PAPER

Extended Feature Descriptor and Vehicle Motion Model with Tracking-by-detection for Pedestrian Active Safety

Hirokatsu KATAOKA[†,††a)], **Kimimasa TAMURA**[†], ***Nonmembers***, **Kenji IWATA**[†††], **Yutaka SATOH**[†††], ***Members***, **Yasuhiro MATSUI**[††††], ***Nonmember***, ***and*** **Yoshimitsu AOKI**[†], ***Member***

SUMMARY The percentage of pedestrian deaths in traffic accidents is on the rise in Japan. In recent years, there have been calls for measures to be introduced to protect vulnerable road users such as pedestrians and cyclists. In this study, a method to detect and track pedestrians using an in-vehicle camera is presented. We improve the technology of detecting pedestrians by using the highly accurate images obtained with a monocular camera. In the detection step, we employ ECoHOG as the feature descriptor; it accumulates the integrated gradient intensities. In the tracking step, we apply an effective motion model using optical flow and the proposed feature descriptor ECoHOG in a tracking-by-detection framework. These techniques were verified using images captured on real roads.

key words: *Pedestrian Active Safety, Tracking-by-detection, ECoHOG, Particle Filter, Vehicle Motion Model*

1. Introduction

In Japan, the number of pedestrian deaths fell to 4411 in 2012 which is likely to decrease further over the next several years. However, the percentage of pedestrian deaths among all deaths in traffic accidents is increasing [1]. In an effort to reduce pedestrian deaths, investigations are being conducted in the area of pedestrian intelligent transport systems. Along these lines, we are currently studying a pedestrian active safety (collision avoidance) system, which is able to detect and track pedestrians by using in-vehicle sensors and automatic braking. There are great expectations for the system. The use of in-vehicle cameras is efficient in detecting obstacles, and many studies have been devoted to pedestrian detection based on cameras.

Pedestrians must be identified in outdoor scenes and from cluttered backgrounds with various illuminations. In addition, pedestrians can be occluded by traffic elements, such as signs and vehicles. There are several surveys of pedestrian detection in traffic scenes. For example, Gandhi *et al.* [2], Geronimo *et al.* [3], and Dollar *et al.* [4] have enumerated approaches that can accurately and rapidly detect pedestrian(s) on real roads.

Geronimo *et al*. used Real AdaBoost to select effective features from Haar-like and Edge Orientation Histograms [5]. Dalal *et al*. presented a human classification feature called the Histograms of Oriented Gradients (HOG) and a linear SVM as a learning classifier. The HOG feature describes the shape of an object from images that are divided into blocks and cells. The HOG can roughly represent a human edge feature employing statistical learning which has been widely used in the field of computer vision. The HOG method has been improved by combining the HOG [7] with the Co-occurrence Probability Feature (CPF) [8]. These HOG feature and CPF provide the co-occurrence feature using the returning value of classifiers. Haar-like [9], Haar-wavelet [10] and Edge Orientation Histograms [11] are also used to detect pedestrians on real-world roads. As shown in the paper [4], Walk *et al.* [12] proposed a state-of-the-art approach in pedestrian detection for active safety. In [12], they proposed improvement of edge-based feature (e.g. Haar-like[9], shapelets[13], and shape context[14]) by combining color self-similarity and the motion features from optic flow. However, this approach cannot be realized in real-time whose properties are obviously required for traffic safety systems. At this point, Co-occurrence of Histograms of Oriented Gradients (CoHOG) is known as a high-standard detection approach for pedestrian detection, by representing edge-pair [15]. They described that the CoHOG descriptor is better than HOG [6], shapelets [13], and M-HOG [16]. Moreover, CoHOG achieved real-time detection on road-scene dataset due to individual processing at each block, which is an effective approach in multi-core systems such as in-vehicle hardware.

On the other hand, because of the high costs of stereo technology, a monocular camera technique is preferable in active safety. The use of monocular cameras as driving recorders has become more popular such as in taxis and buses. Accordingly, we devoted our efforts to developing a pedestrian active safety technique using a monocular camera.

In this paper, we present highly accurate pedestrian detection and tracking techniques using a monocular camera in a tracking-by-detection framework. Our contribution includes:

- The Extended CoHOG (ECoHOG) is proposed for pedestrian detection. We additionally process edge-magnitude accumulation, histogram normalization, and dimensional compression from CoHOG [15].
- Tracking-by-detection framework is realized as a particle filter. Vehicle motion approximation and ECoHOG

Manuscript received January 1, 2011.
Manuscript revised January 1, 2011.
[†]Keio University
[††]Japan Society for the Promotion of Science
[†††]National Institute of Advanced Industrial Science and Technology (AIST)
[††††]National Traffic Safety and Environment Laboratory
a) E-mail: kataoka@aoki-medialab.org
DOI: 10.1587/trans.E0.??.1

