特集 Real-time Pedestrian Detection Using LIDAR and Convolutional Neural Networks*

酒井　映　　緒方　淳
Utsushi SAKAI　　Jun OGATA

This paper presents a pedestrian detection system based on the fusion of sensors for LIDAR and convolutional neural network based image classification. By using LIDAR our method achieves a processing speed of over 10 frames/second. The focus of this paper is the evaluation of the effects of fusing the two systems compared to the image-only system. The evaluation results indicate that fusing the LIDAR and image classifier can reduce the number of false positives by a factor of 2 and reduce the processing time by a factor of 4. The single frame detection accuracy of the system is above 90% when there is 1 false positive/s.

Key words: Pedestrian, LIDAR, Camera, Sensor fusion, CNN

1．INTRODUCTION

More than 3000 pedestrians are killed each year in traffic accidents in Japan. There has been a great deal of interest in recent years in the development of pedestrian detection systems that could help reduce the number and impact of these accidents. Most of the proposed systems use a camera as the sensor, because cameras can provide the high resolution needed for accurate classification and position measurement.

The disadvantage of image-only detection systems is the high computational cost associated with classifying a large number of candidate image regions. Accordingly, it has been a trend for several years to use a hierarchical detection structure combining different sensors. In the first step low computational cost sensors identify a small number of candidate regions of interest (ROI).

2．OVERVIEW OF RELATED WORK

The structure of the pedestrian detection systems described in the literature can roughly be divided into region of interest (ROI) detection, feature extraction, candidate classification and tracking. In this section we give a brief overview of the solutions for ROI detection and classification modules in the literature.

2.1　Region of interest (ROI) detection

The purpose of region of interest (ROI) detection is to select a small number of candidate regions in the image that may contain a pedestrian. Frequently used methods include the use of the (relaxed) flat world model, the use of specialized sensors, and the use of a hierarchy of increasingly complex classifiers.

The flat world model makes use of the fact that pedestrians of interest are located on the ground and the image region corresponding to the ground can be computed from the camera geometry. This assumption is too restrictive in practice, because neither the road is completely flat, nor the parameters of the camera are known accurately due to the varying pitch and roll of the camera during driving.

Specialized sensors include stereo vision followed by a clustering, hot-spot detection in far infrared images and laser based detection. The high cost of the sensor is, however, common for these methods. In contrast with these high-cost sensors we propose to use a low-cost automatic cruise control (ACC) LIDAR that is available in cars on the market for almost a decade.

2.2　Classification

The key component of any detection system is the classifier that makes the final decision. The most popular classifier is the support vector machine (SVM) due to its high generalization ability. Other classifiers include neural networks, boosted combination of linear classifiers and template matching. The common point of these classifiers is that they treat the feature extraction and classification problem independently.

The convolutional neural network (CNN) classifier

＊Reprinted with permission from "Proceedings of the 13th World Congress on ITS in London (2006)."

proposed by LeCun,[1] on the other hand, treats the feature extractor as part of the classifier itself. The feature extraction filters are implemented as a hidden layer with shared weights that are optimized together with the classification component. Since the resulting features are tuned to the detection target the classification accuracy is higher than with generic features. Our choice of classifier is, therefore, the CNN extended with the large margin idea of SVM-s.[2]

3．SYSTEM DESCRIPTION

Our detection system can be divided to a LIDAR-based ROI detection module and a convolutional neural network (CNN) based classification module followed by post-processing modules for merging multiple detections and 3 dimensional location estimation (see **Fig. 1**).

3.1 Sensor specifications

Our detection system is relying on a LIDAR sensor for ROI detection and a CCD camera for classifier input. The detailed specifications of the LIDAR are displayed in **Table 1**.

The camera is a CCD Toshiba IK-M44H model. The specifications of the camera are displayed in **Table 2**. The original output of the camera is an interlaced 30 frame/sec NTSC signal but only one in every 3 frames is retained during digitalization.

