Special Feature: Spatial Information Technology towards Intelligent Vehicle Systems

Research Report

SpaFIND: An Effective and Low-cost Feature Descriptor for Pedestrian Protection Systems in Economy Cars

Takeo Kato, Chunzhao Guo, Kiyosumi Kidono, Yoshiko Kojima and Takashi Naito

Report received on Feb. 7, 2018

ABSTRACT Pedestrian protection systems (PPSs) are important for reducing traffic fatalities. However, the majority of economy cars cannot benefit from such systems due to the relatively high cost of such products. In the present paper, an effective and low-cost sparse feature interaction descriptor (SpaFIND) is proposed for PPSs in economy cars, which have limited computational power. SpaFIND extends the histogram of oriented gradients (HOG) feature and selectively computes the correlations among adjacent components of the HOG. Consequently, SpaFIND can capture the second-order properties of the object appearance with a low computational load, which is the main contribution of the present study. The detection accuracy and computational load of the proposed method are evaluated experimentally. The experimental results demonstrated the effectiveness of the proposed low-cost SpaFIND feature, which satisfies the requirements for implementing a PPS with limited computational power. Therefore, SpaFIND will contribute to the massive deployment of PPSs in economy cars in the near future.

KEYWORDS Advanced Driver Assistance System (ADAS), Feature Extraction, Computer Vision, Object Detection, Pedestrian Protection

1. Introduction

According to traffic accident statistics, globally, more than 270000 pedestrians are killed on roads every year, accounting for 22% of all road traffic fatalities.[1] In Japan, pedestrian deaths account for 36% of all traffic fatalities, which is the largest proportion, which continues to increase, of road fatalities.[2]

Advanced driver assistance systems (ADASs) and autonomous vehicles have been developed in order to improve safety, traffic efficiency, and driver comfort over the last several decades. A pedestrian protection system (PPS) is one of the most important systems in an ADAS, because collisions with pedestrians often result in fatalities. The PPS detects surrounding pedestrians and alerts the driver or performs braking automatically if necessary. According to market data, most PPSs are equipped in premium and standard cars, and the majority of economy cars do not benefit from such systems due to cost. In Japan, the share of economy cars (or light cars) has steadily increased from 17.1% in 1979 to 37.9% in 2014.[3] As such, considering the number of economy cars, the demand for a PPS for use in economy cars should be great.

In order to meet the demand of the PPS for economy cars, two strict and conflicting practical requirements must be met. The system must achieve a considerably high detection accuracy and run in real time, while remaining inexpensive. Therefore, rather than high-definition (HD) LiDAR, HD cameras and high-performance processors, such as GPUs and multicore CPUs, a PPS for an economy car would require the combination of a regular-resolution camera and a low-power-consumption processor with limited computational power. However, since there is a trade-off relationship between detection accuracy and the computational load of a pedestrian detection algorithm, detection accuracy is often sacrificed to meet the computational load limitation based on the acceptable cost of the hardware.

In the present paper, a low-cost sparse feature interaction descriptor (SpaFIND) for real-time pedestrian detection with limited computational power is presented. SpaFIND reduces the computational load while maintaining detection accuracy and meeting today's system requirements, such as operation by regular resolution cameras and low-power processors.[4]

The remainder of the present paper is organized as follows. Sec. 2 outlines related research. Sec. 3 describes the details of the proposed method. In Sec. 4, experimental results are provided in order to demonstrate the detection performance and computational efficiency of the proposed method. Finally, the conclusion is presented in Sec. 5.

2. Related Research

Pedestrian detection has been studied for decades. Some pedestrian detection methods focus on the use of radar/LiDAR sensors.[5-7] However, cameras[8-10] (occasionally in combination with LiDAR)[11,12] are used in the majority of studies because they can provide a great deal of information about the target at low cost.

Recently, the deep neural network (DNN) has been leading the field in object detection.[8,9] Deep neural network-based pedestrian detection methods achieve high detection accuracy and dominate virtually every benchmark. However, DNN-based pedestrian detection methods cannot be implemented in current PPSs due to the limited computational resources of low-cost processors that would be used in economy cars. Therefore, more "classical" pedestrian detection methods must be considered. Classical pedestrian detection consists of primarily three components: a feature descriptor, a pedestrian model, and a detector learner. As mentioned in Sec. 1, the present paper focuses on the feature descriptor of pedestrian detection.

The histogram of oriented gradients (HOG)[10] captures the shape information of an object and is robust against local shape transformations. The HOG is considered to be one of the most powerful feature descriptors and has been widely used in pedestrian detection. However, since the HOG captures only the first-order histogram of local appearance, the HOG may generate false positives for some objects such as utility poles and post boxes, which contain pedestrian-like local appearance from the viewpoint of the HOG. In practical use, that is unacceptable for PPSs because the car would perform unnecessary emergency braking in such situations.

In the present paper, we propose a sparse feature interaction descriptor (SpaFIND) that captures the second-order characteristics of object appearance by selectively calculating the pairwise relationship between neighboring elements of the HOG. The proposed SpaFIND feature has the following characteristics:

- SpaFIND not only measures the locally orientated gradient histogram, but also calculates the interaction, i.e., pairwise relationship, between adjacent histogram elements. Therefore, SpaFIND captures the high-level properties of object appearance while inheriting the advantages of the HOG, such as invariance to geometric and photometric transformations.
- SpaFIND calculates only the interactions between the HOG elements that have a significant influence on pedestrian detection. This selective processing produces a sparse yet distinctive feature vector with a relatively small computational load.
- Since a normalization coefficient is obtained prior to feature calculation, SpaFIND does not require calculation of all of the elements in a feature vector, which is necessary for an ordinary vector normalization technique. Consequently, the computational load of SpaFIND is further reduced to meet the PPS requirements of economy cars.