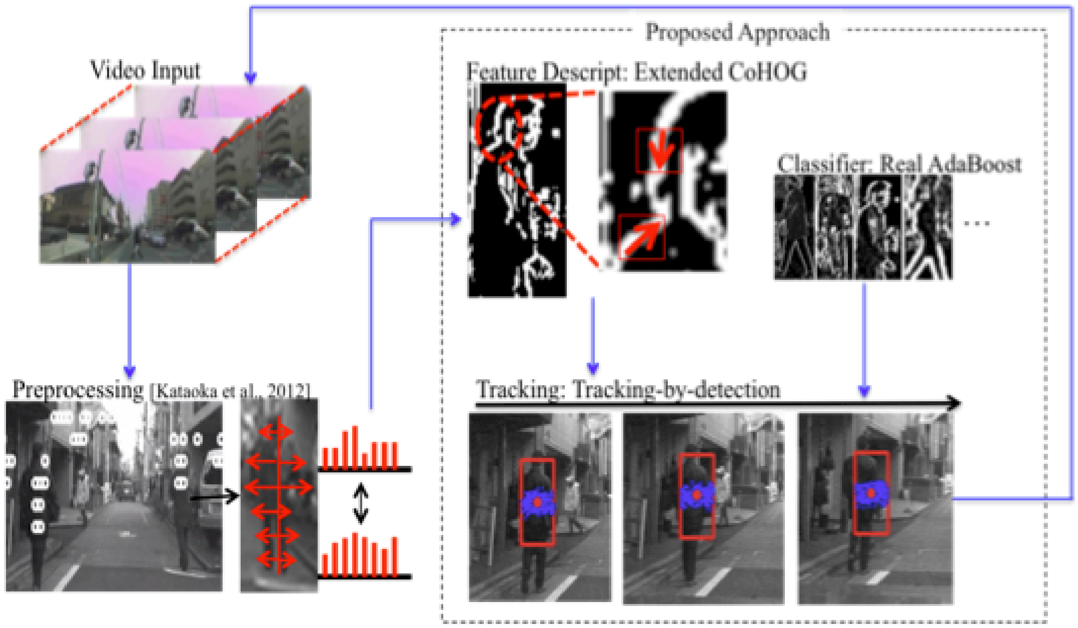


Fig. 1 The process flow of the pedestrian active safety system

are implemented for motion model and likelihood calculation, respectively.

The detection and tracking experiments were carried out on INRIA person dataset, Daimler pedestrian benchmark dataset and self collected dataset for evaluation.

2. Proposed Method

Figure 1 shows the sequence of events for our proposed method. The area in front of the vehicle is taken by the in vehicle camera. The current method for capturing a pedestrian's shape employs co-occurrence histograms of oriented gradients (CoHOG) [15]. In this paper, however, we propose the use of extended CoHOG (ECoHOG). We implemented tracking-by-detection with a particle filter [17] for pedestrian tracking. Optical flow [18] is applied to estimate vehicle motion model. The ECoHOG is applied to calculate likelihood from particles. To test the applicability of ECoHOG, we compared it with CoHOG using standardized datasets. In preprocessing, we applied symmetrical judgment proposed by Kataoka *et al.* [19] which is a fast processing method using the degree of symmetry between left and right images of a human supporting a human body shape is symmetrical. The thresholding value is set using a large number of positive and negative learning images.

2.1 Extended Co-occurrence Histograms of Oriented Gradients (ECoHOG)

We used ECoHOG as an improvement of CoHOG [15] for pedestrian detection. For pedestrian-shape extraction, CoHOG investigates the pedestrian's edge direction minutely

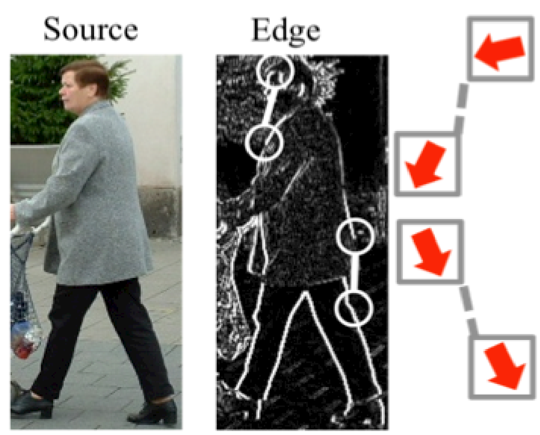


Fig. 2 How to acquire an edge pair from a pedestrian shape

and uses two different pixels in an extraction window as a direction pair. From the edge direction pair, CoHOG can describe such features as the head-shoulder pair and the back-leg pair (Figure 2), which reduces miss detection. Unlike CoHOG, ECoHOG acquires not only the edge-direction pair, but also an edge-magnitude pair. ECoHOG can represent the edge-magnitude characteristics of the pedestrian shape.

Edge magnitude accumulation and histogram normalization: this approach is a possible edge-magnitude accumulation method that is able to be used to generate the pedestrian shape model in minute details. The edge-magnitude-accumulation method is able to describe details more comprehensively than the CoHOG feature descriptor

(Figure 3). In other words, ECoHOG considers not only the pedestrian shape features but also the degree of edge magnitude representing "shape strength". We believe the addition of the edge-magnitude pair is an effective way to classify the pedestrian and background with high accuracy. Equations relating to ECoHOG are as follows.

$$m_1(x_1, y_1) = \sqrt{f_{x1}(x_1, y_1)^2 + f_{y1}(x_1, y_1)^2} \tag{1}$$

$$m_2(x_2, y_2) = \sqrt{f_{x2}(x_2, y_2)^2 + f_{y2}(x_2, y_2)^2} \tag{2}$$

$$f_x(x, y) = I(x+1, y) - I(x-1, y) \tag{3}$$

$$f_y(x, y) = I(x, y+1) - I(x, y-1) \tag{4}$$

$$C_{x,y}(i, j) = \sum_{p=1}^{n} \sum_{q=1}^{m} \begin{cases} m_1(x_1, y_1) + m_2(x_2, y_2) \\ (if\ d(p, q) = i\ and \\ \ d(p + x, q + y) = j) \\ \\ 0 \\ (otherwise) \end{cases} \tag{5}$$

where $m_1(x, y)$ and $m_2(x, y)$ are the magnitude of the pixel of interest (at the center of the window) and the magnitude in the objective window, respectively. $f_x(x, y)$ and $f_y(x, y)$ are the differences between two pixels in the x and y directions. $C(i, j)$ is the co-occurrence histogram that accumulates pairs of the pixel of interest and an objective pixel. Each $C(i, j)$ has 64 dimensions because the direction is quantized into eight possibilities and ECoHOG combines two directions. Coordinates (p, q) indicate the pixel of interest (center of window) and coordinates $(p+x, p+y)$ indicate the objective pixel. m and n are the width and height of the feature extraction window. $d(p, q)$ is a function that quantizes the edge direction as an integer from 0 to 7 at pixel (p, q).

