International Journal of Innovative
Computing, Information and Control
Volume **10**, Number **2**, April **2014**

 ISSN 1349-4198
pp. 717–728

VIDEOTAPED OBSTACLE EXTRACTION FROM A MOVING CAMERA

Shaohua Qian[1], Joo Kooi Tan[1], Hyoungseop Kim[1], Seiji Ishikawa[1]
Takashi Morie[2] and Takashi Shinomiya[3]

[1]Department of Mechanical and Control Engineering
Kyushu Institute of Technology
Sensuicho 1-1, Tobata, Kitakyushu 804-8550, Japan
{ qian; etheltan; ishikawa }@ss10.cntl.kyutech.ac.jp; kim@cntl.kyutech.ac.jp

[2]School of Brain Science and Engineering
Kyushu Institute of Technology
Hibikino 2-4, Wakamatsu, Kitakyushu 804-8550, Japan
morie@brain.kyutech.ac.jp

[3]Department of Business Administration
Japan University of Economics
Sakuragaokacho 24-5, Shibuya-ku, Tokyo, Japan
t.shinomiya@aa.cyberhome.ne.jp

Received March 2013; revised August 2013

Abstract. *In automatic collision avoidance systems, the ability to detect obstacles is important. This paper proposes a method of automatic obstacles detection employing a camera mounted on a vehicle. Although various methods of obstacles detection have already been reported, they normally detect moving objects such as pedestrians and bicycles. In this paper, a method is proposed for detecting obstacles on a road, even if they are moving or static, by the use of background modeling and road region classification. Background modeling is often used to detect moving objects when a camera is static. In this paper, we apply it to a moving camera case in order to obtain foreground images. Then we calculate the camera motion parameters using the correspondence of feature points between two consecutive images and detect the road region using motion compensation. In this road region, we carry out regional classification. We can delete all objects which are not obstacles in the foreground images based on the result of the regional classification. In the performed experiments, it is shown that the proposed method is able to extract the shape of both static and moving obstacles in a frontal view from a car.*
Keywords: GMM, Motion compensation, Obstacles detection, Road region detection, Car vision

1. **Introduction.** Vehicle has brought great convenience to people's lives as one of the main tools of transport in modern society. However, because of the substantial increase of the vehicles in recent years, the phenomenon of traffic congestion has become more and more serious, and the traffic pollution and accidents have caused social general attention. To solve these difficulties, the autonomous collision avoidance systems have been developing rapidly in recent years. These systems are designed to warn the drivers the presence of obstacles on the road and help them take a necessary action in advance. Among these systems, a vision-based obstacles detection system is the mainstream of current researchers. Segmentation of obstacles in video sequences is a basic task in this system. Accurate obstacles segmentation will improve the performance of obstacles tracking, recognition, classification and motion analysis.

Based on the definition of the obstacles, the existing obstacles detection methods can be separated into the following categories [1]: (i) The method based on optical flow [2,3]. This method is sensitive to vehicle motion, and it fails when obstacles have small or null speeds. In addition, it cannot detect static obstacles, and it can only be used to detect moving obstacles. (ii) The method based on features. This method is often used when the obstacles are defined as a specific kind of objects. It is often used to detect vehicles or pedestrians. Because these obstacles have some specific and obvious features, this detection can be based on searching for these features, such as shape [4,5] and symmetry [6]. (iii) The method based on stereo vision [7,8]. Scene images are captured using two or more cameras from different angles simultaneously, and then the obstacles are detected through matching. This method requires a great deal of time in order to make the necessary calculation.

These existing methods described above have some inadequacies. In the first place, although the third method uses two or more cameras, it is generally accepted that the method which uses a monocular camera is much better because of economic aspect and of processing time. Actually the method using a monocular camera is easier to achieve real-time processing, since it does not have to find correspondence among camera images. In the second place, unlike the first method which detects only moving obstacles, a method which can detect both moving and static objects simultaneously is necessary. It is because static objects such as boxes fallen on the road from a car are also dangerous for drivers. In the third place, most of the existing methods cannot extract the shape of obstacles. They only use a rectangular frame that surrounds an obstacle to represent a detected obstacle. However, this shape information is important for obstacles recognition and classification. If the detected obstacles are judged as a pedestrian, for example, we can recognize his/her motion and may predict his/her next action, if necessary.