3. Sparse Feature Interaction Descriptor (SpaFIND)

Figure 1 shows how to calculate SpaFIND. First, the detection window is divided into cells of $p \times p$ pixels (red grids in Fig. 1) and then grouped into blocks of $q \times q$ cells (yellow square in Fig. 1). Subsequently, a feature vector $\boldsymbol{H}$ is obtained by calculating the HOG of each block, i.e.,

$$\boldsymbol{H} = (h_1, ..., h_m)^{\mathrm{T}}, \tag{1}$$

where $m = d \times q \times q$, d is the number of quantized gradient orientations. One way to measure feature interactions is to calculate the pairwise correlation between all of the histogram elements of the block. However, as shown by the histogram in Fig. 1, the contributions of weak elements, e.g., elements in Cell 2, are smaller than those of strong elements in the description of the object appearance. Therefore, calculation of all feature interactions would be a waste of computational power. The proposed method selects the elements that have a strong influence on the appearance description and then calculates the correlations between the selected elements only. In

order to eliminate the weak elements, an adaptive threshold τ is calculated as

$$\tau = k \cdot \frac{1}{m} \sum_{h_i \in H} h_i \ , \tag{2}$$

where k is a weight parameter that determines the sparse ratio. SpaFIND will only calculate the correlations between the elements having magnitudes greater than τ in the block, i.e.,

$$\boldsymbol{H}_{\mathrm{D}} = \left\{ h_i \mid h_i > \tau \right\}. \tag{3}$$

In addition, a feature vector should be normalized before being fed to a classifier. An ordinary normalization technique requires that all elements in the vector are obtained in advance, because the normalization factor is calculated from all of the elements. Therefore, even for a sparse feature vector, the computational load can be reduced only slightly. In the proposed method, the normalization factor is estimated from the HOG elements before the feature correlation is calculated. In this case, feature correlation and normalization can be performed simultaneously, which reduces the computational load further. Formally, let us define the correlation feature vectors $\boldsymbol{S}_{\mathrm{t}}$ and $\boldsymbol{S}_{\mathrm{d}}$, which consist of the upper or lower triangle elements of the symmetric matrix $\sqrt{2}\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ and the diagonal elements of the $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$, respectively, i.e.,

$$\boldsymbol{S}_{\mathrm{t}} = \left\{ \sqrt{2}\, h_i h_j \mid i = 1, \ldots, m-1, j = i+1, \ldots, m \right\}, \tag{4}$$

$$\boldsymbol{S}_{\mathrm{d}} = \left\{ h_i h_i \mid i = 1, \ldots, m \right\}. \tag{5}$$

The sum of the squares of all elements in $\boldsymbol{H}\boldsymbol{H}^{\mathrm{T}}$ can be calculated as

$$\sum_{h_i \in \boldsymbol{H}} \sum_{h_j \in \boldsymbol{H}} \left| h_i h_j \right|^2 = \left\| \boldsymbol{S}_{\mathrm{t}} \right\|^2 + \left\| \boldsymbol{S}_{\mathrm{d}} \right\|^2 , \tag{6}$$