ECoHOG normalizes the co-occurrence histogram to off-set the difference in edge-magnitude due to image brightness. The co-occurrence histogram is divided by its norm:

$$C'_{x,y}(i, j) = \frac{C_{x,y}(i, j)}{\sum_{i'=0}^{7} \sum_{j'=0}^{7} C_{x,y}(i', j')} \tag{6}$$

Dimensional Compression with Principal Component Analysis (PCA-ECoHOG): Principal Component Analysis (PCA) is used for dimensional compression. We believe that a low-dimensional feature is easy to be divided into positive and negative classes. In related work, CoHOG needed about 35,000 dimensions for pedestrian detection [15]; however, we propose a low-dimension feature with PCA. Employing this method, we compressed the feature dimension from 4600 to a few hundreds. The PCA comparison experiment shows the effective number of dimensions for pedestrian detection. In the field of pattern recognition, high-dimensional feature spaces are referred to as "the curse of dimensionality" [20]. The problem is feature dimension increases with respect to the amount of information taken from the object. We believe that data reduction is the best way to solve this problem and improve the accuracy of recognition. We show the relationship between feature size compressed by PCA and detection rate in feature descriptor experiment.

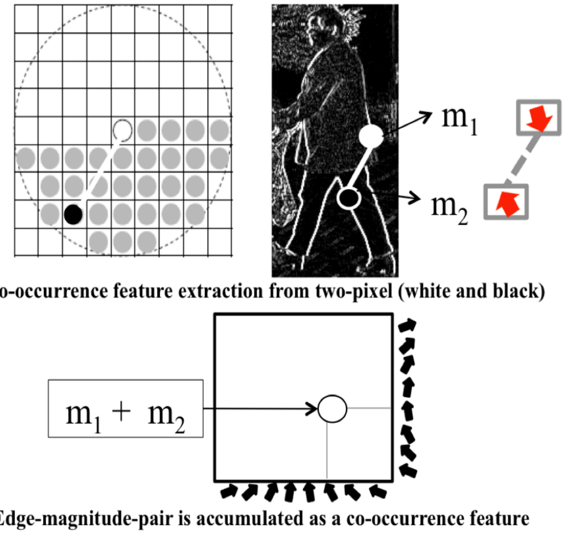


Fig. 3 Edge magnitude accumulation from two different pixels : The ECoHOG incorporates co-occurrence edge magnitude into histogram. The co-occurrence edges are extracted from white and gray pixels in offset.

2.2 Tracking-by-detection in Particle Filter Algorithm

Recently, the tracking-by-detection framework has been found to be effective in the field of computer vision. In addition to the significant detection method, we need to apply statistic-based tracking such as sequential Monte Carlo algorithm (or particle filter). Within this context, particle filter estimates the sampling points in a frame. In this section, we describe the latest works on likelihood calculations using ECoHOG and a vehicle motion model. We believe that edge-based general model is effective for pedestrian tracking in complicated situations (e.g. appearance variation, light source, and camera angle changing).

The searching and optimization are performed using a particle filter [17]. A particle filter is an iterative algorithm including prediction, update and resampling, and is a type of Bayesian estimator that applies the Monte Carlo framework. The current state is estimated as the expectation of the likelihood distribution. The likelihood calculation evaluates the condition of an object by comparing texture between the model image and an image around a particle.

Step1 Initialization

We execute initialization after specifying a pedestrian's position using a detector. The system puts the tracker on a detected point. We arrange particles around the pedestrian as likelihood . A particle has a motion model and observation model. The motion model is based on vehicle motion obtained from the front video, and the observation model is calculated by the ECoHOG detector. These are shown in the step2, state prediction and step3, likelihood calculation.

Step2 State prediction

We apply the vehicle movement model as a motion model. Vehicle movement is calculated using Lucas-Kanade optical flow [18] that tracks a feature point between spatio-temporal frames as seen in Figure 4. The average of the optical flow direction and magnitude in a frame is the movement model. Furthermore, we add Gaussian noise to a particle's velocity and location. We estimate from the linear uniform motion if there is no vehicle movement:

$$X_{t+1} = FX_t + w_t \tag{7}$$

$$X_t = (x, y, v_x, v_y)^{\mathrm{T}} \tag{8}$$

$$F = \begin{pmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{9}$$

where x and y are the location of the particle, and v_x and v_y are the velocity between frames. w_t is the system noise added to the particle location and velocity.

Step3 Likelihood calculation

The pedestrian model is important for pedestrian tracking. We believe that a pedestrian model based on machine learning can be a universal pedestrian model in the likelihood calculation.