In this paper, we propose an obstacles detection method using a vehicle-mounted monocular camera. This camera records the road environment in front of a vehicle when the vehicle is moving, and the computer deals with these captured images to realize obstacles detection. The output of this method is the shape of obstacles. After getting the obstacles information, the drivers can react quickly and make the corresponding actions accurately to prevent car accidents. In this paper, obstacles are defined as objects which protrude from the ground plane, such as pedestrians, vehicles and other 3-D objects on the road (including static and moving objects). Road marks in the road region (e.g., zebra crossings) and objects outside the road region are considered as incorrect obstacles.

When a car is moving forward, stationary objects in a frontal scene are considered as the background, and the foreground can be obtained based on the background model. Because the road has almost no texture, the road can be considered to be static in the frontal video image. In this condition, the foreground image which is obtained from the background model contains the shape of the obstacles and the edges of moving buildings. In order to extract the shape of the obstacles in the foreground image, the following operations are employed. First, calculation of the camera motion parameters is performed using the correspondence of feature points between two consecutive images, i.e., time t image and time $t+1$ image. Second, time t image is warped using camera motion parameters, and it is compared with time $t+1$ image. Then the similar region in these two images is considered as the road region. Third, non-road region in the result of the road region detection is classified as noise region and obstacles region. After the region classification, we have three kinds of regions, noise region, obstacles region and road region. Finally, all objects inside the noise region in the foreground image are considered as noises and can be deleted using the noise region. Road marks in the foreground image are considered as incorrect obstacles inside the road region and can be deleted using the road region. The

shape of the obstacles (e.g., pedestrians) in the foreground image is extracted using the obstacles region.

The advantages of the proposed method over the existing obstacles detection methods are that the proposed method employs a monocular camera to detect both static and moving obstacles on the road, and that it outputs the shape of an obstacle and not a rectangular frame containing the detected obstacle. It should be added that the proposed method can be applied for speed up to 40 km/h which is usually the speed limit within a city.

2. **Outline of the Proposed Method.** Outline of the proposed method is given in the following: Given a video image sequence taken by a moving camera, in Section 3, in order to obtain the shape of obstacles, we employ Gaussian Mixture Model for reconstructing the background and getting the corresponding foreground images. The obstacles are to be included in the foreground images. Because some obstacles outside the road region and noises are also included in the foreground images, we calculate camera motion parameters in Section 4, and then detect the road region using these camera motion parameters in Section 5 to delete these incorrect obstacles and noises. Section 6 describes region classification using the result of road region detection and extraction of obstacles on the road. Experimental procedures are shown in Section 7 and the results are evaluated in Section 8. Finally, the paper is concluded in Section 9. The flow of the proposed method is shown in Figure 1.

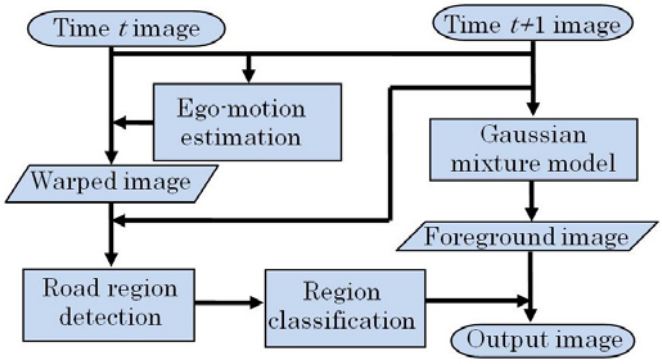


FIGURE 1. Flowchart of the proposed method

3. **Background Modeling.** In this section, we employ a Gaussian Mixture Model (GM M) in reconstructing the background from a video image sequence taken by a moving camera. Using this background model, we get the foreground images.

Detecting obstacles in a video image sequence is an important problem in computer vision, with applications to several fields, such as video surveillance and target tracking. A typical method of detecting obstacles is background modeling, by which we can get the shape of an object directly [9]. Numerous approaches concerning this problem differ in the type of used background models and the procedure used to update the model. There are two kinds of background modeling methods: non-adaptive methods and adaptive methods. Most researches have abandoned non-adaptive methods because of the need of manual initialization. A standard method of adaptive modeling is Gaussian Mixture Model (GMM) [10] because GMM is robust to illumination change. The GMM models the gray value of each pixel on an image by a mixture of K Gaussian distributions. Different Gaussian distributions in the mixture represent different pixel values. The background

models are determined by the parameters of these Gaussian distributions. Then we regard those pixels which do not match the background distributions as foreground pixels. In order to deal with illumination change, the mixture models are updated frame by frame.