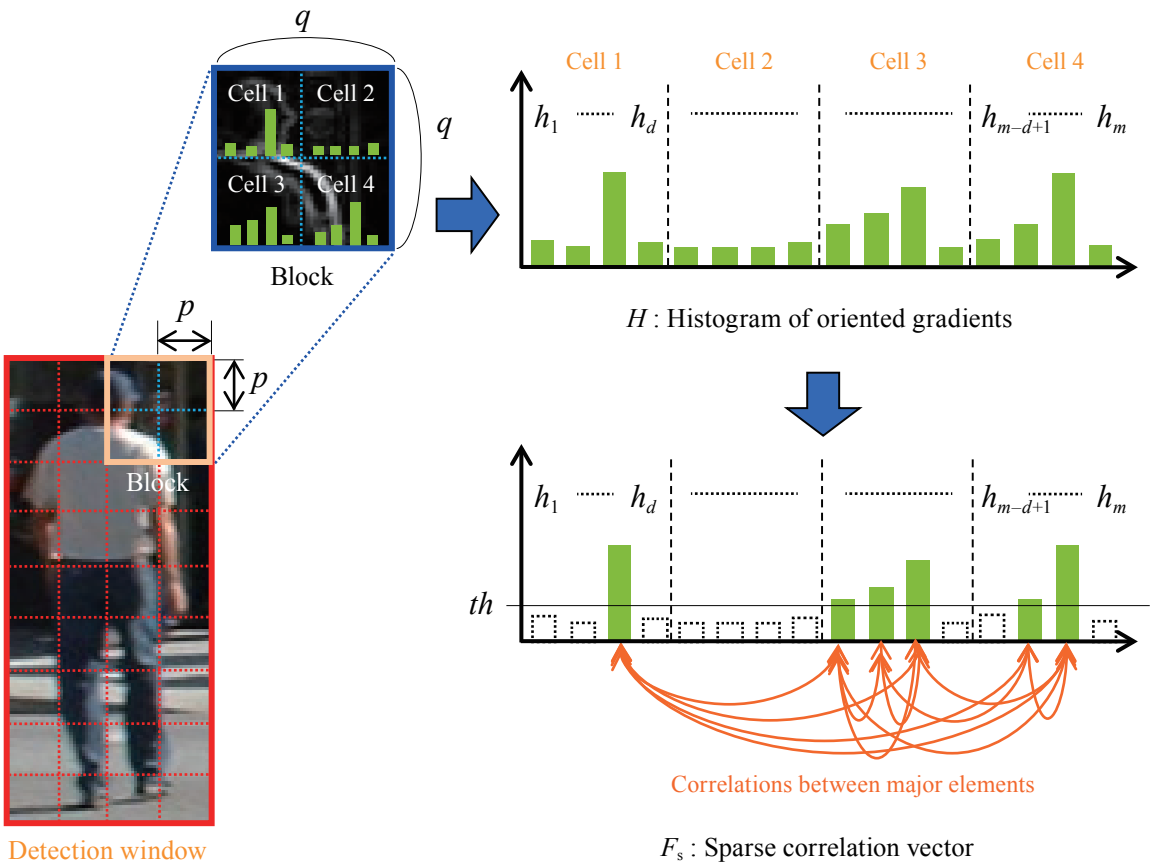


Fig. 1 Computation of a sparse correlation feature vector. The intensity gradient from an image is first calculated, and the HOG divided into d orientations is obtained in each cell, which consists of $p \times p$ pixels. The HOG elements h_i in $q \times q$ cells in a block are then concatenated into a vector, $\boldsymbol{H} = (h_1, \ldots, h_m)^{\mathrm{T}}$, where $m = d \times q \times q$. Finally, the sparse correlations are obtained among pairs $\{h_i, h_j\}$ for the HOG elements, the magnitudes of which are greater than a threshold τ.

where $\|\boldsymbol{S}_t\|^2$ and $\|\boldsymbol{S}_d\|^2$ are the sums of squares of the elements in $\boldsymbol{S}_t$ and $\boldsymbol{S}_d$, respectively. In this case, the correlation feature vector $\boldsymbol{F}$ can be defined as

$$\boldsymbol{F} = \boldsymbol{S}_t \cup \boldsymbol{S}_d \ , \tag{7}$$

where the *L2-norm* of F is

$$\begin{aligned} \|\boldsymbol{F}\|_2 &= \sqrt{\sum_{h_i \in \boldsymbol{H}} \sum_{h_j \in \boldsymbol{H}} \left| h_i h_j \right|^2} \\ &= \sqrt{\sum_{h_i \in \boldsymbol{H}} \sum_{h_j \in \boldsymbol{H}} \left| h_i \right|^2 \left| h_j \right|^2} \\ &= \sum_{h_i \in \boldsymbol{H}} \left| h_i \right|^2 . \end{aligned} \tag{8}$$

For the *L2-norm* vector normalization, i.e., $\boldsymbol{F}' = \boldsymbol{F} / \|\boldsymbol{F}\|_2$, the normalization factor is defined as

$$a = 1 \Big/ \sum_{h_i \in \boldsymbol{H}} \left| h_i \right|^2 . \tag{9}$$

Finally, the correlations among elements in $\boldsymbol{H}_D$ are used to calculate second-order features. The elements generated by $\boldsymbol{S}_d$ are ignored in the final feature vector because autocorrelations do not empirically contribute to detection accuracy.[13] Therefore, the sparse correlation vector is determined as

$$\boldsymbol{F}_s = \left\{ a \cdot h_i \cdot h_j \mid h_i \in \boldsymbol{H}_D \wedge h_j \in \boldsymbol{H}_D \wedge i \neq j \right\}. \tag{10}$$

Due to the sparse interaction and the prior normalization factor estimation, the computational load and memory requirements are greatly reduced.

Figure 2 shows the average magnitudes of the HOG elements in 100 randomly selected images of a real road, in which the HOG elements are sorted in descending order. Here, the number of orientations of the HOG is set to 8 (i.e., d = 8). Moreover, the cell size is 4 × 4 pixels (i.e., p = 4), and the block size is 2 × 2 cells (i.e., q = 2). The obtained HOG elements are *L2*-normalized in each block. As shown in Fig. 2, the eleventh largest element is approximately 1/10 of the largest element. The cumulative ratio of the first eleven elements exceeds 99.5% of the overall magnitude. Therefore, with 1/3 of the HOG elements, the distribution in a HOG feature space is able to be approximated in an error of less than 0.5%. Therefore, only 1/9 (= 1/3²) of the SpaFIND elements generated by the interactions among the selected HOG elements can approximate a full feature interaction among all of the HOG elements with less than 0.5% error. This result demonstrates that the proposed SpaFIND can achieve high-level description capability while maintaining a low computational load. This is also observed based on the experimental results presented in Sec. 4, thereby confirming the effectiveness of SpaFIND.

4. Experiments

The detection performance and computational load of the proposed method were evaluated experimentally using image sequences acquired by an on-board camera. Two datasets were used for the evaluation. The first dataset was collected on a road in Japan, which consisted of approximately 20000 frames and was acquired in the daytime. Each frame in the sequence was a gray-scale image, the image resolution and frame rate were 640 × 480 pixels and 15 fps, respectively. Each frame had annotations as rectangles with borders circumscribed to the pedestrians. The second dataset