3.2 Region of interest (ROI) detection

The region of interest (ROI) detector in our system receives the signal from the LIDAR sensor and outputs a list of boxes in 3 dimensional (3D) world-coordinates. The 3D ROI-boxes are obtained by clustering the LIDAR measurements. Each 3D box is projected to the image plane using the intrinsic and extrinsic camera parameters. In real life, however, the extrinsic parameters of the camera are varying due to the pitching of the car. Therefore it is desirable to include a tolerance range for the camera parameters. We call this extended model the relaxed flat world model.

Table 1　Specification of LIDAR parameters

FOV (Field of view)	Horizontal	36° (451 directions)
	Vertical	7.125° (6 planes)
Directional resolution	Horizontal	0.08°
	Vertical	1.425°
Distance resolution		0.01 m
Update rate		100 ms (10Hz)

Table 2　Specification of camera parameters

Type		Toshiba IK-M44H
Sensor		Color CCD
Resolution	Horizontal	640 pixel
	Vertical	480 pixel
Pixel size	Horizontal	0.01 mm/pixel
	Vertical	0.01 mm/pixel
FOV (Field of view)	Horizontal	46.2°
	Vertical	35.5°
Focal length		7.5 mm
Intensity resolution		8 bits/color
Frame rate		100 ms (10 frames/s)

3.3 Convolutional neural network-based classification

In this section we give an overview of the convolutional neural network-based classification module.

Convolutional Neural Networks[1] are a special variant of multilayer perceptrons (MLP) in which the first layers are configured to act as a hierarchical feature extractor. The difference to the usual fully connected MLP is that each processing node in the feature extracting layers (also called "feature maps") is connected to a different subrectangle of the preceding layer and processing nodes in each feature map share the same weight vector. The last layers of the CNN are fully connected, implementing a general purpose classifier over the features extracted by the earlier layers.

The structure and intermediate processing results of the CNN used in our experiments are shown in **Fig. 2**. The size

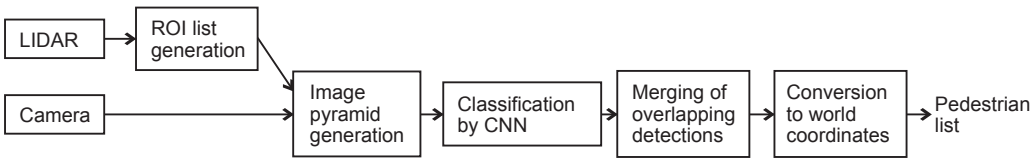

Fig. 1　The block-diagram of our pedestrian detection system

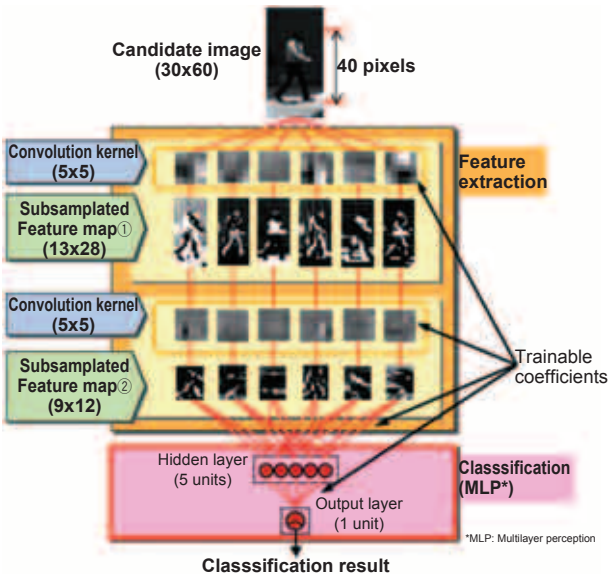

Fig. 2　The structure and intermediate processing results of the convolutional neural network

of the input layer is 30 × 60 pixels. The height of the pedestrian in the input image is 40 pixels during training. The relatively large margin of 10 pixels at the top and bottom are needed in order to compensate for boundary effects. There are 6 feature maps in both the first and second level. Each second-level feature map is connected to exactly one first-level feature map. The size of the hidden layer, fully connected to all level 2 feature maps, is 5 units.