The GMM was always employed in a fixed camera case. However, in this research, since a camera is mounted on a vehicle, when the vehicle is moving on the road, the camera is as well moving. In order to employ GMM in a moving camera case, we need to construct a virtual scene using the realistic scene. In this virtual scene, the camera is regarded as static. Then we employ the GMM in this virtual scene in reconstructing the background.

3.1. **Virtual scene construction.** In this research, since a camera is mounted on a vehicle, when the vehicle is moving, the camera is moving as well. This is the realistic situation. However, when we see the frontal scene in the frontal video image, the camera can be considered to be static, and then buildings, the road and static objects are moving according to the relative motion. Moreover, since the road has almost no texture, we can assume that the road is static in the frontal video image. Thus, the virtual scene will be defined as the frontal scene (in the videotaped image) with the assumption of the road being static. In this virtual scene, the camera is static; the road area which is classified as the background is static; objects (including static and moving objects) and pedestrians on the road, buildings, road marks and zebra crossings which are classified as the foreground are moving. Then we employ the GMM to reconstruct the background in this moving camera case.

4. **Ego-Motion Estimation.** In order to detect the road regions, first we need to calculate the camera motion parameters, we also call it ego-motion estimation. Accurate estimation of the ego-motion of the vehicle relative to the road is important for autonomous driving and computer vision based on driving assistance. In this section, we will describe a robust method of computing the ego-motion of the vehicle relative to the road. In this method, two consecutive images are used at any time, i.e., the image I_t taken at time t and the image I_{t+1} taken at time $t+1$.

4.1. **Feature points detection.** In the image I_t, we detect feature points m_t using Harris corner detector [11], then detect the corresponding points m_{t+1} in the image I_{t+1} using Lucas-Kanade method [12]. Next we use RANSAC [13] to delete *outliers* among these point pairs. Now we get a number of point pairs $m_t \leftrightarrow m_{t+1}$.

4.2. **Fundamental matrix.** The fundamental matrix F is a unique 3*3 rank 2 homogeneous matrix which satisfies

$$m_{t+1}^T F m_t = 0 \tag{1}$$

for all point pairs $m_t \leftrightarrow m_{t+1}$. In this paper, we use the 8-point algorithm [14] to calculate the fundamental matrix.

The key to success with the 8-point algorithm is proper careful normalization of the input data before constructing the equations to solve. In the case of the 8-points algorithm, the suggested normalization is translation and scaling of each image so that the centroid of the reference points is at the origin of the coordinates and the distance of the points from the origin is equal to or less than $\sqrt{2}$.

4.3. **Camera motion parameters.** The camera motion parameters consist of a 3*3 rotation matrix R and a 3*1 translation matrix T. The relationship between the fundamental and essential matrices is

$$E = K^T F K \tag{2}$$

Here, K is a camera inner parameters matrix, and E is an essential matrix.

The essential matrix can be represented by motion parameters of a camera between two images, the rotation matrix R and the translation T.

$$E = [T]_X R \tag{3}$$

Here

$$[T]_X = \begin{bmatrix} 0 & -t_Z & t_Y \\ t_Z & 0 & -t_X \\ -t_Y & t_X & 0 \end{bmatrix} \tag{4}$$

is the corresponding skew-symmetric matrix of the translation T.

By applying the singular value decomposition to the matrix E, we can calculate the rotation matrix R and the translation T.

5. **Road Region Detection.** Because the camera is moving, the foreground which is obtained from the background modeling often contains a lot of noises. These noises are caused by the objects outside the road region. In order to delete these noises, we need to detect the road region. In this section, we detect a road region using the method based on motion compensation. This method uses two consecutive image frames, and warps the first image according to the geometrical relationship between two consecutive images. This geometrical relationship is expressed by the camera motion parameters. Then the road region is extracted by comparing the warped image with the second image.

We assume all the 3-D points in the world coordinate system which correspond to the pixels in the first image are on the road plane. Using the camera motion parameters and epipolar geometry, we can warp the first image and get the warped image. Then we compare the second image (real image) and warped image (virtual image). The differences between these two images are caused by the 3-D points which are not on the road plane. So a similar region in these two images is regarded as a road region.