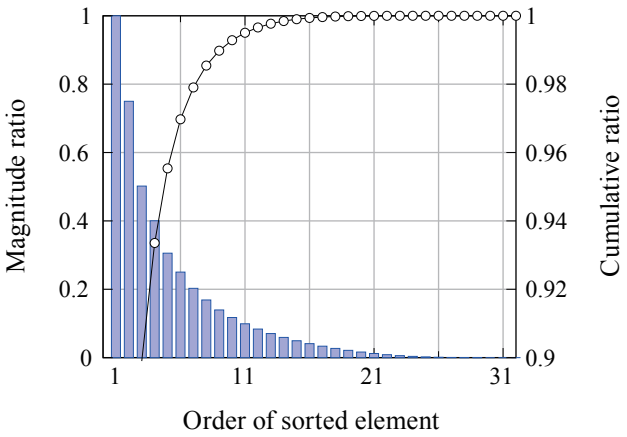


Fig. 2 Analysis of the magnitudes of the HOG elements. The x-axis indicates the order of the sorted HOG elements in descending order. The bars indicate the magnitude of the sorted elements, and the ratio with respect to the largest element of each element is indicated on the left y-axis. The circle symbols indicate the cumulative ratio of the elements, and the ratio with respect to the sum of the elements is indicated on the right y-axis. The HOG elements are sorted in descending order and averaged for all blocks in 100 images that were randomly selected from the Caltech dataset from set00 to set05.

used for evaluation was the Caltech Pedestrian dataset,[14,15] which consisted of approximately 250000 frames in image sequences. Each frame in the sequences was a JPEG compressed color image. However, only the gray-scale information was used in the proposed method. The image resolution and frame rate were 640 × 480 pixels and 30 fps, respectively. In each frame, pedestrians were annotated by rectangles with borders circumscribed around the pedestrians. A total of 350000 pedestrians in images were annotated for approximately 2300 unique pedestrians.

A sliding window technique was used for pedestrian detection. An image in a rectangular window that was scanned sequentially in shrunken images was classified by a classifier trained prior to detection. As for the shrunken images, the interval to become a half-sized image was seven. Non-maximum suppression was applied to combine overlapped detections. The classifier was trained by a linear support vector machine (SVM)[16] with images cropped from image sequences, which were separated from the test images. Hard negative bootstrapping was applied during the training. On the initial stage, the classifier was trained using positive samples, i.e., cropped images of pedestrians and negative samples cropped at random from images in which no pedestrians exist. Then, detection was performed on the images in the same manner as the detection process. The images were cropped from falsely detected regions as hard negatives and were retrained recursively.

Examples of detection results obtained using the proposed method are shown in **Fig. 3**. Rectangles in the image indicate detected regions. All pedestrians were detected, whereas no false detections occurred, as indicated in Figs. 3(a) through 3(d). In addition, a comparison of the detections between the HOG and the proposed method are shown in **Fig. 4**. Figure 4(a) shows a pedestrian that cannot be detected with the HOG. However, this pedestrian was detected using the proposed method. Figure 4(b) shows a false detection with the HOG, whereas no false detections occurred using the proposed method. We hypothesize that the pedestrian in Fig. 4(a) was undetected using the HOG because it was affected by the straight edges of crosswalks and vehicles. In Fig. 4(b), the false detection corresponds to the handlebar of a bicycle. The gradient of the image in the falsely detected region would resemble the gradient of a silhouette of a pedestrian image. These results demonstrate that

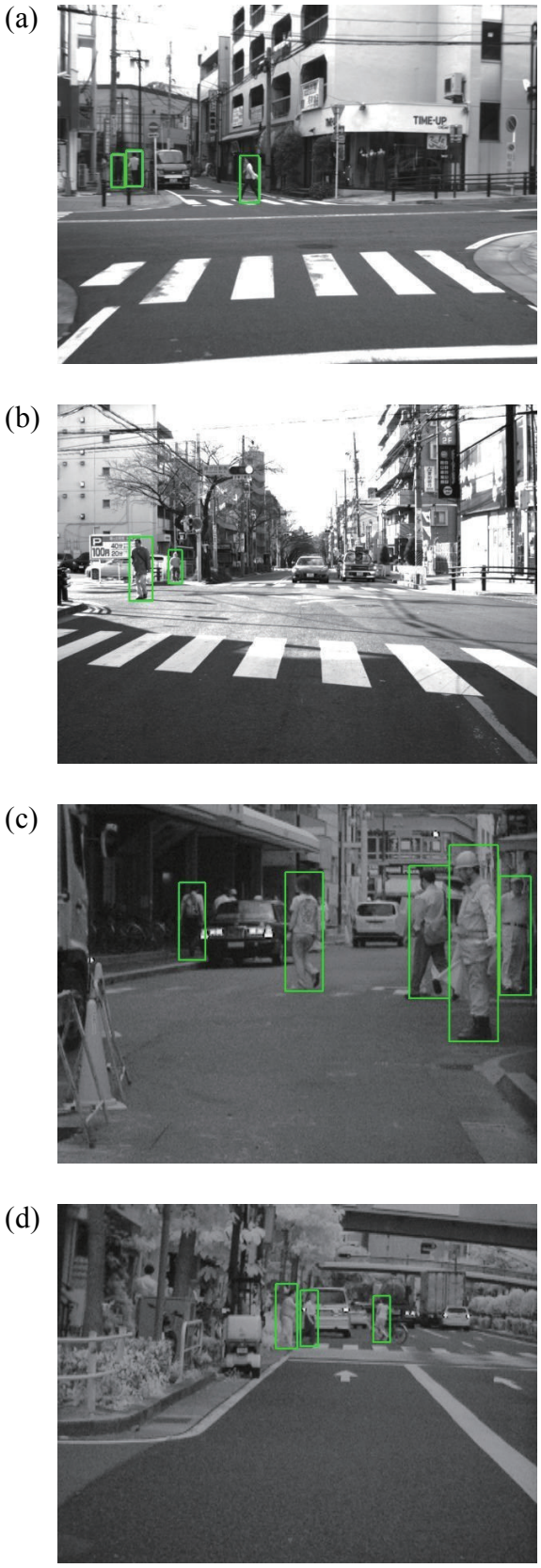


Fig. 3 Detection examples obtained by the proposed method. The green rectangles in each image indicate the detected regions. All pedestrians in the images are detected with no false detections.

