Automotive Innovation (2018) 1:362–373
https://doi.org/10.1007/s42154-018-0036-6

Blind Spot Obstacle Detection from Monocular Camera Images with Depth Cues Extracted by CNN

Yuxiang Guo[1] · Itsuo Kumazawa[1] · Chuyo Kaku[2]

Received: 28 July 2018 / Accepted: 3 November 2018 / Published online: 26 November 2018

Abstract
The images from a monocular camera can be processed to detect depth information regarding obstacles in the blind spot area captured by the side-view camera of a vehicle. The depth information is given as a classification result "near" or "far" when two blocks in the image are compared with respect to their distances and the depth information can be used for the purpose of blind spot area detection. In this paper, the proposed depth information is inferred from a combination of blur cues and texture cues. The depth information is estimated by comparing the features of two image blocks selected within a single image. A preliminary experiment demonstrates that a convolutional neural network (CNN) model trained by deep learning with a set of relatively ideal images achieves good accuracy. The same CNN model is applied to distinguish near and far obstacles according to a specified threshold in the vehicle blind spot area, and the promising results are obtained. The proposed method uses a standard blind spot camera and can improve safety without other additional sensing devices. Thus, the proposed approach has the potential to be applied in vehicular applications for the detection of objects in the driver's blind spot.

Keywords Coarse-to-fine analysis · Convolutional neural network · Blind spot detection · Principal component analysis · Discrete cosine transformation

Abbreviations

CNN	Convolutional neural network
BSD	Blind spot detection
ADAS	Advanced driver assistance systems
PCA	Principal component analysis
DCT	Discrete cosine transformation

1 Introduction

In the real world, advanced driver assistance systems (ADAS) increasingly act and interact with complex environments to support driving tasks. For this purpose, more and more complex environmental detection techniques are being developed with multiple sensors. These sensors can include sonar, radar, LiDAR, and cameras. To implement proper ADAS function behavior, the functional system is often based on a model that is combined with the sensor signal inputs.

Sensor technology provides the basic external information for the functional module system. In a vehicle, imaging data and ranging data (i.e., distance and speed) are collected by sensors. The reactions to objects are appropriately calculated using the model algorithm. Because of the complexity of vehicle dynamics and the ego vehicle environment, the determined detection algorithm is very often adaptively calibrated. The object detection algorithm is often insufficient for the overwhelming complexity of the real world. These functional insufficiencies can lead to unintended system behavior. To make the detection algorithm more robust, artificial intelligence approaches [1] and models such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs) are increasingly used for deep learning and optimization.

In the area of ADAS, low-cost monocular cameras are widely used to provide driving assistance functions. Such camera sensors can accomplish many tasks, including the

✉ Chuyo Kaku
guozy@cnlbn.com

[1] Department of Information and Communication Engineering, Tokyo Institute of Technology, Tokyo 152-8550, Japan

[2] Research and Development Center, LBN Corporation Ltd., Shanghai 200233, China

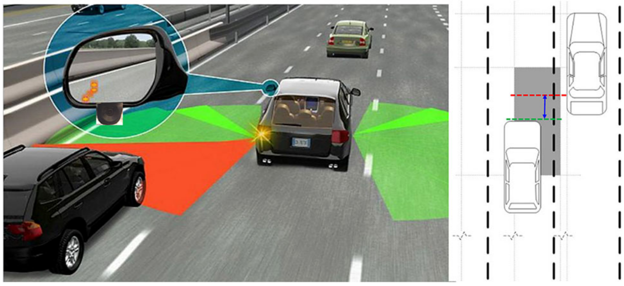


Fig. 1 Driver blind area and side-view camera

detection of lane marks, vehicles, and pedestrians, as well as determining the distance from the ego vehicle to an obstacle. The image data returned by a monocular camera sensor can be used to issue lane departure warnings, blind spot obstacle warnings, collision warnings, or to design a lane keep assist control system. In conjunction with other sensors, it can also be used to implement an emergency braking system and other safety–critical features.

In the real world, many fatal accidents are caused by blind spot ignorance, especially in the case of long trucks or buses negotiating corners. In Europe and Japan, the car rearview and side-view mirrors can be replaced by camera monitoring systems. Therefore, commercial mirror-less cars can be expected soon, and the applied ADAS functions based on camera image processing in such monitoring systems are of vital importance.

In this paper, we focus on the task of vehicle blind spot detection (BSD) through monocular camera image processing. We propose an image processing algorithm to detect the rear side area (Fig. 1), which contains the driver's blind spot area. The image data are captured by a monocular camera from the rear side view. Using a single two-dimensional (2D) image, an algorithm detects the near and far area between the ego vehicle and the obstacle in the driver's blind spot.

Normally, whether a car is close to the blind spot zone is judged using 2D image information. For example, a classifier has been used to determine whether an image region is a vehicle or non-vehicle based on a feature vector [2], and vehicle shadow/light location information in the image can be used to perform blind spot detection [3].

In this paper, we utilize the inferred depth information from the 2D image and relative position to achieve the robust BSD performance augmentation. The depth information is the key to recognizing the near and far areas [4, 5]. In the field of computer vision, constructing a depth mapping from a single 2D image is a challenging task [6]. It is difficult to obtain precise depth information if there is no other reference parameter in a single 2D image. The mentioned depth information in the 2D image is a relative scale other than absolute value. The methods of depth information estimation from images rely on the structure from motion, binocular and multi-view stereo. These observations come from the multiple view of the scene under different lighting conditions. To overcome this limited conditions, the monocular depth estimation as a supervised learning task attempts to directly predict the depth of each pixel in an image by the off-line trained model. The learning-based methods have proved effective for the depth estimation in single images. CNN training objective is proposed to learn to perform single image depth estimation with an image reconstruction loss by the generated disparity images [7]. However, this approach uses binocular stereo footage to enforce consistency between the disparities produced relative to both the left and right images.

In this paper, the monocular depth cue plays the key rule. Supposed that the monocular camera focus is set at an infinite point, the depth information is inferred from the combination of the blur cues [8, 9] and texture cues [10–12]. The blue cues and the texture cues are extracted from the image. The texture cues mean the density of the edge inside the block, while the blur cues mean the degree of blur and the sharpness of the edge. We proposed to conduct the local cues extraction from the monocular image. The depth estimation is derived from the feature comparison between two image blocks selected within a single image. In our proposed algorithm, the following methods are used.

Coarse-to-fine analysis is used to increase the probability of feature extraction from the limited image block area, because the information of interest generated from large-scale features may be lost in high-resolution image blocks, but will be included in lower-resolution image blocks. The low-resolution level is ideal for an overview of the image scene, whereas more detail can be found at higher- or finer-resolution levels [13].

Principal component analysis (PCA) is applied to extract the most important information from the image data [14]. PCA is a statistical technique used for image data compression and data structure analysis. It is used to extract the edge lines as texture cues from a single image. To classify the edge line orientations for higher spatial frequency density, we configure the PCA results into four categories. This allows us to obtain clearer texture cues with higher spatial frequency density along the edge line orientation.

