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes", In IEEE Conf. on Computer Vision and Pattern Recognition, pp. 878-885, 2005.

[14] D. M. Gavrila, V. Philomin, "Real-time object detection for smart vehicles", in International Conference on Computer Vision, 1999.

[15] F. Arnell, Vision-Based Pedestrian Detection System for use in Smart Cars, Master's Thesis, Stockholm, Sweden, 2005.

[16] W. Abd-Almageed, M. Husseinm, M. Abdelkader, and L. Davis, "Real-time human detection and tracking from mobile vehicle", in IEEE Intelligent Transportation Systems Conference, 2007.

[17] D. M. Gavrila, S. Munder, "Multi-cue pedestrian detection and tracking from a moving vehicle", International Journal of Computer Vision, Vol. 73, No. 1, pp. 41-59, 2007.

[18] T. Watanabe, S. Ito, K. Yokoi: "Co-occurrence of histograms of oriented gradients for human detection", IPSJ Transactions on Computer Vision and Applications, pp. 39-47, 2010.

[19] S. Messulodi, et al., "Vision-based bicycle/motorcycle classification", Pattern Recognition Letters, Vol. 28, pp.1719-1726, 2007.

[20] Yun Wu, et al., "Human and bicycle detection system using range sensor", J. Shanghai Jiaotong Univ. (Sci.), Vol. 16, No. 4, pp. 447-451, 2011.

Heewook JUNG

He received the M.S degree and Ph.D degree in mechanical and control engineering from Kyushu Institute of Technology in 2011 and 2014. His research interests include object recognition, pattern recognition and machine learning.

Appendix 1. RealAdaBoost algorithm

1.Suppose that $S=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\cdots,(\boldsymbol{x}_N,y_N)\}$ is a sample space, where $\boldsymbol{x}_i \in X\ (i=1,2,...,N)$ are feature vectors and $y_i \in \{-1,+1\}\ (i=1,2,...,N)$ are labels.

2. D_1 is the initial sample weight defined by

$$D_1(i)=1/N\ (i=1,2,...,N).$$

3. For *t*=1 to *T* do

For *m*=1 to *M* do

Compute probability distribution W_t of weak classifier $h_i(x)$ by

$$W_+^j=\sum_{i,j\in J\wedge y_i=+1}^{n}D_t(i),\ W_-^j=\sum_{i,j\in J\wedge y_i=-1}^{n}D_t(i).$$

Compute estimation Z_m by

$$Z_m=2\sum_j\sqrt{W_+^jW_-^j}\ .$$

Select a weak classifier h_t having the smallest Z_m by

$$h_t=\arg\min Z_{t,m}.$$

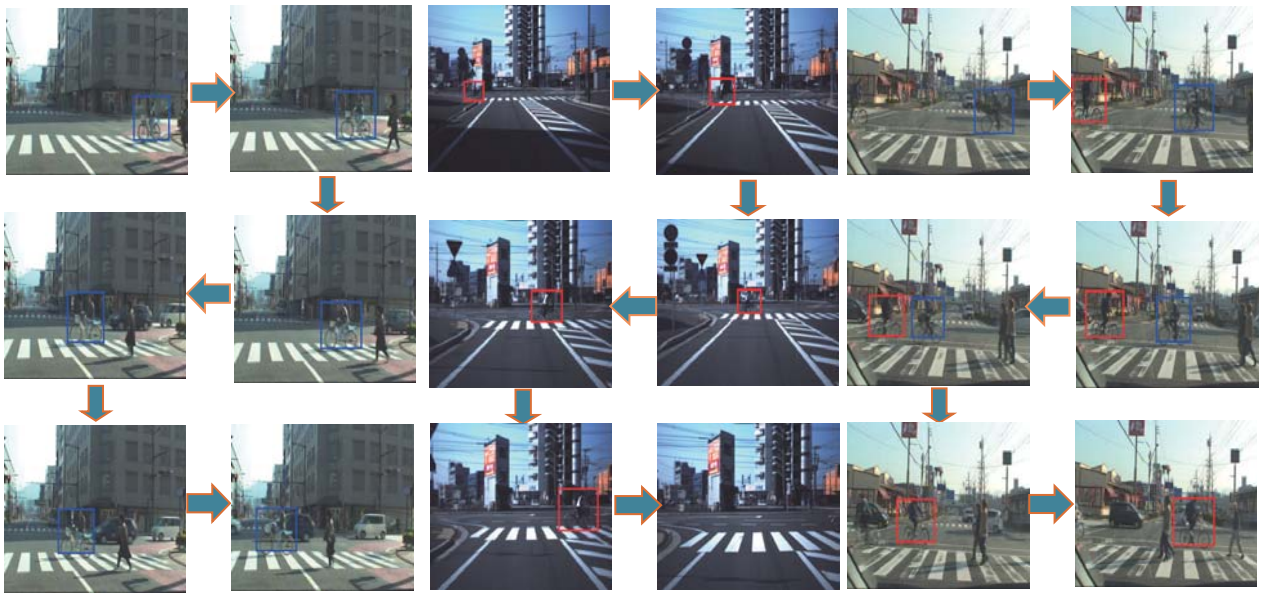
Renew the sample weight $D_t(i)$ by

$$h_t(x_i)=\frac{1}{2}\ln\frac{W_+^j+\varepsilon}{W_-^j+\varepsilon}\ ,\qquad D_{t+1}(i)=D_t(i)\exp[-y_ih_i(x_i)]$$

4. The final strong classifier is given by

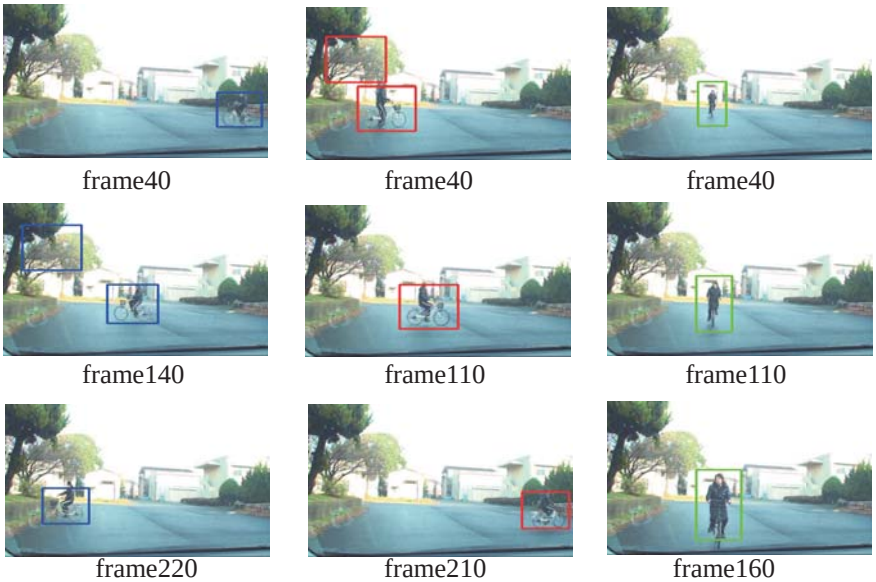
$$H(\boldsymbol{x})=sign\left(\sum_{t=1}^{T}h_t(\boldsymbol{x})\right).$$

Appendix 2.



(a) (b) (c)

Fig. A1. Results of the bicycle detection with PF tracking in real traffic scenes (Exp.2): (a) Case 1, (b) Case 2, and (c) Case 3.



(a) (b) (c)

Fig. A2. Results of the detection of driving direction (Exp.3): (a) Left, (b) right, and (c) front.