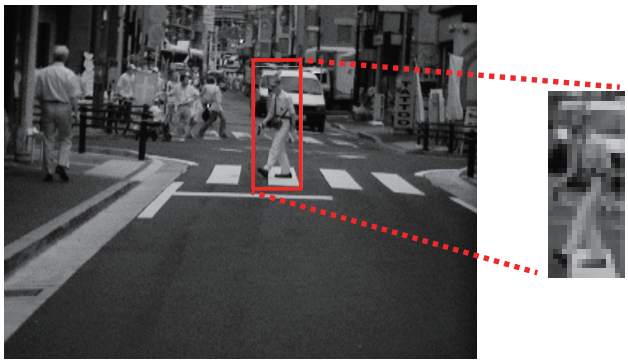
the proposed method has the ability to obtain more enriched properties of appearance than the HOG due to its ability to obtain correlations between HOG elements.

4. 1 Detection Performance

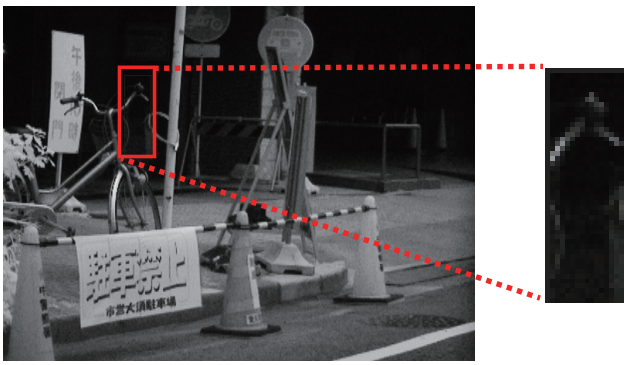
The detection performances of the HOG, the CoHOG[17], and the proposed method were compared under the conditions described above. The two datasets mentioned above were used for detection performance evaluation.

4. 1. 1 Japanese Pedestrian Dataset

The first dataset was acquired in Japan. The training data consisted of 13981 positives and approximately 15000 negatives. The test data contained 8166 pedestrians in 7100 frames. The size of the sliding window was 16 × 48. **Figure 5** shows the detection performances of the HOG, the CoHOG, and the proposed method. The horizontal and vertical axes denote the false detection rate and the detection rate, respectively. The detection rate, i.e., the true positive ratio, is the number of true positives divided by the number of annotated rectangles (8166). A detection is counted as a true positive if the overlap of the detected rectangle with the annotated rectangle is over 50% for both the detected and annotated rectangles. The false detection rate, i.e., the false positive ratio, is the total number of detected rectangles that do not overlap an annotated rectangle and detected rectangles that overlap an annotated rectangle by less than 50% divided by the total number of frames (7100). As for the proposed method, the detection performances were obtained by varying the detection threshold by 0.1. The constant k was varied as 0.0, 0.5, 1.0, 1.5, and 2.0. **Table 1** shows the ratio of the calculated elements in the SpaFIND feature vector at k = 0.0, 0.5, 1.0, 1.5, and 2.0.



(a) Detected correctly only by SpaFIND



(b) Detected falsely only by HOG

Fig. 4 Detection differences between SpaFIND and the HOG.

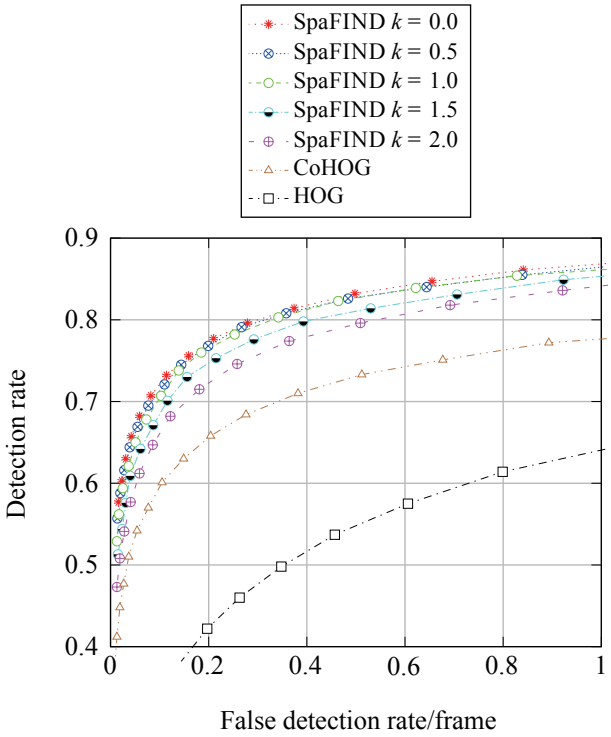


Fig. 5 Detection performance on the Japanese dataset. The horizontal and vertical axes denote the false detection rate per frame and the detection rate, respectively. The detection rate, i.e., the true positive ratio, is the number of true positives out of all annotated rectangles (8166). The false detection rate, i.e., the false positive ratio, is the number of total false detections out of all frames (7100).

In Fig. 5, the proposed method outperforms the HOG and the CoHOG significantly. As the sparsity of a feature vector increases (i.e., as k increases), the degradation of the detection performance of the proposed method increases. However, the detection performance of the proposed method outperforms that of the HOG and the CoHOG, even when the constant k is set to 2.0, for which only 2% of elements in a feature vector are used.