5.1. **Motion compensation.** In this road region detection, two consecutive images are used at any time, i.e., the image I_t taken at time t and I_{t+1} taken at time $t+1$ are used to estimate the road region at time $t+1$. Because a camera is moving in this research, the camera is in different locations in these two moments. The camera locations and coordinates at time t and $t+1$ are shown in Figure 2. Here, two blue points are camera lens at time t and time $t+1$, respectively; (X_t, Y_t, Z_t) and $(X_{t+1}, Y_{t+1}, Z_{t+1})$ are camera coordinates at time t and time $t+1$, respectively; (X_W, Y_W, Z_W) is the world coordinate which coincides with the camera coordinate at time t: M is a 3-D point in the world coordinate system: m_t and m_{t+1} are the 2-D points in the virtual image planes which correspond to the 3-D point M: Matrix R and matrix T are camera motion parameters from time t to time $t+1$ which have been calculated by the method explained in Section 4.

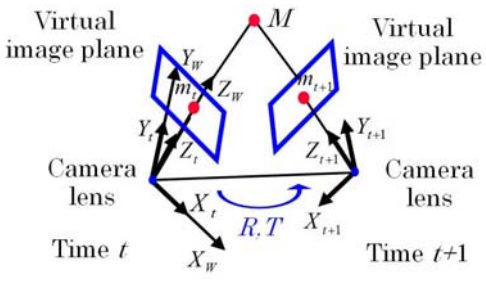


FIGURE 2. Camera locations and coordinates

A projection equation indicates the relationship between 2-D points in the image plane and 3-D points in the world coordinate system. Projection equations at time t and time $t+1$ are given by

$$\lambda m_t = K[I \quad 0]X \tag{5}$$

$$\lambda m_{t+1} = K[R \quad T]X \tag{6}$$

Here m_t and m_{t+1} are the 2-D points in the virtual image planes which correspond to the 3-D point M. X is the coordinate of M in the world coordinate system. Matrices R and T are camera motion parameters from time t to time $t+1$. λ is a constant. K is the camera internal parameters matrix, which can be obtained from camera calibration.

Equations (5) and (6) can be rewritten in terms of the known coordinates m_t and m_{t+1} as follows:

$$\lambda \begin{bmatrix} u_t \\ v_t \\ 1 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{7}$$

$$\lambda \begin{bmatrix} u'_{t+1} \\ v'_{t+1} \\ 1 \end{bmatrix} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{8}$$

Here $(u'_{t+1}, v'_{t+1}, 1)$ is the coordinate of 2-D point in the warped image corresponding to $(u_t, v_t, 1)$ in the first image.

Because we assume that all 3-D points in the world coordinate system are on the road plane at time t, Y is equal to the height of a camera above the ground in (15). Based on (15) and (16), we can calculate the coordinates of 2-D points $(u'_{t+1}, v'_{t+1}, 1)$ in the warped image. Then we use the coordinates (u'_{t+1}, v'_{t+1}) to create the warped image. In the warped image, the pixel value of the coordinate (u'_{t+1}, v'_{t+1}) is assigned to the pixel value of the coordinate (u_t, v_t) in the image I_t.

5.2. **Normalized cross-correlation.** The obtained warped image assumes that all 3-D points in the world coordinate are on the road plane. According to this assumption, we can compare the real image and the warped image and all differences between these two images are caused by the points which are not consistent with the above assumption. It also means that these points are not located on the road plane. So we define the road region as the high similarity region between the real image and the warped image.

Here we use normalized cross-correlation (denoted as NCC) [15] to measure the similarity of two images. NCC is a method of measuring similarity of two signals.

6. **Obstacles Extraction.** In order to extract the shape of obstacles in the foreground images, we need to delete two kinds of things: road marks and noises outside the road. For road marks, we can use the result of the road region detection to delete them. For the noises outside the road, when we use the result of the road region detection to delete noises, it also deletes the obstacles outside of the road. In order to solve this problem (reserving the obstacles and deleting the noise outside the road), we need to divide the non-road region into obstacles region and noise region. Then we use this noise region to delete the noise outside the road in the foreground image.