We perform classification by learning with Real AdaBoost classifier and the proposed feature descriptor ECoHOG in the tracking step. Real AdaBoost outputs the evaluated value as a real number from the learning sample. In this situation, the learning sample is distinguished either a positive (pedestrian) or negative (background) image. We prepare a large number of learning images for pedestrian recognition. Figure 5 shows the comparison between general model that is learned and extracted feature from observation area.

The followings are the value of the weak classifier and the value of the strong classifier accumulating weak classifiers in the Real AdaBoost algorithm:

$$h(x) = \frac{1}{2} \ln \frac{W_{positive} + \epsilon}{W_{negative} + \epsilon} \tag{10}$$

$$H(x) = \sum_{t=1}^{T} h_t(x) \tag{11}$$

where h(x) and H(x) are respectively the returning values of the weak and strong classifier. $W_{positive}$ and $W_{negative}$ are the weight values of Real AdaBoost. ϵ is set as $\frac{1}{N}$ using the number of learning samples N.

A rectangle area around a particle is input to calculate the likelihood. We can obtain the likelihood with a classifier in the rectangle area which is then used to capture a pedestrian.

Step 4 Likelihood evaluation

The center of gravity is calculated in this step:

$$(g_x, g_y) = (\sum_{i=1}^{n} L_i x_i, \sum_{i=1}^{n} L_i y_i) \tag{12}$$

where g is the center of gravity of the pedestrian. L is each particle's likelihood and x and y are the position of the particle. The likelihood is normalized as $\Sigma_{i=1}^{n} L_i = 1$. The center of gravity (g_x, g_y) is calculated with average with weight from particle position and normalized likelihood.

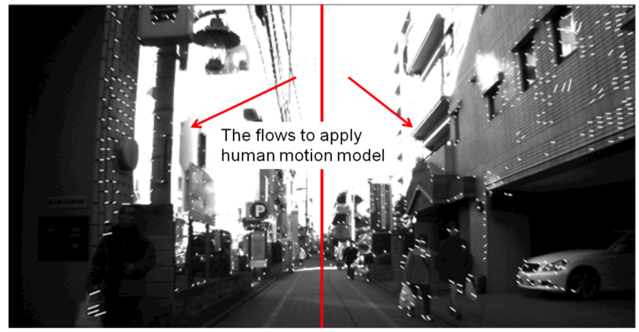


Fig. 4 Flow vector calculation from driver's view

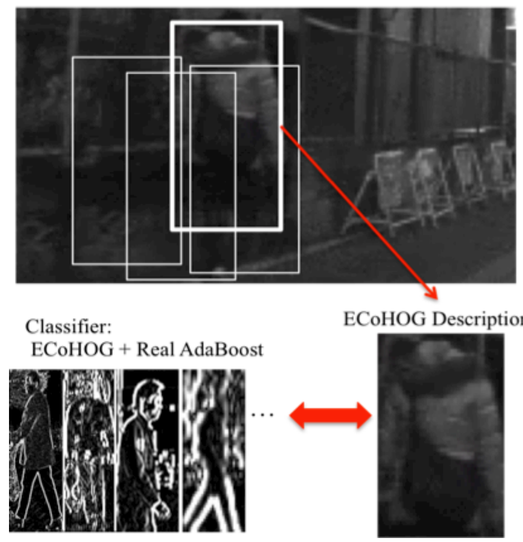


Fig. 5 ECoHOG feature description from observation area and matching with classifier consist of ECoHOG and Real AdaBoost

3. Experiment

We carried out experiments to validate the detection and tracking approaches. Experiment 1 was conducted to compare our proposed feature descriptor (ECoHOG) with CoHOG and choose the feature descriptor in real-road scenes, and experiment 2 was to verify the tracking methods. We used the INRIA person dataset [21] and Daimler pedestrian benchmark dataset [22] as benchmarks for experiment 1, and self-acquired real-road datasets for experiment 2.

First, we describe how to implement the feature descriptor and tracking-by-detection.

3.1 Implementation

We describe the implementations of ECoHOG and tracking-by-detection. The parameters and detailed implementations are given below.

ECoHOG: In this method, we set the parameters same as those used for the CoHOG feature descriptor. The window size, image block division, number of offsets, histogram division and dimension are given in Table 1.

Tracking-by-detection: The tracking-by-detection and its machine learning are described here. In the particle filter algorithm, the number of particles, motion model, and likelihood calculation should be input to capture the objective movement. In addition, the extraction size is input for calculating ECoHOG. To construct Real AdaBoost, we need to set the number of weak classifiers and learning samples. The settings of tracking-by-detection and machine learning are tabulated in table 2.

Real AdaBoost: Both of detection and tracking step, we calculate the likelihood by using Real AdaBoost classifier. The number of image is preferable to input 10×10^4 – 10×10^5 order. We input 3,000 positive and 20,000 negative images as self-collected dataset for pedestrian tracking. We does not implement on-line learning in this paper, therefore, Real AdaBoost classifiers are constructed in advance.

The tracking algorithm is implemented on a standard laptop (Windows 7, Intel Core i7-2620M, 2.70 GHz CPU, 4.0 GB RAM). We applied C++ and OpenCV2.4 programming for pedestrian tracking.

Table 1 Parameters of ECoHOG for pedestrian detection

Window size	7×7 pixels
Block division	4 blocks (2×2)
Number of offsets	18 offsets
Histogram division	8 directions
Number of dimensions	4608 dimensions
Number of dimensions(after PCA)	100 dimensions

Table 2 Settings of tracking-by-detection and machine learning for pedestrian tracking

Number of particles	150
Extraction size	80×160 pixels
Learning samples	3,000 positive images 20,000 negative images
Motion model	Optical flow
Likelihood	ECoHOG classifier
Weak classifier	50 classifiers

3.2 Experiment 1: Feature Descriptor Verification using the INRIA Person Dataset and Daimler pedestrian benchmark dataset

We used the INRIA person dataset [21], which includes pedestrian images and a Detection Error Tradeoff (DET) curve for the verification. The number of pixels for feature extraction is 18 pixels. ECoHOG has 4608 dimensions, and PCA-ECoHOG has 100 dimensions.