Comparing SpaFIND at $k = 0.0$ (i.e., full interaction), $k = 0.5$ and SpaFIND at $k = 1.0$, the detection performances differ only slightly. The degradation in the detection performance gradually becomes significant when $k > 1.0$, which indicates that the ratio of the effective elements for detection is less than approximately 0.09, because the sparse ratio is 0.09 (Table 1) at $k = 1.0$. This is consistent with the analysis of the effective element ratio in the HOG and supports the hypothesis of sparsity described in Sec. 3. Although, in an actual PPS, k must be defined in consideration of the acceptable computational load and the required detection performance, we herein set k to approximately 1.0 based on the experimental results.

4. 1. 2 Caltech Pedestrian Dataset

In order to confirm that the effectiveness of the proposed method did not depend on the datasets, the detection performance of the proposed method was evaluated using Caltech pedestrian datasets. The Caltech pedestrian dataset consisted of 10 sets of image sequences, from set00 to set09. As a normal setting of the Caltech benchmark, every 30th image from set06 to set09 was used for testing, and every 30th image from set00 to set05 was used for training. The size of the sliding window for the proposed method was 32 × 64. The detection rate was evaluated for annotated ground truths of pedestrians having heights of greater than 50 pixels, including partially occluded pedestrians, more than 64% of whom were visible in the bounding boxes.

The detection performances at sparse constants $k = 0.0$, 0.5, 1.0, 1.5, and 2.0 on a Caltech dataset are plotted in **Fig. 6**. The degradation of detection ability is slight when $k \leq 1.0$ and gradually becomes significant when $k > 1.0$, which is the same as the result on the Japanese dataset (Fig. 5). Therefore, the influence on the detection ability of parameter k is confirmed to be independent of the dataset.

Table 1 Sparsity of computation.

Sparse factor k	Sparse ratio
0.0	1
0.5	0.19
1.0	0.09
1.5	0.04
2.0	0.02

4. 2 Computational Load

4. 2. 1 Processing Speed

For the HOG, the CoHOG, and SpaFIND, the processing times required to calculate features were measured on a desktop PC (CPU: 3.9-GHz Core i7-3930K; no GPU was used to compute features). The processing times required to calculate features

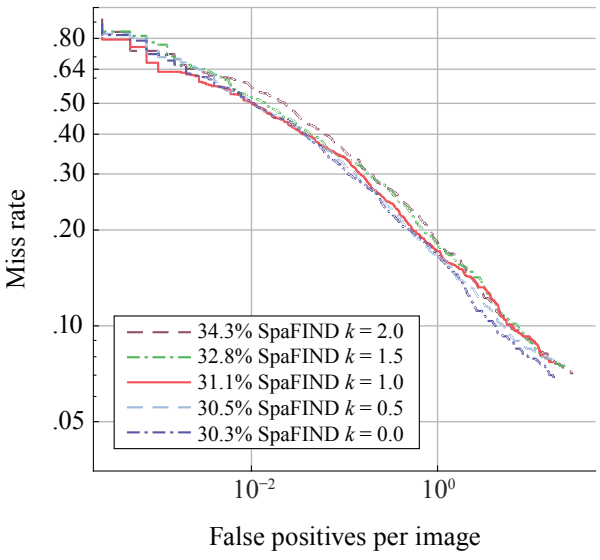


Fig. 6 Detection performance on the Caltech pedestrian test dataset. The horizontal and vertical axes denote the false detection rate and the miss detection rate, respectively. The detection rate was evaluated for the annotated ground truth of pedestrians whose heights were greater than 50 pixels, including partially occluded pedestrians, more than 64% of whom were visible in the bounding boxes.

are shown in **Fig. 7**. The horizontal axis denotes the time required to calculate features in a 16 × 48 pixel sliding window. The number on the right-hand side of each bar is the ratio to the processing time of the HOG. The sparser the feature vector, i.e., the larger k, the less time that is required, as confirmed by the comparison of the processing times of SpaFIND for various values of k. As shown in Fig. 7, the processing time of the CoHOG is nearly 50 times greater than that of the HOG, whereas the processing time is 17.3 times greater than that of the HOG, even at the full interaction (k = 0.0). Moreover, the processing time is reduced to 1.7 times that of the HOG at k = 2.0, and the detection performance is better than that of the CoHOG (Fig. 5). Although both the proposed method and CoHOG compute the second-order statistics of HOG features, these results demonstrate that the proposed method achieves greater detection performance than the CoHOG with a reduced computational load as compared to the CoHOG. The processing time of SpaFIND at k = 1.0 is 0.14 times smaller than that of the full interaction. However, the detection performances of the full interaction and SpaFIND at k = 1.0 are almost identical. This result indicates that the proposed method is able to efficiently select the HOG elements that contribute to detection.

4. 2. 2 Computational Complexity

In order to investigate the computational loads, the computational complexities of the feature extraction methods, the HOG, the CoHOG, and the proposed SpaFIND, are analyzed. A feature extraction algorithm consists of basic mathematical operations, namely, addition, multiplication, and division, and several arithmetic functions, such as square root, exponent-logarithm, and triangle functions. The arithmetic functions can be resolved into basic mathematical operations. Therefore, the required number of basic mathematical operations, i.e., additions, multiplications, divisions, and square-root functions, to extract features per pixel is estimated for all of the algorithms. The estimated numbers of operations for each of the methods are listed in **Table 2**. In the present paper, the computational complexity of the addition is assumed as $O(n)$. For multiplication, the Karatsuba algorithm is assumed. Therefore, the complexity is $O(n^{1.585})$. For division and finding the square root, Newtown's method is assumed. Therefore, the computational complexities are $O(M(n))$. Here, $M(n)$ is the computational complexity of multiplication, and n is the bit length of a computation.[18] In the present paper, the bit length n is defined as 32. Thus, the computational complexities of the multiplication, division, and square-root operations are approximately 7.59 times that of addition. Based on the number of operations (Table 2) and the ratios of the computational complexities of the operations, the total relative computational complexity of each feature extraction is estimated. The estimated computational complexities are rescaled to show the relative computational load from the HOG. The relative computational load is shown in **Fig. 8**. The horizontal axis denotes the relative computational load. The number on the right-hand side of each bar