(a) (b) (c)

FIGURE 3. (a) The result of road region detection, (b) road region template image, (c) the result of region classification

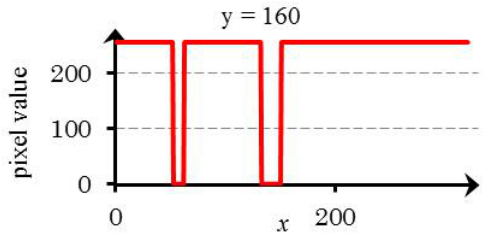


FIGURE 4. Pixel values distribution of the 160^{th} row in the road region template image

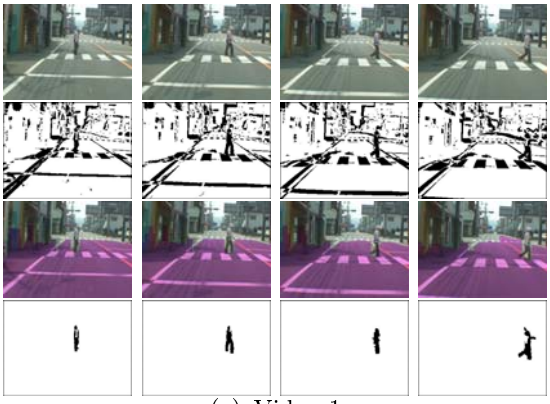
6.1. **Region classification.** Here we want to divide the non-road region into obstacles region and noise region. Figures 3(a) and 3(b) show the result of road region detection and the corresponding road region template image. In this road region template image, *black* pixels are road pixels, whereas *white* pixels are non-road pixels. If we check the pixels of one particular row in this image, we get a curve as shown in Figure 4. Based on this curve, we consider *white* regions (high values) which have two adjacent *black* regions (low values) both on the left and right sides as the obstacles region. We check each row in the road region template image to carry out region classification. Because, in this research, the obstacles which we want to detect are defined as the 3-D objects located on the road, other 3-D objects located outside the road are no danger to the driving (classified as noise outside the road). Based on this definition, we know the obstacles must be located in the road region (surrounded by a road region) or have adjacent road regions on both left and right sides. Figure 3(c) shows the result of the region classification.

6.2. **Post-processing.** In the foreground image, black pixels mean the pixels of foreground objects. These black pixels contain the pixels of obstacles, the pixels of road marks and the noises. In order to extract the shape of obstacles, we should change the black pixels which represent the road marks and noises to white. We check each black pixel's position in the result of region classification (shown in Figure 3(c)). If the current black pixel is located in the noise region (white region), this black pixel is considered as the noise and is changed to white in the foreground image. If the current black pixel is located in the road region (black region), this black pixel is considered as the road mark and is changed to white in the foreground image. Because the road region detection method based on motion compensation uses the 2-D and 3-D information, the result of this road region detection is actually the road plane region detection. The road marks are located on the road plane. So the results of road region detection contain the road marks region. Using this road region, we can delete the road marks in the foreground image. If the current black pixel exists in the obstacle region, it is left unchanged. By

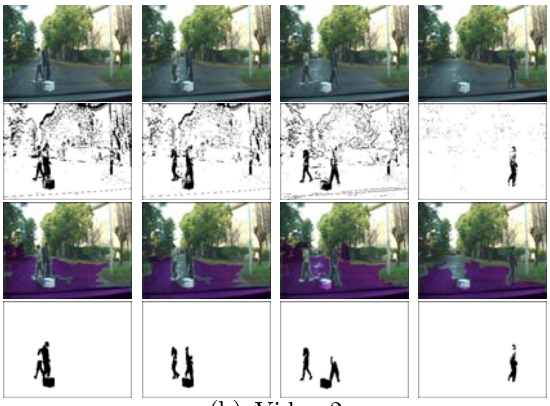
this operation, road marks and other noises are deleted. Then we carry out an erosion operation and regional expansion as in the post-processing.

7. **Procedures.** The camera, which is fixed in front of the vice driver's seat, records the road conditions in front of the car when the car moves forward. Because correct obstacles are defined as arbitrary objects that protrude from the ground plane in the road region, including static and moving objects. Road marks in the road region (e.g., zebra crossings) and objects outside the road region are considered as incorrect obstacles. According to this definition, a correct object in video 1 is a pedestrian; correct objects in video 2 are two pedestrians and a box.