First, we compare the proposed and previous frameworks in Figure 6. Figure 7 shows the comparison between PCA-ECoHOG and ECoHOG. Figure 8 shows the relationship between the number of dimensions and the detection rate. The vertical axis of the DET curve is miss rate, and the horizontal axis is the false positive rate; therefore, the bottom-left of the DET curve proves better performance of ECoHOG.

Table 3 Processing time of feature descriptors (ECoHOG, PCA-ECoHOG)

Feature descriptor	Processing time
ECoHOG	11.2 ms
PCA-ECoHOG	12.9 ms

Figure 6 shows that the proposed framework provides the best performance; ECoHOG accumulates the edge magnitude and normalizes an image. The experiment shows that the effectiveness of co-occurrence extraction (from HOG [6] to CPF [8], CoHOG [15], and ECoHOG), pixel-pair representation (CPF and CoHOG, ECoHOG), and edge-magnitude accumulation (the difference between CoHOG and ECoHOG). The ECoHOG represents more effective human-edge model on INRIA person dataset by means of co-occurrence extraction, edge-magnitude accumulation and histogram normalization. Edge-magnitude representation captures degree of edge strength, therefore, the feature allow us to be robust in blurred and occluded scenes. ECoHOG indicated better results than CoHOG on INRIA person dataset.

In the comparison of ECoHOG and PCA-ECoHOG (Figure 7), PCA-ECoHOG is proven to be the best method for pedestrian detection. It is easy to divide the feature space into two classes because PCA-ECoHOG compresses the number of dimensions from 4608 to 100. Moreover, we do not need a large number of learning samples in small feature space (e.g. million order of learning samples). Table 3 gives the processing times of ECoHOG and PCA-ECoHOG for each frame. The matrix calculation step is only needed for dimensional compression. The ECoHOG is comparable to the CoHOG with respect to processing time. Both methods realized real-time processing on the INRIA person dataset.

Figure 8 shows the relationship between the dimension number and detection rate. The number of dimensions is compressed by a factor of 5-200 (i.e., 5, 7, 15, 20, 50, 100,

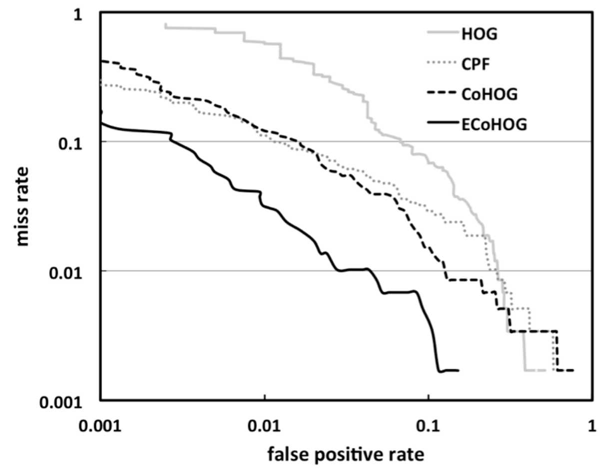


Fig. 6 Comparison of four feature descriptors applying the Detection Error Tradeoff (DET) curve : ECoHOG, CoHOG[15], CPF[8], HOG[6]

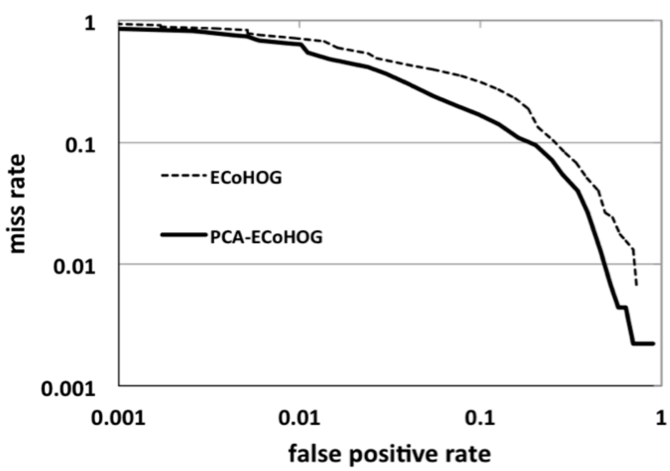


Fig. 7 Detection rates of two proposed methods: ECoHOG and PCA-ECoHOG

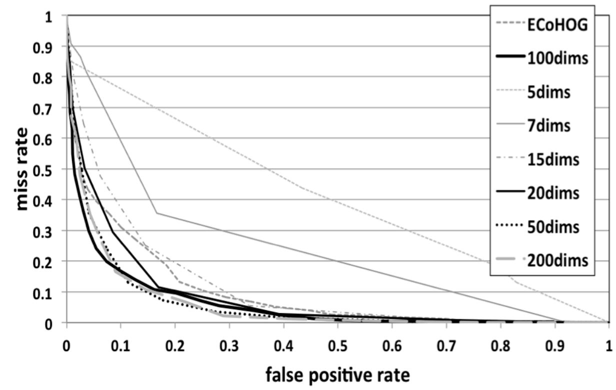


Fig. 8 Detection rates of PCA-ECoHOG: The dimensions are set as 5, 7, 15, 20, 50, 100, 200, respectively