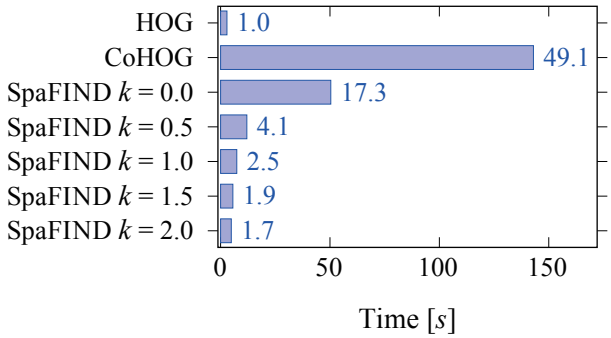


Fig. 7 Processing time of feature extraction. The horizontal axis denotes the time required to calculate the features in a pixels sliding window. The number on the right-hand side of each bar is the ratio with respect to the processing time of the HOG.

Table 2 Number of operations for feature extraction.

	Add.	Mul.	Div.	Sq. root
HOG	165	68	1	2
CoHOG	1691	3379	1	2
SpaFIND k = 0.0	196	1029	1	1
SpaFIND k = 0.5	196	222	1	1
SpaFIND k = 1.0	196	123	1	1
SpaFIND k = 1.5	196	81	1	1
SpaFIND k = 2.0	196	61	1	1

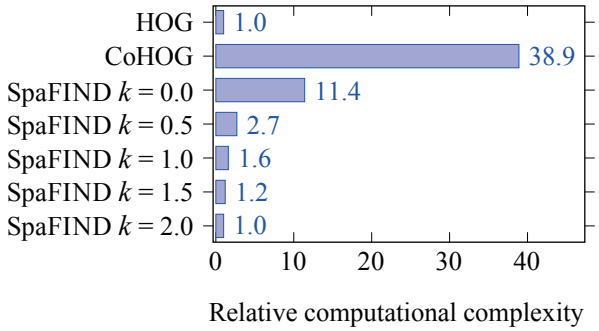


Fig. 8 Analysis of the computational complexity of the proposed method with respect to the HOG. The horizontal axis denotes the relative computational load. The number on the right-hand side of each bar indicates the ratio with respect to the HOG.

shows the ratio with respect to the HOG.

In Fig. 8, the relative complexity of SpaFIND at k = 0.0, i.e., full interaction, which is less than one-third that of the CoHOG, exceeds 10 times that of the HOG. However, the relative complexity of SpaFIND decreases dramatically when $k \geq 0.5$ and does not exceed 1.6 times at k = 1.0, which is the most balanced setting for computational load and detection performance (Figs. 5 and 6).

5. Conclusion

The prevention of collisions between vehicles and pedestrians is important because such collisions often result in fatalities. In order to reduce road traffic fatalities, it is important to implement inexpensive PPSs in economy cars in the near future. In the present paper, a feature descriptor called SpaFIND, which is suitable for inexpensive PPSs due to its effective and low computational load properties, is proposed. In order to achieve high pedestrian detection performance with limited computational power, sparse interactions of the robust HOG features are employed. SpaFIND has the following characteristics: selective correlation of HOG elements, which greatly contributes to describing the appearance of pedestrians in order to prevent description loss in the sparse processing; and prior estimation of the normalization factor, which allows simultaneous computation and normalization of features.

In the future, we intend to implement the proposed SpaFIND-based pedestrian detection system in a low-cost electronic control unit (ECU) of an economy car. SpaFIND is expected to facilitate the widespread deployment of PPSs in economy cars and to enhance the detection performance of existing inexpensive PPSs.

References

(1) World Health Organization, “More than 270000 Pedestrians Killed on Roads Each Year”, *World Health Organization*, <http://www.who.int/mediacentre/news/notes/2013/make_walking_safe_20130502/en/>, (accessed 2017-11-29).

(2) MLIT and NTSEL, “Pedestrian Safety Research in Japan”, *UNECE*, <https://www.unece.org/fileadmin/DAM/trans/doc/2016/wp29grsp/GRSP-59-21e.pdf>, (accessed 2017-11-29).

(3) *Zenkeijikyo*, “*Sireba siruhodo iine kei-jidousha*” (in Japanese), *Zenkeijikyo*, <http://www.zenkeijikyo.or.jp/zenkei17/zen/wp-content/uploads/2017/04/kei-car2014.pdf>, (accessed 2017-11-29).

(4) Kato, T., Guo, C., Kidono, K., Kojima, Y. and Naito, T., “SpaFIND: An Effective and Low-cost Feature Descriptor for Pedestrian Protection Systems in Economy Cars”, *Trans. IEEE Intell. Veh.*, Vol. 2, No. 2 (2017), pp. 123-132.

(5) Fuerstenberg, K. C., Dietmayer, K. C. J. and Willhoeft, V., “Pedestrian Recognition in Urban Traffic Using a Vehicle Based Multilayer Laserscanner”, *Proc. IEEE Intell. Veh. Symp.*, Vol. 1 (2002), pp. 31-35, IEEE.

(6) Ritter, H. and Rohling, H., “Pedestrian Detection Based on Automotive Radar”, *Proc. IET Int. Conf. Radar Syst.* (2007), pp. 1-4, IET.