We reconstruct the background model using a Gaussian mixture model in the input images. In Figure 5, the first row shows the input images and the second row shows the corresponding foreground images. Because the camera is moving, the foreground images as shown in Figure 5 contain a lot of noises. This noise is mostly caused by the objects outside the road region. In order to delete this noise, we need to detect the road region. Next, we calculate the camera motion parameters using the correspondence of feature points between two consecutive images, time t image and time $t+1$ image. Then we detect the road region in the time $t+1$ image using motion compensation. The third row of Figure 5 shows the results of road region detection. In these resultant images, the purple color region means the road region. Finally, we carry out region classification in the road region template image, and then delete road marks and noises in the foreground

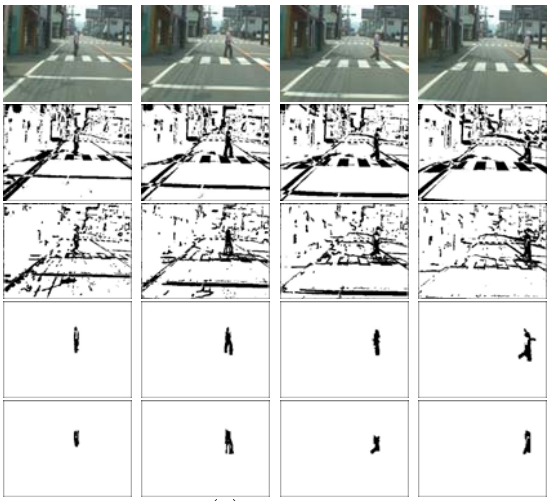


(a) Video 1

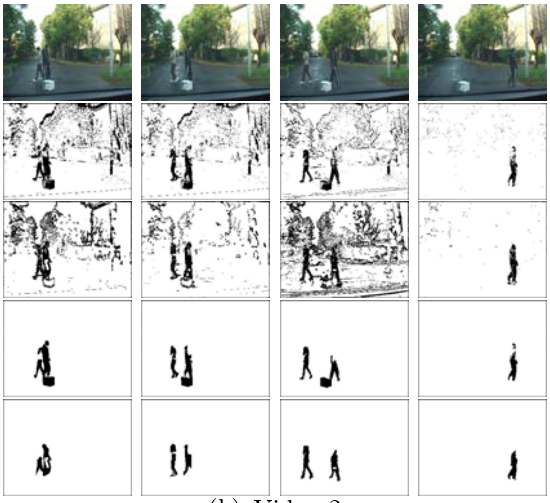


(b) Video 2

FIGURE 5. Experimental results. First row: input images, second row: foreground images, third row: the result of road region detection, fourth row: the result of obstacles detection.



(a) Video 1



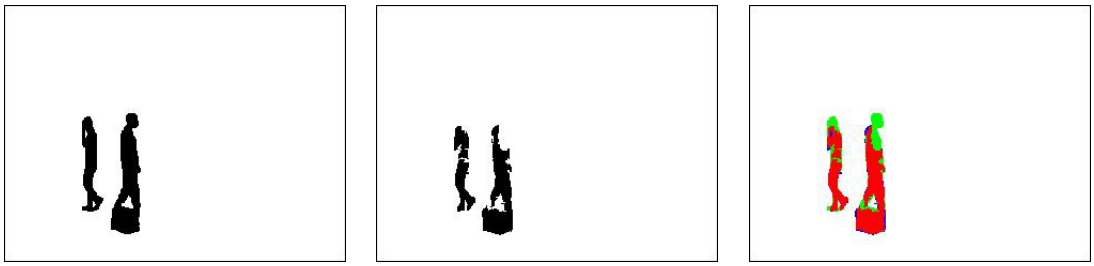
(b) Video 2

FIGURE 6. Comparative experiment. First row: input images, second row: foreground images obtained from GMM, third row: foreground images obtained from nonparametric kernel density estimation, fourth row: the result of obstacles detection obtained from the proposed method, fifth row: the result of obstacles detection in which nonparametric kernel density estimation is employed for the background modeling.

image using the result of the region classification. In Figure 5, the fourth row shows the results of obstacles detection. In these resultant images, the obstacles are represented by shape.

Figure 6 shows the results of comparative experiment. In comparative experiment, we reconstruct the background model using nonparametric kernel density estimation [16]. The second row and the third row of Figure 6 show the foreground images obtained from two different background modeling methods. The fourth row of Figure 6 shows the results of obstacles detection using the proposed method. The fifth row shows the results of obstacles detection of comparative experiment.