Discrete cosine transform (DCT) is applied to extract the features of texture cues and blur cues from the image. DCT detects the edge line density allocated in the spatial frequency domain with regard to the texture cues. For the blur cues, DCT detects the sharpness of the edges.

Convolutional neural network (CNN) is applied to classify the depth cues from the DCT analysis results. Through deep learning, a better trained neural network model with acceptable accuracy is developed.

The effectiveness of the proposed approach is evaluated through a series of case studies [15]. In this paper, the outline of the proposed approach is described, and one of the test results is shown. The application for BSD purposes is evaluated using real road traffic image data.

The proposed method is found to be able to detect depth cue information for the purpose of BSD. The adaptation to more complex environments and improved algorithm performance will be considered in future work.

2 Algorithm

The basic theory and methods discussed in this paper are now briefly introduced. The proposed method uses both texture cues and blur cues to obtain depth information, whereas existing methods only utilize one of these cues [12]. Depth perception allows us to perceive the world around us in three dimensions and to estimate the distance between ourselves and other objects. One of the techniques for depth perception involves the monocular cues. When perceiving the world around us, many of these monocular cues work together to contribute to our experience of depth estimation. In computer vision, the image taken by a monocular camera is what the computer sees. When the monocular camera makes the focus point at an object that located in the far distance, the image contents nearby to the camera position become blur, while the farther area has the obvious texture cues. This correlation between blur cues and texture cues can be interpreted as depth cues. The monocular depth cues are derived from an image that is characterized by texture cues and blur cues. The texture cues refer to the density of the edges inside the block, whereas the blur cues are the degree of blur and the sharpness of the edges. In this study, we mainly focus on the local cues to simplify the computations.

Starting from this point, we attempt to enhance the effectiveness of the extraction of texture and blur cues. Several relevant concepts are now introduced.

The multi-resolution image representation of coarse-to-fine analysis is discussed first. PCA is not described in detail, as it is a popular method of extracting features from image data. PCA is used for image data preparation in the proposed method, with the edge lines extracted through eigenvalue and vector analysis of the image covariance matrix. The DCT is applied to identify the feature distribution density of the edge lines for the object image. The proposed approach uses the combined application of PCA and DCT.

2.1 Coarse-to-Fine Analysis

An image can be resized with different resolutions, and the resized images can be represented in a pyramid structure. For local analysis, better information coverage can be obtained by using the pyramid structure approach (Fig. 2). The different resolution images are obtained by re-sizing the original image to smaller pixels. In this paper, we define four resolution levels, from level 0 to level 3. Conceptually, the multi-resolution images provide the probability of feature extraction from the limited image block selection. The upper level of the pyramid, i.e., the low-resolution coarse level, is ideal for acquiring an overview of the image scene, whereas details can be obtained further down the pyramid at higher-resolution levels. This is the basis of the coarse-to-fine image analysis process and it is helpful to augment the feature extraction of the blur cues and texture cues.

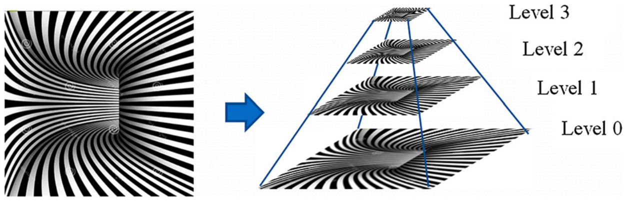


Fig. 2 Image pyramid of coarse to fine

2.2 PCA for Image Processing

2.2.1 Principal Component Analysis

PCA is a statistical method that converts a set of variables into a set of linearly independent variables through an orthogonal transform. The converted variables are the principal components. In the analysis of large multivariate datasets, PCA is often applied to reduce the dimensionality of the data. The process of PCA uses the covariance or correlation of the data. PCA provides a method for data analysis and pattern recognition, and is often used in signal and image processing. Because of its statistical properties, PCA is widely used for dimension reduction and feature extraction, such as edge lines. PCA can also extract the orientation information of the edge lines. In the process of image feature extraction, PCA is often used to identify the main eigencharacteristics for the image dataset through covariance matrix analysis. Suppose the points in an image lie on the x–y plane. From the point data, we can calculate the variance $c(x,\ x)$ in the x-direction and the variance $c(y,\ y)$ in the y-direction. However, the horizontal and vertical spread of the data does not explain the clear diagonal correlation. If the x-value of a data point increases, the y-value also increases, resulting in a positive correlation. This correlation can be captured by extending the notion of variance to what is called the "covariance" of the data: $c(x, x)$, $c(y,\ y)$, $c(x,\ y)$ and $c(y, x)$. These four values can be summarized in a matrix, which is called as the covariance matrix:

$$C(x,\ y) = \begin{bmatrix} c(x,\ x) & c(x,\ y) \\ c(y,\ x) & c(y,\ y) \end{bmatrix} \quad (1)$$

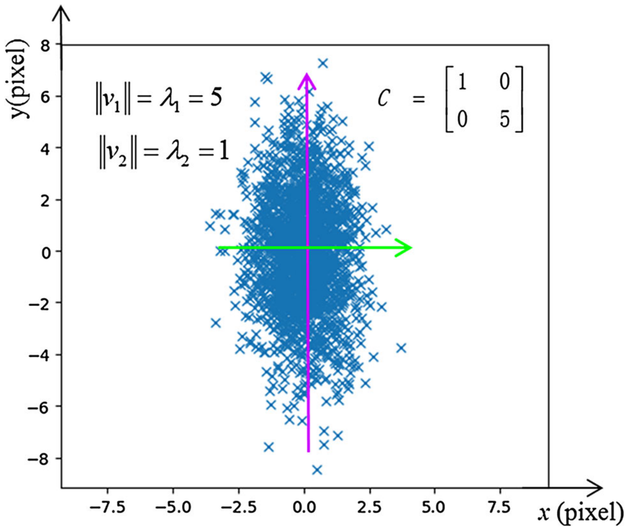


Fig. 3 PCA analysis

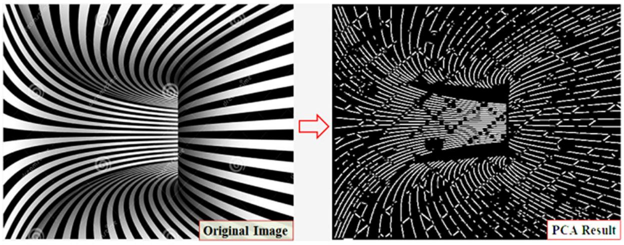


Fig. 4 PCA visual result

From this 2×2 covariance matrix, two eigenvalues λ_1 and λ_2 and two eigenvectors v_1 and v_2 can be obtained. The line characteristics can then be derived from the eigenvectors and eigenvalues. The PCA process is illustrated in Fig. 3, where the eigenvalues and vectors are also listed.