We trained the CNN with the cross-entropy error function using the stochastic-gradient based training algorithm. In order to avoid overtraining the network we applied the large-margin training method described in[2].

3.4　Detection merging

The CNN classifier will give several hits for the same pedestrian. In applications it is desirable to have a single result per pedestrian therefore we perform a multiple detection merging on the raw CNN detections. The multiple detection merging operation is using the algorithm described in 4).

3.5　Position estimation

The final step of the detection process is the estimation of the pedestrian's position relative to the car. We are using a world coordinate system with the positive x axis pointing to the right, the positive y axis pointing upwards and the negative z axis pointing towards the driving direction of the car. The origin is at the center of the front bumper of the car with y=0 being the height of the road. We are assuming that the pedestrian is standing on the road with y=0, while x and z are estimated by inverse perspective projection of the center of the lower edge of the detected bounding box, assuming a flat road.

4．EVALUATION OF DETECTION

4.1　CNN training conditions

The CNN classifier was trained using a large number of 30 × 60 pixel images containing either a pedestrian ("positive samples") or a background image ("negative samples"). Both the positive and negative samples were collected using a camera mounted on the roof of a car.

The negative samples were automatically generated using the bootstrap method of Sung and Poggio.[3] Two hundred input images that contain no pedestrians were used in the bootsrap procedure. The size of the different data sets is displayed in **Table 3**.

4.2　Evaluation data

The evaluation data set comprises 10 video recordings, 10 seconds long each. In order to ensure complete independence of the test data from the training data we recorded the two data sets on different days and in a different city. The resolution of the test images is 640 × 480 pixels.

4.3　Evaluation method

During evaluation we computed a semi-correct world coordinate for each reference bounding box using the inverse perspective procedure. A detection result was

Table 3　The size of different datasets

Date set	Positive samples	Negative samples
Training data (30x60)	37,592	60,000
Test data (30x60)	1000 images, includes both positive and negative regions	

considered correct if the difference between the estimated and reference z coordinates was smaller than 20% of the reference z coordinate of the pedestrian. The tolerance for the x coordinate was fixed at 40 cm. Based on this criterion we computed the ROC curve by changing the detection threshold between the minimum and maximum value.

4.4 Evaluation results

The evaluation has been conducted using 6 different settings for the search procedure. The resulting ROC curves are displayed in **Fig. 3** and the associated processing times are displayed in **Fig. 4**.

In the first 3 experiments we used only the input from the camera and did not use the LIDAR-based ROI detector. We evaluated 3 different settings for the search region. In the baseline setting we conducted a full search of the input image evaluating all possible subrectangles. In the second setting we used a relaxed flat-world model, permitting a ± 3° deviation from the nominal value of the camera pitch. The third setting was using the most restrictive flat-world model.

The second set of 3 experiments used both the camera and the LIDAR input. In the basic setting we used only the horizontal information from the LIDAR ROI-boxes. In the second setting we used both the horizontal and the distance information and computed the permitted feet positions using the relaxed flat-world model with a ± 3° tolerance. In the final setting we computed the feet position using the flat-world model.

The relaxed flat world model reduced the computation time by a factor of 3 and the flat world model reduced it by a factor of 6. The use of the LIDAR information reduced the processing time by an additional factor of 4 compared to the equivalent image-only setting. The LIDAR-based object detection method also reduced the number of false positives by about a factor of 2 compared to the image-only recognition.

Figure 5 illustrates the reduction of false positives due to the LIDAR-based ROI detector.