(7) Premebida, C., Ludwig, O. and Nunes, U., “Exploiting Lidar-based Features on Pedestrian Detection in Urban Scenarios”, *Proc. 12th Int. IEEE Conf. Intell. Transp. Syst.* (2009), pp. 1-6, IEEE.

(8) Krizhevsky, A., Sutskever, I. and Hinton, G. E., “Imagenet Classification with Deep Convolutional Neural Networks”, *Advances in Neural Information Processing Systems 25* (2012), pp. 1097-1105, Curran Associates.

(9) Angelova, A., Krizhevsky, A., Vanhoucke, V., Ogale, A. and Ferguson, D., “Real-time Pedestrian Detection with Deep Network Cascades”, *Proc. British Machine Vision Conf.* (2015), pp. 32.1-32.12, BMVA.

(10) Dalal, N. and Triggs, B., “Histograms of Oriented Gradients for Human Detection”, *Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recognit.*, Vol. 1 (2005), pp. 886-893, IEEE.

(11) Szarvas, M., Sakai, U. and Ogata, J., “Real-time Pedestrian Detection Using Lidar and Convolutional Neural Networks”, *Proc. IEEE Intell. Veh. Symp.* (2006), pp. 213-218, IEEE.

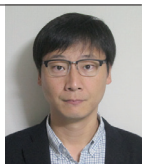
(12) Kidono, K., Naito, T. and Miura, J., "Reliable Pedestrian Recognition Combining High-definition Lidar and Vision Data", *Proc. 15th IEEE Conf. Intell. Transp. Syst.* (2012), pp. 1783-1788, IEEE.
(13) Kato, T., Kidono, K., Kojima, Y. and Naito, T., "Sparse Find: A Novel Low Computational Cost Feature for Object Detection", *Proc. FAST-Zero,* No. TS1-7-3 (2013), pp. 1-5, JSAE.
(14) Dollár, P., Wojek, C., Schiele, B. and Perona, P., "Pedestrian Detection: A Benchmark", *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.* (2009), pp. 304-311, IEEE.
(15) Dollár, P., Wojek, C., Schiele, B. and Perona, P., "Pedestrian Detection: An Evaluation of the State of the Art", *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 34, No. 4 (2012), pp. 743-761.
(16) Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R. and Lin, C.-J., "LIBLINEAR: A Library for Large Linear Classification", *J. Mach. Learn. Res.*, Vol. 9 (2008), pp. 1871-1874.
(17) Watanabe, T., Ito, S. and Yokoi, K., "Co-occurrence Histograms of Oriented Gradients for Pedestrian Detection", *Proc. 3rd Pacific-Rim Symp. Image and Video Technol.* (2009), pp. 34-47, Springer.
(18) Wikipedia, "Computational Complexity of Mathematical Operations", <https://en. wikipedia.org/wiki/Computational_complexity_of_mathematical_operations>, (accessed 2017-11-29).

Figs. 1-3, 5-8 and Tables 1-2
Reprinted from IEEE Trans. Intell. Veh., Vol. 2, No. 2 (2017), pp. 123-132, Kato, T., Guo, C., Kidono, K., Kojima, Y. and Naito, T., SpaFIND: An Effective and Low-cost Feature Descriptor for Pedestrian Protection Systems in Economy Cars, © 2017 IEEE, with permission from IEEE.

Fig. 4
Reprinted from J. Jpn. Soc. Precis. Eng., Vol. 79, No. 11 (2013), pp. 1063-1068, Kato, T., Guo, C., Kidono, K., Kojima, Y. and Naito, T., Sparse FIND: A Novel Low Computational Cost Feature for Object Classification, © 2013 JSPE.

Takeo Kato
Research Fields:
- Machine Learning
- Computer Vision
- Intelligent Vehicles

Award:
- SICE Annual Conference International Award, 2003



Chunzhao Guo
Research Fields:
- Machine Learning
- Computer Vision
- Intelligent Vehicles
- Cooperative Driving
- Robotics

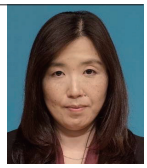
Academic Degree: Dr.Eng.
Award:
- IAPR Best Application Paper Award, Pacific Rim Symposium on Image and Video Technology, 2015

Kiyosumi Kidono
Research Fields:
- Computer Vision
- Intelligent Robot
- ITS

Academic Degree: Dr.Eng.
Academic Societies:
- The Institute of Electronics, Information and Communication Engineers
- The Robotics Society of Japan

Awards:
- SSII Audience Award, Symposium on Sensing via Image Information, 2012
- SSII Best Paper Award, Symposium on Sensing via Image Information, 2013
- IAPR Best Application Paper Award, Pacific Rim Symposium on Image and Video Technology, 2015

Yoshiko Kojima



Research Fields:
- Navigation of Land Vehicles Using Multiple Sensors
- Construction of Spatial Map for ADAS

Academic Degree: Ph.D.
Academic Societies:
- The Institute of Electronics, Information and Communication Engineers
- Information Processing Society of Japan
- IEEE

Awards:
- IPSJ Yamashita SIG Research Award, 2002
- IPSJ Best Paper Award, 2002
- Best Paper Award, 10th International Symposium on Advanced Vehicle Control, 2010

Takashi Naito

Research Fields:
- Environmental Understanding
- Human Monitoring
- Sensors for Advanced Driver Assistance Systems and Autonomous Driving

Academic Societies:
- The Institute of Electronics, Information and Communication Engineers
- The Robotics Society of Japan

Award:
- ACCV The Best Paper, Honorable Mentioned, CVPR, 2010