The largest eigenvector of the covariance matrix always lies along the direction of the largest variance of the data, and the magnitude of this vector is equal to the corresponding eigenvalue. The second-largest eigenvector is always orthogonal to the largest eigenvector and lies along the direction of the second-largest spread of the data.

Figure 4 shows the visual result of edge extraction from the original image using PCA.

2.2.2 Four Proposed Categories

We propose that all edge lines extracted from an image by PCA can be configured into four categories (Fig. 5) of computed line orientation according to the eigenvector directions. To the best of our knowledge, no previous studies have considered PCA using the orientation information from the covariance matrix eigenvectors. Under the proposed processing, the density of the edge lines in the spatial frequency domain can be intensively expressed in each category.

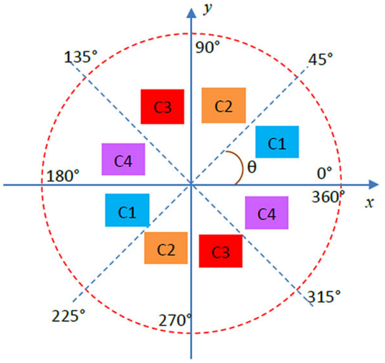


Fig. 5 Category for edge lines

Combined with the coarse-to-fine analysis described in the previous section, the four categories of edge lines obtained by PCA are very helpful in extracting depth cues because of the enhanced effectiveness of the spatial frequency analysis process. The orientation categories are described as follows, where the angle information for each edge line segment is contained in the eigenvector.

Consider eigenvector e_1 and eigenvalue λ_I:

$$e_1 = (x, y) = (\lambda_1 \cos\theta, \lambda_1 \sin\theta) \tag{2}$$

Accordingly, the edge lines can be classified into the following four categories (see Fig. 5).

Category 1 (C1):

$$0^\circ \le \theta < 45^\circ: \quad \sin\theta \in \left(0, \frac{\sqrt{2}}{2}\right) \quad \cos\theta \in \left(\frac{\sqrt{2}}{2}, 1\right) \tag{3}$$

$$180^\circ \le \theta < 225^\circ: \quad \sin\theta \in \left(-\frac{\sqrt{2}}{2}, 0\right) \quad \cos\theta \in \left(-1, -\frac{\sqrt{2}}{2}\right) \tag{4}$$

Category 2 (C2):

$$45^\circ \le \theta < 90^\circ: \quad \sin\theta \in \left(\frac{\sqrt{2}}{2}, 1\right) \quad \cos\theta \in \left(0, \frac{\sqrt{2}}{2}\right) \tag{5}$$

$$225^\circ \le \theta < 270^\circ: \quad \sin\theta \in \left(-1, -\frac{\sqrt{2}}{2}\right) \quad \cos\theta \in \left(-\frac{\sqrt{2}}{2}, 0\right) \tag{6}$$

Category 3 (C3):

$$90^\circ \le \theta < 135^\circ: \quad \sin\theta \in \left(\frac{\sqrt{2}}{2}, 1\right) \quad \cos\theta \in \left(-\frac{\sqrt{2}}{2}, 0\right) \tag{7}$$

$$270^\circ \le \theta < 315^\circ: \quad \sin\theta \in \left(-1, -\frac{\sqrt{2}}{2}\right) \quad \cos\theta \in \left(0, \frac{\sqrt{2}}{2}\right) \tag{8}$$

Category 4 (C4):

$$135^\circ \le \theta < 180^\circ: \quad \sin\theta \in \left(0, \frac{\sqrt{2}}{2}\right) \quad \cos\theta \in \left(-1, -\frac{\sqrt{2}}{2}\right) \tag{9}$$

$$315^\circ \le \theta < 360^\circ: \quad \sin\theta \in \left(-\frac{\sqrt{2}}{2}, 0\right) \quad \cos\theta \in \left(\frac{\sqrt{2}}{2}, 1\right) \tag{10}$$

2.3 Edge Line Density Extraction by Discrete Cosine Transform

2.3.1 Discrete cosine transform

To determine the edge line density and sharpness in the frequency domain, a transformation is required to deal with the image in the spatial domain. The DCT helps separate the image into parts of differing importance with respect to visual quality. The DCT is similar to the discrete Fourier transform: it transforms an image from the spatial domain to the frequency domain, but it can approximate lines well with fewer coefficients using only real numbers. DCTs operate on real data with even symmetry.

The general equation for the 1D DCT (N data items) can be written as:

$$F(u) = \sqrt{\frac{2}{N}} \sum_{i=0}^{N-1} \Lambda(i) \cdot \cos\left[\frac{\pi u}{2N}(2i+1)\right] f(i) \qquad (11)$$

where

$$\Lambda(i) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } i = 0 \\ 1 & \text{otherwise} \end{cases}.$$

The values $F(u)$ are the DCT coefficients of Λ. The general equation for a 2D DCT ($N \times M$ image) is:

$$F(u, v) = \sqrt{\frac{2}{N}} \sqrt{\frac{2}{M}} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} \Lambda(i) \Lambda(j) \cdot \cos\left[\frac{\pi u}{2N}(2i+1)\right] \cdot \cos\left[\frac{\pi v}{2M}(2j+1)\right] f(i, j) \qquad (12)$$

where

$$\Lambda(\xi) = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } \xi = 0 \\ 1 & \text{otherwise} \end{cases}, \quad \xi = i, j.$$

In the proposed method, the spatial frequency response is calculated using the DCT.

2.3.2 Line Density Expressed by DCT Spectrum

As the binarized image texture resulted from PCA process is calculated by DCT, the edge line density in the image can be expressed by the spectrum density of spatial frequency. In other words, the DCT transforms the edge distribution of the image into the frequency distribution.

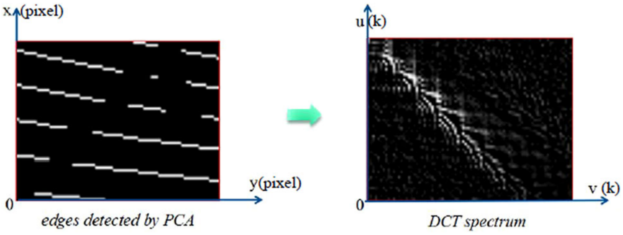


Fig. 6 DCT spectrum for edge detection result

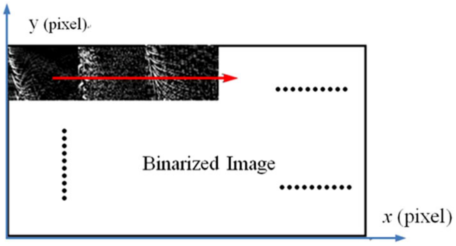


Fig. 7 Spatial frequency calculation procedure

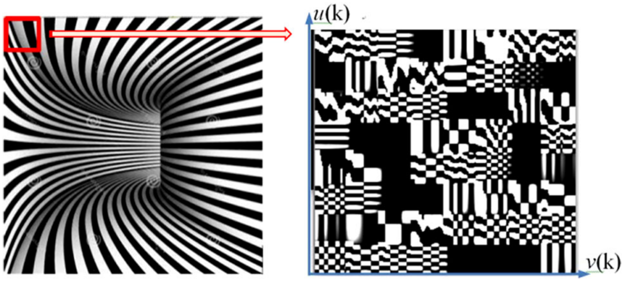


Fig. 8 Block selection and its DCT result

The DCT is used to calculate the spatial frequency of the selected image block. According to the properties of the DCT, low frequencies are concentrated in the top left of the spectrum, and high frequencies are concentrated in the bottom right.