Green rectangles in the top **Fig. 5(a.)** indicate false positives that were detected by the image-only system. Since there were no LIDAR reflections at the distance corresponding to these false positives, they were eliminated

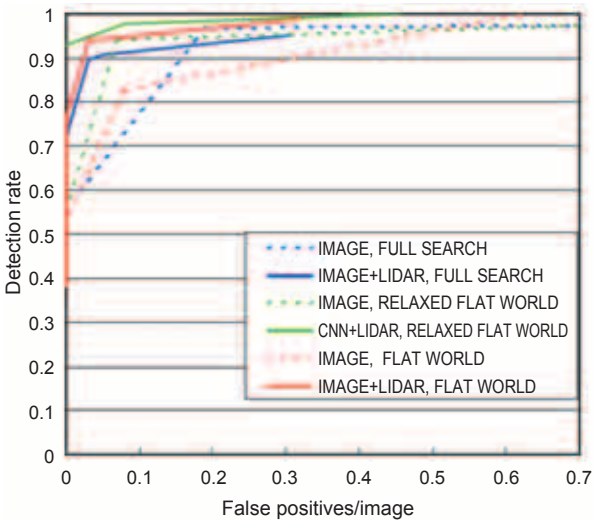

Fig. 3　The ROC curve of the pedestrian detection system with and without the use of LIDAR-based ROI detection

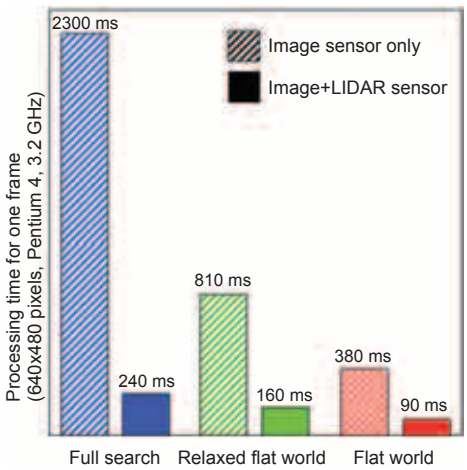

Fig. 4　Average processing time of the pedestrian detection system with and without the use of LIDAR-based ROI detection

in the combined system (bottom, **Fig. 5(b.)**)

5. CONCLUSION

In this paper we introduced a real-time pedestrian detection system utilizing a LIDAR-based object detector and a convolutional neural network-based image classifier. The evaluation results indicate that utilizing the LIDAR information can reduce the amount of false positives by a factor of 2 and reduce the processing time by a factor of 4. We also evaluated the effect of perspective information on classifier performance. The results indicate that using the

a. Image only　　b. LIDAR and image combination

Fig. 5　Illustration of the reduction of false positives due to the use of LIDAR

flat-world model can reduce processing time by about a factor of 6, but it does not give optimal accuracy. The relaxed flat-world model gives a smaller improvement of processing time but it gives higher accuracy.

The proposed system can process real-life video sequences with over 10 frames/second speed on a desktop workstation. The single frame accuracy is 90% detection with less than 1 false positives per second.

Our future work includes integration of a tracking module to improve the detection accuracy and reducing the computation load so that the current frame-rate can be realized in embedded environments.

ACKNOWLEDGEMENT(S)

The authors would like to thank the support of the Image Processing Groups at DENSO IT LABORATORY and DENSO CORPORATION in conducting the experiments.

REFERENCES

1) Y. Lecun, L. Bottou, Y. Bengio, P. Haffner (1998). Gradient-based learning applied to document recognition. In Proceedings *IEEE*, Vol.86, No.11, pp. 2278-2324.

2) M. Szarvas, A. Yoshizawa, M. Yamamoto, J. Ogata (2005). Pedestrian detection with convolutional neural networks. In Proceedings *IEEE Intelligent Vehicle Symposium*, Las Vegas, USA (IV2005).

3) K. Sung, T. Poggio (1998). Example-based learning for view-based human face detection. In Proceedings *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.20, No.1, pp. 39-51.

4) M. Szarvas, U. Sakai, J. Ogata (2006). Real-time pedestrian detection using LIDAR and convolutional neural networks. In Proceedings *IEEE Intelligent Vehicle Symposium*, Tokyo, Japan (IV2006).

＜著　者＞

酒井　映
（さかい　うつし）
システム開発部
走行環境センシング関連の要素技術開発に従事

緒方　淳
（おがた　じゅん）
(株)デンソーアイティーラボラトリ
研究開発グループ
画像処理技術の研究・開発に従事