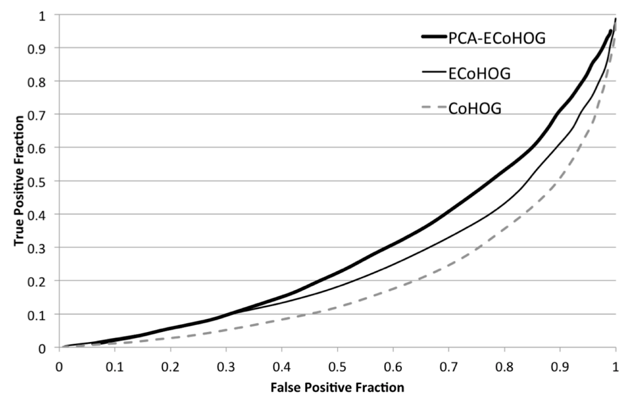


Fig. 9 The ROC curve on the Daimler pedestrian benchmark dataset

200 dimensions) in the figure. This experiment shows that the best number of dimensions is 100. Five dimensions provide little information, whereas 200 dimensions provide a large feature space. From this result, we can describe two types of characteristics: (i) the prevalence of model edge information and (ii) the feature space size. Higher rate of (i) the prevalence of model edge information and lower value of (ii) the feature space size are preferable for learning-based pedestrian detection. We achieved the best result when the number of dimensions is 100. In this paper, the dimensions of PCA-ECoHOG is reduced 100 from 4608-dimension.

Additionally, we carried out the experiment on Daimler pedestrian benchmark dataset [22] to validate in a traffic scene. In this experiment, we applied three approaches which achieved the best three accuracy on the INRIA person dataset. The Receiver Operating Characteristic (ROC) curve are given in Figure 9. The vertical and horizontal axis of ROC curve indicate true positive rate and false positive rate, respectively. The top-left of the graph shows better performance, therefore, PCA-ECoHOG and ECoHOG prove better accuracy rate than CoHOG. We show the detailed analysis in Figure 10, for example, only CoHOG miss-detected posture changing, occluded by pedestrian, cluttered background, and occluded/shadowed by umbrella. The Daimler pedestrian benchmark dataset is known as small image (18×36 pixels) detection in traffic scenes. We believe that ECoHOG can capture edge magnitude features even if a pedestrian is in complicated situations. Edge magnitude accumulation allows us to derive more robust features from difficult scenes by using blurred edge information.

3.3 Experiment 2: Effectiveness of Tracking in Real Scenes

We carried out a tracking experiment using our real-road dataset. The objective is to track a single pedestrian without occlusions or the crossing of other pedestrians. The examples of tracking dataset are shown in Figure 11. We collected totally 2 hours of 30 Hz video taken from a driving vehicle during daytime. We have edited pedestrian-centered

Fig. 10 The images which are miss-detected only in CoHOG detection; (from left to right) posture changing, occluded by pedestrian, cluttered background, and with object

Table 4 Results of tracking experiment: (i) grayscale histogram matching + linear uniform motion, (ii) texture matching + linear uniform motion, and (iii) texture matching + vehicle motion model are compared as related approaches. (iv) ECoHOG classifier + vehicle motion model (proposed approach)

Method	Accuracy	Frame per seconds
(i)	18.7%	62.42
(ii)	72.3%	52.00
(iii)	87.2%	26.99
(iv)	99.2%	5.85

image sequences. The pedestrian tracking dataset approximately contains 10,000 images which pedestrians are appeared. The dataset is captured in shadow areas, cluttered backgrounds, and an urban scene. Moreover, our vehicle velocity is set between 30 - 60 kilometers per hour. The dataset includes difficulties of computer vision such as occlusion, complicated background, posture variation, and illumination changing.

We applied three related works: (i) grayscale histogram matching + linear uniform motion, (ii) texture matching + linear uniform motion and (iii) texture matching + vehicle motion model are compared as related approaches. In addition, the proposed approach (iv) ECoHOG classifier + vehicle motion model was applied. The initial position was set manually and the histogram or texture was extracted from the initial rectangle in related works. Our proposed framework includes a pedestrian model obtained by learning positive and negative samples; therefore, we do not need to capture the pedestrian model in the initial frame. Table 4 shows the results of the tracking experiment and Figure 12 shows the tracking for the real-road dataset.

The grayscale histogram does not evaluate human likelihood effectively since it ignores the location of the human's texture of pose and close. The histogram feature is confusing unless it is color depth. Furthermore, the grayscale value changes dramatically in real time. We applied texture matching including location information from the pedestrian. In this case, the time-series texture of the pedestrian has few changes. Texture matching is a better way to capture a pedestrian from in-vehicle video. However, texture accumulates a gap due to a changing angle and scale if tracking continues for a long period. We need to fix the template using a point-of-interest tracker; for example, the SIFT (Scale Invariant Feature Transform) [23] or SURF (Speeded Up Robust Features) [24].

Fig. 11 The examples of self-collected tracking dataset in urban areas

Our proposed approach had the best accuracy of 99.2% among the examined methods. The approach includes a general pedestrian model in the classifier obtained from positive and negative samples. Learning-based general pedestrian model handles variations of pedestrian's posture, camera angle and light source. We believe that the model is more effective than other models such as texture or color histogram. Figure 12 shows the pedestrian tracking with texture matching (left) and ECoHOG + Real AdaBoost classifier (right) in real-road. The two methods apply vehicle motion model as a state prediction. In texture matching, tracking is not stable because of appearance and background changing depending on the vehicle movement. The tracking is lost in the last frame of Figure 12. On the other hand, the proposed method successfully tracks the pedestrian even in complicated situations. The experiment shows that a general pedestrian mdoel is effective for pedestrian tracking in traffic scenes.