Lower frequencies indicate the sharpness of the edges, which is used to evaluate the blur cues. Both low and high frequencies are used to evaluate the density of the edges inside the block. As an example, the spatial frequency of the DCT spectrum is illustrated in Fig. 6.

In Fig. 6, ***k*** denotes the wave number period in length units, and v and u denote the horizontal and vertical frequencies of 2D waves, respectively.

To process the local cues obtained from the coarse-to-fine representation, we apply a moving window to compute the spatial frequency response. In the coarse-to-fine representation, an appropriate window is defined to separate each resolution level into an integer number. The DCT is then computed from one window to the next, as shown in Fig. 7. An explicit DCT result is shown in Fig. 8. CNN takes the

Fig. 9 CNN structure

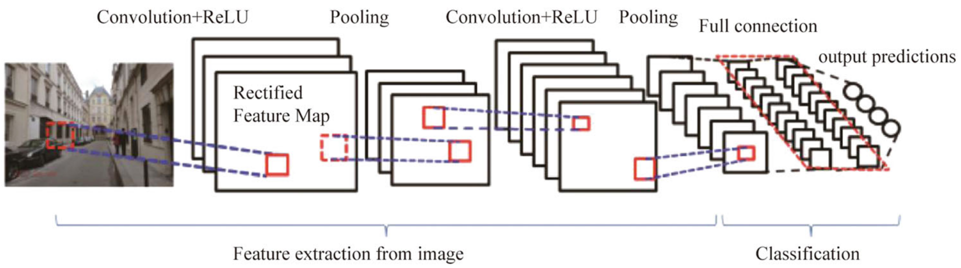


Fig. 10 Proposed structure frame work

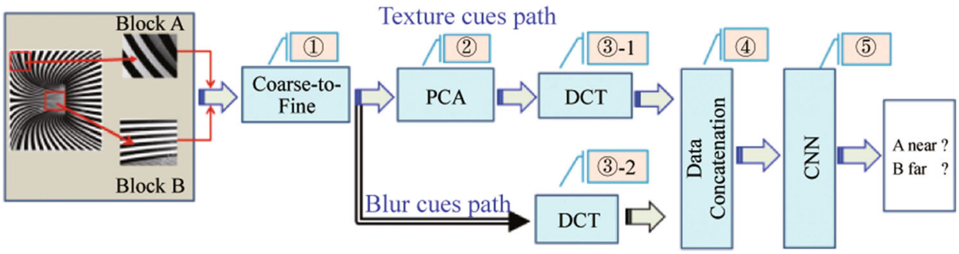
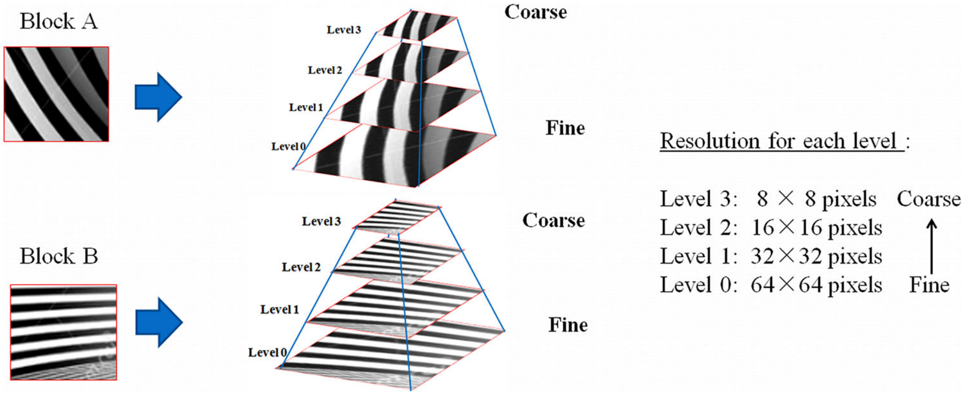


Fig. 11 Coarse-to-fine processing



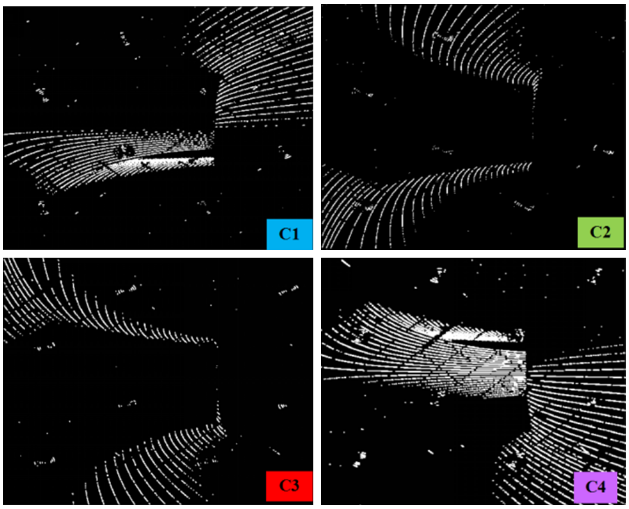


Fig. 12 Four-category PCA results

DCT result as its input and outputs the depth inference from the deep learning process.

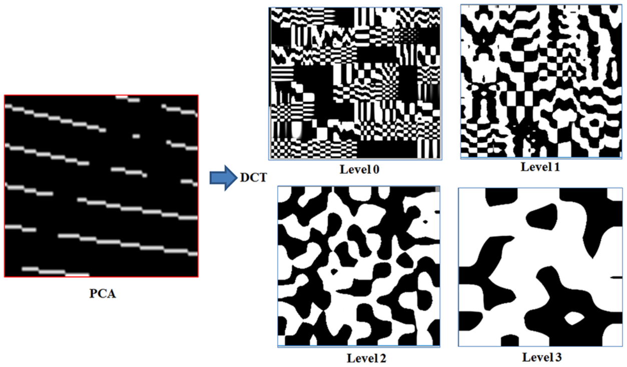


Fig. 13 DCT results: a block at each resolution level

DCT produces more obvious features. In the DCT results, the low-frequency density part is concentrated in the top left of the DCT map, and the high-frequency density part is concentrated in the bottom right [16]. This frequency distribution feature is fed to CNN for deep learning.