8. **Results.** In order to evaluate the effectiveness of the proposed obstacles detection method, we compare the result of obstacles detection (shown in Figure 7(b)) with the Ground Truth (shown in Figure 7(a)). In the resultant image of comparison (shown in



(a) Ground truth (b) Obstacles (c) Resultant image

FIGURE 7. Images employed for evaluation

TABLE 1. The results of evaluation

Videos	Evaluation Values			
	The Proposed Method		Comparative Experiment	
	Precision	Recall	Precision	Recall
Video 1	92.9%	80.0%	89.3%	75.5%
Video 2	94.9%	74.6%	91.9%	62.0%

Figure 7(c)), the red area means the overlap part of (a) and (b), this part is called *True Positive*; Blue means the part which is included in (b) but not in (a) and this part is called *False Positive*; Green means the part which is included in (a) but not in (b) and this part is called *False Negative*. We calculate '*recall*' using the following formula:

$$recall = \frac{TP}{GT} \times 100[\%] \tag{9}$$

where TP is the number of pixels in the True Positive area; GT is the number of black pixels in the Ground Truth image. If '*recall*' is larger than 0.5, we consider this object has been extracted. Then we calculate '*Recall*' and '*Precision*' using the following formulas:

$$Recall = \frac{N_{TP}}{N_{GT}} \times 100[\%] \tag{10}$$

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \times 100[\%] \tag{11}$$

where N_{TP} is the number of correct objects in the resultant images; N_{TP} is the number of objects in the GT images; N_{FP} is the number of incorrect objects in the resultant images. '*Precision*' can be seen as a measure of exactness or fidelity, whereas '*Recall*' is a measure of completeness. The results of evaluation are shown in Table 1.

9. **Discussion and Conclusion.** In this paper, we proposed an obstacles detection method using a video taken by a vehicle-mounted monocular camera.

The performance of the proposed method depends on the size of an obstacle in an image or the distance between the obstacle and the car. If the size of an obstacle in an image is too small, it is recognized as noise and deleted. According to the performed experiments, the maximum feasible distance of detecting a pedestrian is about 70m.

The distance that a car moves during the period from the detection system starts detecting obstacles until the car stops is defined as a stop distance. When this stop distance is larger than the maximum feasible distance, the obstacle detection makes no sense as the car crashes against the obstacle. We calculate the stop distance using the

following formula:

$$d = vt_f + vt_r + d_v \tag{12}$$

where v is the speed of a car; t_f is the processing time of each frame; t_r is the driver reaction time; d_v is the distance needed to stop the car from the instance of brake application begins. The term vt_f means the distance that the car moves during the processing time of the detection; vt_r is the distance that the car moves during the driver reaction time. The processing time of the detection method t_f is 695ms/f. The parameters t_r and d_v can be found in [17]. Then the stop distance can be calculated and the results are shown in Table 2.

TABLE 2. Stop distance

Speed of a car (km/h)	20	30	40	50	60
Stop distance (m)	24	41	58	75	97

Because the stop distance must be smaller than the maximum feasible distance, the proposed method can be applied to a vehicle driving up to 40km/h. This obstacles detection method can be applied to the real driving condition of a vehicle in the city.

In Section 3, we applied a GMM in the moving camera scene. The GMM is an effective background modeling method which was often used in the static camera scene. However, we expanded so that it can be applied to a moving camera case. This expansion is important to the industrial applications of a vehicle-mounted camera based obstacle detection system.

In Section 5, the proposed method detected road regions using motion compensation. This detection method is independent of road marking and lanes and can be applied to both marked and unmarked roads. Moreover, this method can not only be used in the road detection, but it can also be used in 2-D and 3-D objects classification, because, when 2-D objects are lying on the road (such as papers), they have the same characteristics with the road plane. We can use this method to classify 2-D objects as the road region and classify 3-D objects as the non-road region. This feature is important to an obstacle detection system, because 2-D objects on the road are not dangerous to driving. We should avoid these 2-D objects being detected as much as possible, because these 2-D objects will raise a false positive.