According to calculation of the processing time, our proposed approach can be realized near to real time. Although the processing time of 5.85 fps is fast, we require program optimization with in-vehicle hardware to process in real time. In addition, we should additionally implement a fast and accurate recognition method on the software side.

Figure 13 shows the likelihood transition under pedestrian tracking. The black line and gray line show the results of tracking-by-detection and texture matching. Both methods apply a vehicle motion model. Tracking-by-detection

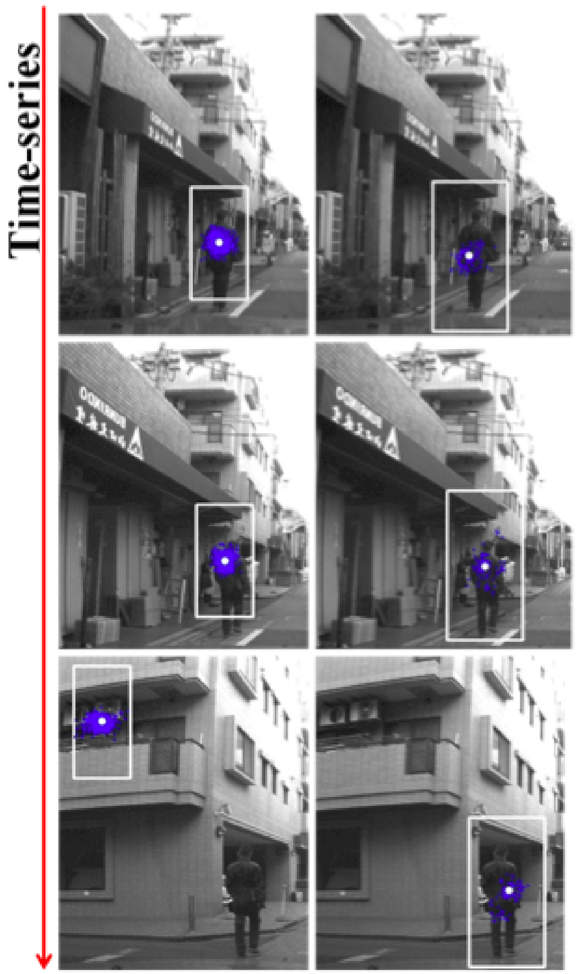


Fig. 12 Pedestrian tracking-by-detection: (left) texture matching and vehicle motion model (right) ECoHOG + Real AdaBoost classifier and vehicle motion model.

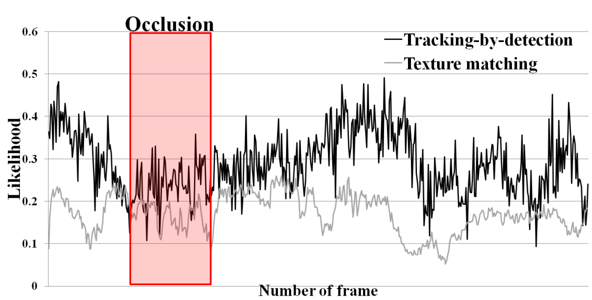


Fig. 13 Likelihood transition under pedestrian tracking.

outputs a higher value in most cases. The pedestrian model obtained by machine learning allows us to keep a likelihood in tracking. The many variations of the image are included in the machine learning step; e.g. occluded situations, complicated backgrounds and dark or bright scenes. Machine learning allows tracking in human-human occluded scenes.

4. Conclusion

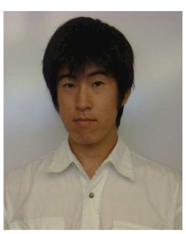
This paper proposed a pedestrian tracking method for active pedestrian safety The contributions of this approach are (i) a state-of-the-art detector that combines ECoHOG with Real AdaBoost and (ii) tracking-by-detection in the particle filter framework using the ECoHOG classifier and a vehicle motion model.

In future work, a tracking method robust against occlusion and multiple people should be considered. Our current pedestrian model includes only a person model obtained from machine learning thus occlusion and a multi-person model in pedestrian tracking should be developed. Furthermore, we will attempt ”pedestrian motion prediction” based on an estimation algorithm. The prediction framework is important for the development of advanced active safety systems.