The PCA image data processing algorithm and DCT spatial frequency analysis method have been described in this

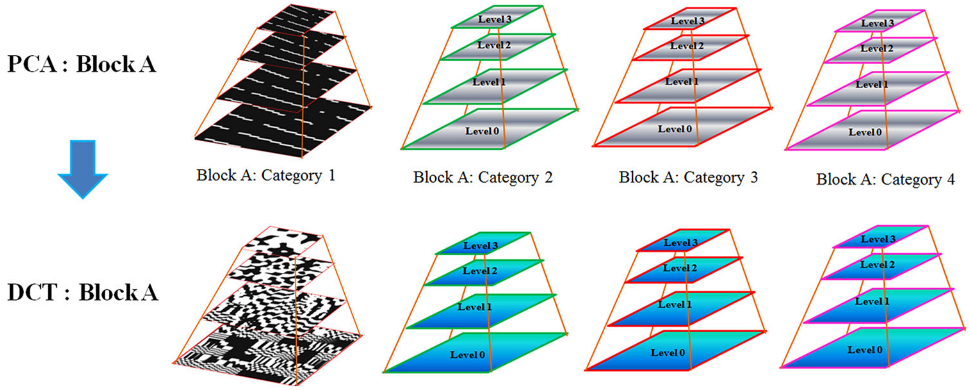


Fig. 14 PCA-DCT: four resolution representations of four categories

section. CNN is described in the next section for the estimation of the depth cues.

3 Depth Cues Derived by CNN

CNN is one of the most powerful deep learning neural networks. Especially in the image processing application, CNN performance is highly evaluated. Through a lot of structure and performance improvements, CNN has become a widely used method. The advantages of CNNs in image processing are as follows:

(1) It reduces the number of weights in the calculation process;
(2) It is robust in terms of object recognition under slight distortions, deformations, and other interference;
(3) It has the characteristics of automatic learning and feature induction;
(4) It is not sensitive to changes in object position in the image.

3.1 Convolutional Neural Network

There are three main neural layers in the CNN structure:

(1) Convolutional layer;
(2) Pooling layer;
(3) Fully connected layer.

A general connection layer is typically used in multilayer sensing neural networks. The main role is to combine the previous convolution layer and the pooling layer to extract the feature vector required for image classification. As shown in Fig. 9, after passing through the convolutional layers and max pooling layers, the data move to the fully connected layer (red dotted area), where the weighting and bias processes are conducted.

The outline of a CNN structure is shown in Fig. 9.

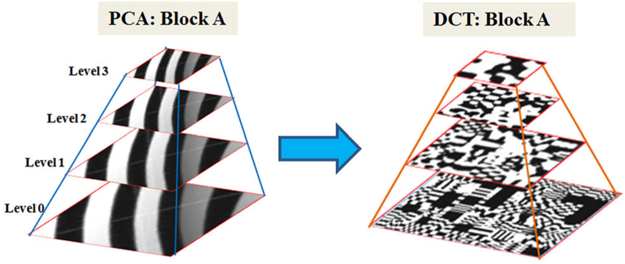


Fig. 15 Blur cue path DCT

3.2 Proposed Structure Framework

Figure 10 shows the framework of the proposed structure. In this framework, the texture cue path and blur cue path denote the preprocessing phases of extracting texture features and blur features, respectively.

CNN is trained to learn the far and near information, which is the basis to classify the depth information of "near" or "far." The depth estimation is then obtained through the feature comparison of two different blocks on the image.

The flow of the proposed structure framework is as follows:

Two blocks from the original 2D image are selected as the inputs. These two blocks are denoted as Block A and Block B.

① Apply the coarse-to-fine analysis to the blocks selected from the image to obtain different resolution pictures. In this study, four resolution levels are used for the coarse-to-fine process. Thus, there are four pictures of different resolutions for each block.
② In the texture cue path, apply PCA with the four-category process to Block A and Block B. The PCA results are then applied to the DCT (③-1) to get the feature map.
③ In the blur cue path, PCA is not applied in order to retain the blur features of Block A and Block B. The DCT (③-2) calculation is applied directly to obtain the feature map.
④ The DCT feature maps are concatenated and aggregated.

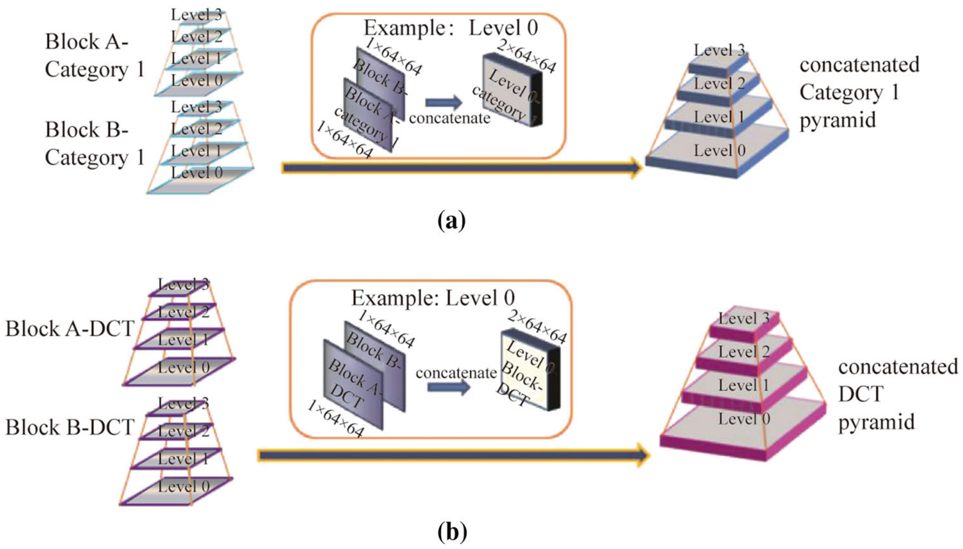


Fig. 16 Data aggregation and concatenation process: (**a**) category 1 aggregation; (**b**) DCT concatenation

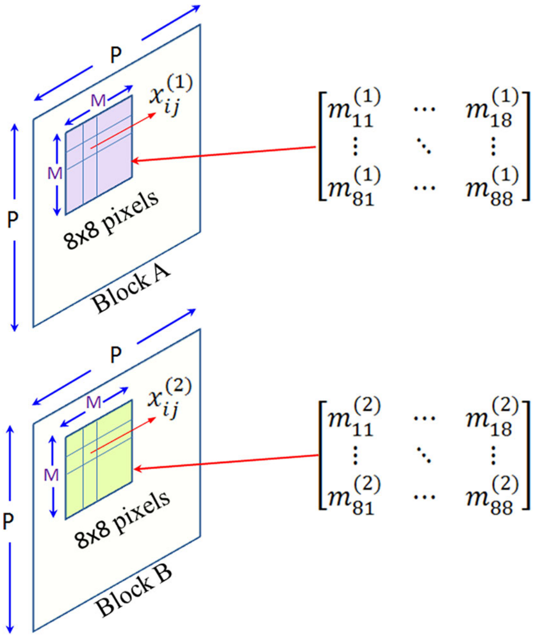

P
M
$x_{ij}^{(1)}$
8x8 pixels
Block A
$\begin{bmatrix} m_{11}^{(1)} & \cdots & m_{18}^{(1)} \\ \vdots & \ddots & \vdots \\ m_{81}^{(1)} & \cdots & m_{88}^{(1)} \end{bmatrix}$
P
M
$x_{ij}^{(2)}$
8x8 pixels
Block B
$\begin{bmatrix} m_{11}^{(2)} & \cdots & m_{18}^{(2)} \\ \vdots & \ddots & \vdots \\ m_{81}^{(2)} & \cdots & m_{88}^{(2)} \end{bmatrix}$

Fig. 17 Concatenation illustration

⑤ Through this concatenation, the output is obtained by the CNN deep learning process.