The proposed method has some advantages over the existing obstacles detection methods. In the first place, the proposed method uses a monocular camera. This realizes an economic system and smaller computation time. It is also advantageous for achieving real-time processing. In the second place, the proposed method can detect arbitrary objects which may pose a threat to safe driving on the road, not just to detect specific objects. In the third place, the proposed method can also detect both static and moving objects simultaneously. To the best of our knowledge, no researches have ever proposed a method which detects both static and moving objects simultaneously. This is helpful because static objects such as boxes fallen on the road from a car are dangerous for drivers. Most of the existent methods concentrate only on detecting moving objects such as pedestrians, bicycles and cars. In the fourth place, the output of the proposed method is the shape of obstacles. Currently most of the existing obstacles detection methods cannot extract the shape of obstacles. They only use a rectangular frame which surrounds an obstacle to represent a detected obstacle. We understand that extraction of the shape of an obstacle is important for obstacle recognition. If the detected obstacle is a pedestrian, we can also use the shape to carry out the recognition of his/her motion. The proposed method is now under improvement so that it may be applicable to a slightly curved road.

Acknowledgment. This study was supported by JSPS KAKENHI Grant Number 25350 477.

REFERENCES

[1] C. Demonceaux and D. Kachi-Akkouche, Robust obstacle detection with monocular vision based on motion analysis, *Proc. of IEEE Intelligent Vehicles Symposium*, pp.527-532, 2004.

[2] W. Kruger, W. Enkelmann and S. Rossle, Real-time estimation and tracking of optical flow vectors for obstacle detection, *Proc. of IEEE Intelligent Vehicles Symposium*, pp.304-309, 1995.

[3] G. Lefaix, E. Marchand and P. Bouthemy, Motion-based obstacle detection and tracking for car driving assistance, *Proc. of Pattern Recognition*, pp.74-77, 2002.

[4] A. Broggi, M. Bertozzi, A. Fascioli and M. Sechi, Shape-based pedestrian detection, *Proc. of IEEE Intelligent Vehicles Symposium*, pp.215-220, 2000.

[5] M. Lutzeler and E. D. Dickmanns, Road recognition with marveye, *Proc. of IEEE Intelligent Vehicles Symposium*, pp.341-346, 1998.

[6] A. Kuehnle, Symmetry-based vehicle location for AHS, *Proc. of SPIE-Transportation Sensors and Controls: Collision Avoidance, Traffic Management, and ITS*, vol.2902, pp.9-27, 1998.

[7] M. Bertozzi and A. Broggi, Gold: A parallel real-time stereo vision system for generic obstacle and lane detection, *Proc. of IEEE Image Processing*, pp.62-81, 1998.

[8] R. Labayrade and D. Aubert, Robust and fast stereovision based obstacles detection for driving safety assistance, *Proc. of Machine Vision Applications*, pp.624-627, 2004.

[9] J. K. Tan, S. Ishikawa, S. Sonoda, M. Miyoshi and T. Morie, Moving objects segmentation at a traffic junction from vehicular vision, *ECTI Transactions on Computer and Information Technology*, vol.5, no.2, pp.73-88, 2011.

[10] C. Stauffer and W. E. L. Grimson, Adaptive back-ground mixture models for real-time tracking, *Proc. of Conference on Computer Vision and Pattern Recognition*, vol.2, pp.246-252, 1995.

[11] C. Harris and M. Stephens, A combined corner and edge detector, *Proc. of the 4th Alvey Vision Conference*, pp.147-151, 1988.

[12] B. D. Lucas and T. Kanade, An iterative image registration technique with an application to stereo vision, *Proc. of International Joint Conference on Artificial Intelligence*, vol.2, pp.674-679, 1981.

[13] M. Fischler and R. Bolles, Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography, *Communications of the ACM*, vol.24, pp.381-395, 1981.

[14] R. Hartley, In defense of the eight-point algorithm, *Proc. of IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.580-593, 1997.

[15] J. C. S. Jacques Jr, C. R. Jung and S. R. Musse, Background subtraction and shadow detection in grayscale video sequences, *Proc. of the 18th Brazilian Symposium on Computer Graphics and Image Processing, SIBGRAPI 2005*, pp.189-196, 2005.

[16] A. Elgammal, R. Duraiswami, D. Harwood and L. S. Davis, Background and foreground modeling using nonparametric kernel density estimation for visual surveillance, *Proc. of the IEEE*, vol.90, pp.1151-1163, 2002.

[17] AASHTO, *A Policy on Geometric Design of Highways and Streets*, 2004.