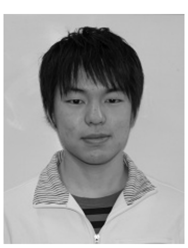
References

[1] ”Institute for Traffic Accident Research and Data Analysis of Japan (ITARDA)”, Annual Traffic Accident Report in 2010, Tokyo, 2011.
[2] Tarak Gandhi, Mohan M. Trivedi, ”Pedestrian Collision Avoidance Systems:A Survey of Computer Vision Based Recent Studies”, IEEE Intelligent Transportation Systems Conference (ITSC), pp.976-981, 2006.
[3] David Geronimo, Antonio M. Lopez, Angel D. Sappa, Thorsten Graf, ”Survey of Pedestrian Detection for Advanced Driver Assistance Systems”, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.32, No.7, 2010.
[4] P. Dollar, C. Wojek, B. Schiele, P. Perona, ”Pedestrian Detection: An Evaluation of the State of the Art”, IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI), vol.34, no.4, pp.743-761, 2012.
[5] D. Geronimo, A. Sappa, A. Lopez, and D. Ponsa, ”Adaptive Image Sampling and Windows Classification for On-Board Pedestrian Detection”, Int’ Conf. Computer Vision Systems, 2007.
[6] N. Dalal, B. Triggs, ”Histograms of oriented gradients for human detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp886-893, 2005.
[7] T. Mitsui, Y. Yamauchi, and H. Fujiyoshi, ”Object Detection by Two-Stage Boosting with Joint Features”, The Institute of Electronics, Information and Communication Engineers Transactions on Information and Systems,Vol. J92-D, No. 9, pp. 1591-1601, 2009.
[8] Y. Yamauchi, H. Fujiyoshi, Y. Iwahori, and T. Kanade, ”People Detection based on Co-occurrence of Appearance and Spatio-Temporal Features”, National Institute of Informatics Transactions on Progress in Informatics, No. 7, pp. 33-42, 2010.
[9] P. Viola, M. J. Jones, ”Rapid Object Detection using a Boosted Cascade of Simple Features”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2001.
[10] M. Oren, C. Papageorgiou, P. Sinha, E. Osuma, T. Poggio, ”Pedestrian Detection Using Wavelet Templates”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1997.
[11] K. Levi, Y. Weiss, ”Learning object detection from a small number of examples: the importance of good features”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2004.
[12] S. Walk, N. Majer, K. Schindler, B. Schiele, ”New Features and Insights for Pedestrian Detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010.
[13] P. Sabzmeydani, G. Mori, ”Detecting Pedestrians by Learning Shapelet Features”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2007.
[14] G. Mori, S. Belongie, J. Malik, ”Efficient Shape Matching Using

Shape Contexts", IEEE Trans. Pattern Analysis and Machine Intelligence (TPAMI), vol.27, no.11, pp.1832-1837, 2005.
[15] T. Watanabe, S. Ito, K. Yokoi, "Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection", IPSJ Trans. Computer Vision and Applications (CVA), vol.5, no.2, pp.659-667, 2010.
[16] S. Maji, A. C. Berg, J. Malik, "Classification using Intersection Kernel Support Vector Machines is Efficient", IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2008.
[17] Isard M, Blake A, "Condensation : conditional density propagation for visual tracking", International Journal of Computer Vision, vol.28, no.1, pp.5-28, 1998.
[18] B.D.Lucas, Takeo Kanade, "An iterative image registration technique with an application to stereo vision", Proceedings of Imaging Understanding Workshop, pp.121-130, 1981.
[19] Hirokatsu Kataoka, Yoshimitsu Aoki, Yasuhiro Matsui, "Symmetrical Judgment Area Reduction and ECoHOG Feature Descriptor for Pedestrian Detection", International Journal of Vehicle Safety, 2012.
[20] Charu C. Aggarwal, "On k-anonymity and the curse of dimensionality", VLDB, pp.901-909, 2005.
[21] INRIA Person Dataset, http://pascal.inrialpes.fr/ data/human
[22] S. Munder, D. M. Gavrila, "An Experimental Study on Pedestrian Classification", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.28, No.11, pp.1863-1868, 2006.
[23] David G. Lowe, "Distinctive image features from scale-invariant keypoints," International Journal of Computer Vision, 60, 2 (2004), pp. 91-110
[24] Herbert Bay, Andreas Ess, Tinne Tuytelaars, Luc Van Gool "SURF: Speeded Up Robust Features", Computer Vision and Image Understanding (CVIU), Vol. 110, No. 3, pp. 346–359, 2008



Hirokatsu Kataoka was born in 1986. He received his M.E. from Keio University in 2011, visiting researcher at University of California in 2011-2012, and research internship at AIST in 2012-2013. Currently, he is a Ph.D. student in Keio Univesity and a research fellow at JSPS. His research interests include computer vision, pattern recognition, and data mining. He is mainly engaged in human activity recognition indoor situation, pedestrian active safety and sports video analysis.



Kimimasa Tamura received his B.E. from Keio University in 2012. Currently, he is a M.E. student in Keio University. His research interests include computer vision, HMI, drivers' safety. He is engaged in human gaze estimation.

Kenji Iwata was born in 1975. He received a Ph.D. in engineering from Gifu University in 2002. He is a research scientist at the Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST). His research interests include computer vision, pattern recognition, and middle-ware for image processing.

Yutaka Satoh was born in 1971. He received a Ph.D. in engineering from Hokkaido University in 2001. He is a senior research scientist at the Intelligent Systems Research Institute, National Institute of Advanced Industrial Science and Technology (AIST). He is also an associate professor in the Graduate School of Systems and Information Engineering, University of Tsukuba. His research interests include computer vision, robust image matching, and stereo omnidirectional cameras.

Yasuhiro Matsui is a senior chief researcher at National Traffic Safety and Environment Laboratory, Tokyo, Japan. His research interests are pedestrian protection and occupant protection for active and passive safety technologies. He holds mechanical engineering degrees from Nihon University (BS, MS), and Nagoya University (PhD), Japan.

Yoshimitsu Aoki is an Associate Professor, Department of Electronics & Electrical Engineering, Keio University. He received his PhD in Engineering from Waseda University in 2001. From 2002 to 2008, he was an associate professor in Shibaura Institute of Technology. Since 2008, he is an associate professor at Department of Electronics & Electrical Engineering in Keio University. He performs researches in the areas of Computer Vision, Pattern Recognition, and Media Sensing/Understanding.