The details of this workflow are described in the following sections.

3.2.1 Coarse-to-Fine Analysis

The image is preprocessed by the coarse-to-fine analysis with a multi-resolution image representation, where four resolution levels are used in this study. The coarse-to-fine representation can cover more information for image processing.

In this study, Block A and Block B are selected from the original image, and the coarse-to-fine analysis is applied to give the four resolution levels (see Fig. 11). This method is applied to get more texture cue and blur cue information.

3.2.2 Texture Cue Path

PCA is used to extract the edge lines from the image. The four-category edges computed by PCA are used to classify the edge line directions as category 1, category 2, category 3, and category 4. The four-category PCA results are shown in Fig. 12. Each category can then be analyzed to obtain more information in certain directions.

After the PCA preprocessing, DCT is used to calculate the spatial frequency of the selected block. Low frequencies are used to evaluate the degree of blur cues, i.e., the degree of blur and the sharpness of the edge. Both low and high frequencies are used to evaluate the density of the edges inside the block, which is the texture cue.

Figure 13 shows a visualized demonstration of the DCT results from blocks at each resolution level. As we are processing the coarse-to-fine analysis, Fig. 14 partially demonstrates the visual understanding of the PCA-to-DCT representations regarding category 1 of Block A. Categories 2, 3, and 4 are processed using the same approach.

3.2.3 Blur Cue Path

Referring to Fig. 10, PCA is not applied to the blur cue path so as to retain the blur features of Block A and Block B. DCT is directly applied to the original image. The approach illustrated in Fig. 15 is conducted to calculate the DCT for

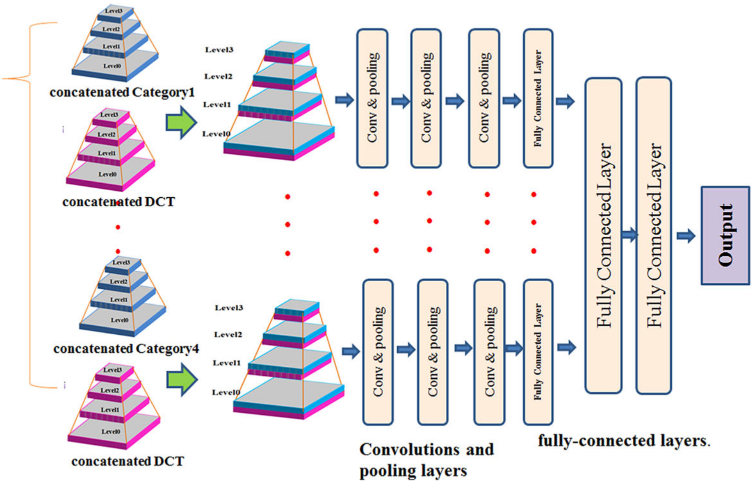


Fig. 18 Diagram of the convolution process

Each category and Block-DCT are concatenated.

Through the processing in convolutions and pooling layers, the data is finally processed in the fully-connected layers.

different resolutions of Block A. The same approach is used to apply DCT to the different resolutions of Block B.

3.2.4 Data Concatenation

The DCT results must be concatenated for input to the CNN. The data concatenation results in an array of size $C \times H \times W$=$2 \times 64 \times 64$.

Figure 16a illustrates the data aggregation process for Block A (category 1) and Block B (category 1) at each resolution; consequently, the process is repeated for categories 2, 3, and 4 at each resolution level. Similarly, the DCTs of Block A and Block B at each resolution level are concatenated, as shown in Fig. 16b.

Figure 17 shows a detailed example of the concatenation of blocks containing P × P pixels with an 8 × 8 kernel size.

$$C_{kl} = \sum_{n=1}^{4}\sum_{i=1}^{8}\sum_{j=1}^{8} m_{ij}^{(n)} x_{ij}^{(n)} \tag{13}$$

where $1 \leqq k \leqq \mathrm{P}-8+1$, $1 \leqq l \leqq \mathrm{P}-8+1$; $m_{ij}^{(n)}$ is the element of the filter mask at coordinates i and j in the block, where n is the order number of the matrix; C_{kl} is the convolution result at position k, l; and $x_{ij}^{(n)}$ is the element of the filter at coordinates $k-i$ and $l-j$. A visual representation of the convolution is shown in Fig. 18.

To provide an overview of the entire calculation architecture, a diagram of the PCA–DCT data preprocessing, data concatenation, and CNN frame diagram are shown in Fig. 19. The texture cue path and blur cue path are expressed in the preprocessing frame, and then, the data concatenation is aggregated for CNN input. Details of the CNN calculation configuration are presented in Fig. 20.

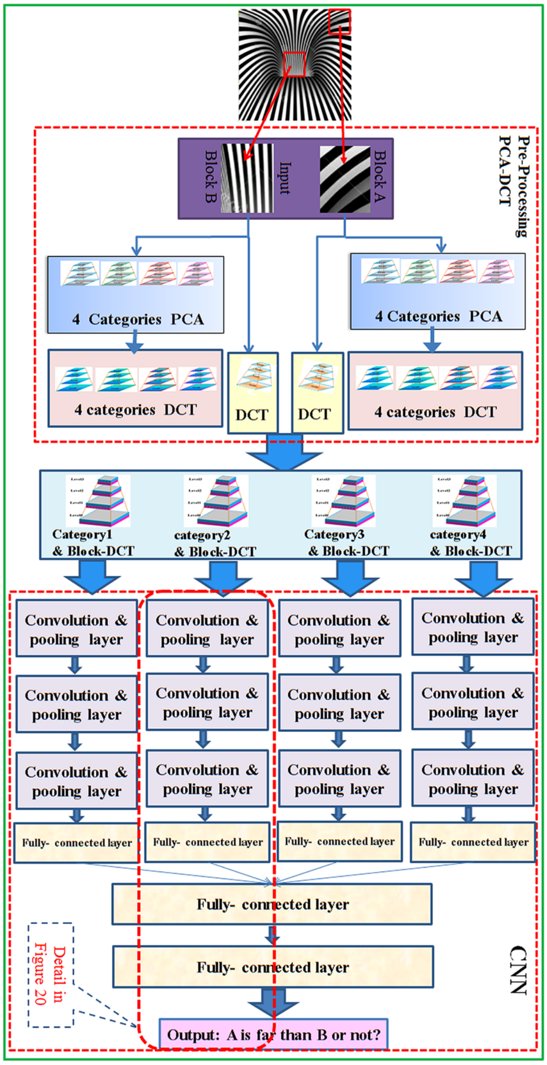


Fig. 19 Calculation architecture

Fig. 20 CNN framework

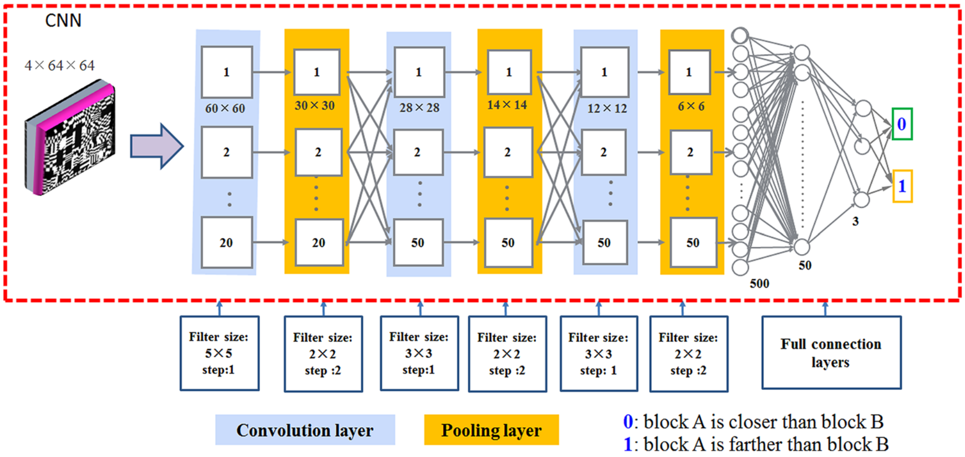


Fig. 21 Image dataset with clear texture

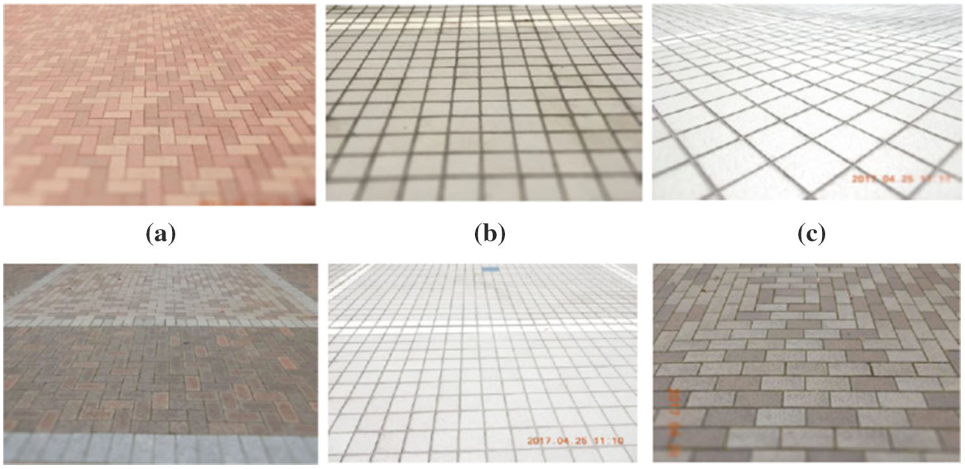


Table 1 Test results

Images for the test	(a)	(b)	(c)	(d)	(e)	(f)	Average
Accuracy of the test	93.13%	95.25%	93.63%	92.38%	97.75%	97.63%	94.96%

3.3 CNN Diagram and Parameters

The target is to evaluate the proposed approach of using texture and blur as cues for estimating the depth information. Thus, the training image data must contain blur and texture.

This net is setup in the Caffe environment and the details are introduced by the example of category 1, at resolution level 0:

(1) Image data of size $4 \times 64 \times 64$ are acquired from the concatenation result.

(2) Convolution layer: the input image is convoluted using a set of filters, and each filter produces one feature map in the output image. This CNN structure uses three layers.

(3) Rectified linear units (ReLU) are applied to get a better stochastic gradient descent (SGD) convergent speed and to prevent from going to dead zone without convergence.

(4) Pooling layer: reduces the dimension of the feature vector output by the convolution layer output by subsampling.

(5) Inner product: the fully connected layer.

In the next section, the CNN model and the experimental results are evaluated to verify the parameter settings.

4 Evaluation

4.1 Experiments for the Proposed Method

For details of the test scenarios, see Ref. [15], in which the effectiveness of the proposed method is demonstrated. In this paper, we present only the test image data (Fig. 21) and experimental results (Table 1). This average accuracy of 94.96%

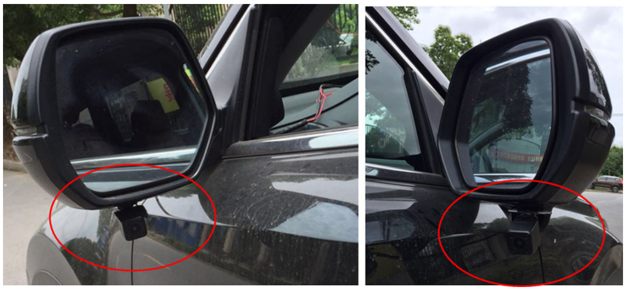


Fig. 22 Measurement setup

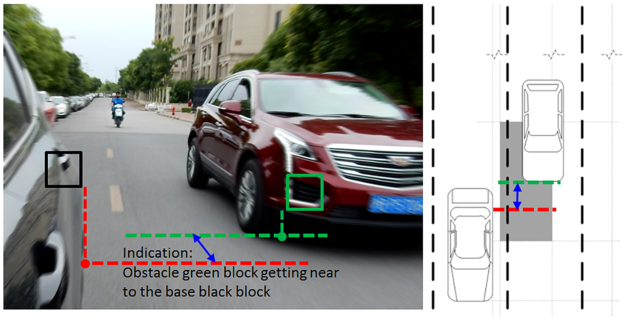


Fig. 23 Depth information estimation

verifies that the proposed approach is effective with regard to the image features of theoretical assumption.

4.2 Experiment for Vehicle Blind Spot Detection

As the focus point of the monocular camera is assumed to be located far beyond the vehicle side mirror view, the images taken by the rear side-view camera have the following features: The near area is the blur zone and the area far from the blind spot is the clear zone.

These features are compatible with the proposed approach. In this study, experiments were conducted to ensure that the proposed method is applicable to vehicle BSD, and the effectiveness was verified using road traffic image data. The camera measurement setup is shown in Fig. 22. The camera was located on the rear side-view mirror, as would be the case in a real system. Our proposed method detects obstacles that are close to the ego vehicle in its blind spot area. The detected depth information is relative to the ego vehicle base block (see Fig. 23). All other blocks are calculated by window shifting (see Fig. 7), and the candidate nearest block with the highest accuracy is output by the supervised deep learning CNN (see Figs. 19 and 20). Currently, the proposed method cannot estimate the absolute distance value. Images under driving conditions in the test cases are presented in Fig. 24.

The CNN model in the proposed method is the same as the trained model used in previous experiments [15]. The test results from images in Fig. 24 are listed in Table 2. The test data preconditions are the same as those described in Ref. [15]. The base block is selected close to the car's rear end, and a total of 1237 image pairs are classified for testing. While referring to the base block in the black line block (Fig. 24), the perceived near-obstacle block is shown by the green line block. The accuracy refers to the model performance with real road scenes.

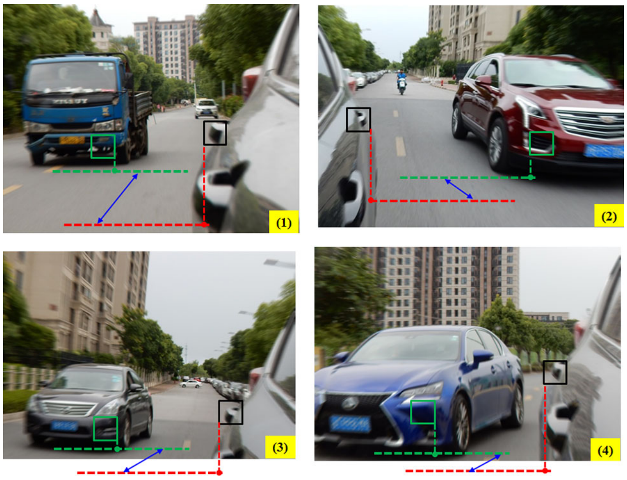


Fig. 24 BSD test cases

In summary, under relatively ideal test conditions, the CNN deep learning model can achieve high accuracy. In terms of vehicle blind area detection, the proposed method is also effective, with an accuracy rate of 78.44%, though the performance must be improved to handle more complex vehicle environments.

5 Conclusions and Future Work

5.1 Conclusions

The proposed method for obtaining monocular depth cues has been described and tested, and the experimental results have verified its effectiveness. The deep learning CNN model achieved satisfactory accuracy under ideal test conditions. Using the same CNN model for vehicle BSD, the results exhibited reasonable accuracy. We therefore believe that the proposed method has the potential to be applied in more complex driving situations. Several advantages and disadvantages are summarized as follows:

(1) The proposed approach combining coarse-to-fine analysis with the four-category PCA process configuration enhances the effectiveness of depth cue extraction.

(2) The proposed method is robust, as the detection is not limited to vehicular shape profiles. Any obstacles approaching the ego host vehicle can be detected,

Table 2 BSD results

Images for the test	(1)	(2)	(3)	(4)	Average
Accuracy of the test	74.13%	78.13%	80.12%	81.38%	78.44%

because the method uses local cues to derive the relative depth information.

(3) The proposed method is not related to the image color. However, the detection performance is not good enough for images taken at night.

(4) Because of the shifting window block in a single image calculation for local cues, the total calculation cost is still the concerning due to the current microprocessor capacity. This may be overcome by using 7-nm chips and specific artificial intelligence chips.

5.2 Future Work

More parameter sets and training datasets need to be applied to the proposed method under vehicle driving conditions related to blind spot area detection. Using huge real driving scene image datasets, a robust CNN model can be developed through the effects of deep learning. Better results with higher accuracy could then be achieved. The output from our method can be used as a warning signal to drivers when changing lanes, alerting them to the presence of a vehicle in their blind spot. In real driving situations, online, real-time blind spot detection is required, and so, improvements in the calculation capacity and efficiency of our algorithm are required. Future research efforts will focus on this considerable challenge.

References

1. Li, J., Cheng, H., Guo, H., et al.: Survey on artificial intelligence for vehicles. Automot. Innov. **1**(1), 2–14 (2018)
2. Li, S.: A new vehicle detection method for blind spot detection system based on DSP. Int. J. Res. Eng. Sci **4**(5), 27–29 (2016)
3. Wu, B.F., Kao, C.C., Li, Y.F., et al.: A real-time embedded blind spot safety assistance system. Int. J. Veh. Technol. (2012). https://doi.org/10.1155/2012/506235
4. Pinard, C., Chevalley, L., Manzanera, A., et al.: Learning structure-from-motion from motion. arXiv:1809.04471v1 [cs.CV]. Accessed 12 Sep 2018
5. Wu, T.Y., Liu, Y.: Position estimation of camera based on unsupervised learning. arXiv:1805.02020 [cs.CV]. Accessed 5 May 2018
6. Vijayanarasimhan, S., Ricco, S., Schmid, C., et al.: Learning of structure and motion from video. arXiv:1704.07804v1 [cs.CV]. Accessed 25 Apr 2017
7. Godard, C., Aodha, O., Brostow, G.J.: Unsupervised monocular depth estimation with left-right consistency. arXiv:1609.03677v3 [cs.CV]. Accessed 12 Apr 2017
8. Hazirbas, C., Leal-Taixé, L., Cremers, D., et al.: Deep depth from focus. Comput. Vis. Pattern Recognition. arXiv:1704.01085 (2017)
9. Carvalho, M., Saux, B.L., Trouvé-Peloux, P., et al.: Deep depth from defocus: how can defocus blur improve 3D estimation using dense neural networks? arXiv:1809.01567v2 [cs.CV]. Accessed 6 Sep 2018
10. Huang, X.J, Wang, L.H., Huang, J.J., et al.: A depth extraction method based on motion and geometry for 2D to 3D conversion. In: Third International Symposium on Intelligent Information Technology Application, pp. 294–298 (2009)
11. Tsai, T.H., Fan, C.S.: Monocular vision-based depth map extraction method for 2D to 3D video conversion. EURASIP J. Image Video Process. **2016**, 21 (2016)
12. Han, K., Hong, K.: Geometric and texture cue based depth-map estimation for 2D-3D image conversion. In: IEEE International Conference on Consumer Electronics (ICCE) (2011)
13. Moukari, M., Picard, S., Simon, L., et al.: Deep multi-scale architectures for monocular depth estimation. arXiv:1806.03051v1 [cs.CV]. Accessed 8 Jun 2018
14. Hua, J.Z., Wang, J.G., Peng, H.Q., et al.: A novel edge detection method based on PCA. Int. J. Adv. Comput. Technol. **3**(3), 228–238 (2011)
15. Guo, Y.X.: Investigation of monocular depth cues obtained through coarse-to-fine image analysis. Thesis of Master Degree, Department of Information Processing, Tokyo Institute of Technology (2017)
16. Oliveira, R.S., Cintra, R.J., Bayer, F.M., et al.: Low-complexity 8-point DCT approximation based on angle similarity for image and video coding. arXiv:1808.02950v1 [eess.IV]. Accessed 8 Aug 2018