International Journal of Innovative Computing, Information and Control
Volume **9**, Number **9**, September **2013**
ICIC International ©2013 ISSN 1349-4198
pp. 3561–3572

ROAD REGION ESTIMATION AND OBSTACLES EXTRACTION USING A MONOCULAR 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 August 2012; revised January 2013

Abstract. *An obstacles detection method is proposed which employs a camera mounted on a vehicle. Although various methods of obstacles detection have already been reported, they normally detect only moving obstacles and not static obstacles, and a detected obstacle is represented by a rectangular frame that surrounds this obstacle, which does not provide shape of the obstacle. In this paper, a method is proposed for detecting obstacles on a road, irrespective of moving or static, by use of background modeling and road region detection. The output of the proposed method is the shape of obstacles. 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 extract the road region using SVM. In this road region, we carry out region classification. We can delete all the objects which are not obstacles in the foreground images based on the result of the region classification. In the performed experiments, it is shown that the proposed method is able to extract the shape of both static and moving obstacles when a car is driving.*
Keywords: Obstacle detection, Monocular vision, GMM, Road region detection, SVM

1. **Introduction.** In recent years, autonomous collision avoidance systems have been developed for realizing safe driving to prevent car accidents. These systems should warn the drivers of the presence of obstacles and help them to take action in advance. In these systems, the ability to detect obstacles is essential. We know that safe driving of a car depends heavily on vision. The vision of a driver can be improved by the systems that give information on the environment around the vehicle that cannot be seen or hardly seen by human eyes. Therefore, a vision-based obstacle detection system is the current research focus and the mainstream in intelligent vehicle technology.

The existing obstacles detection methods are separated into three categories [1]: (i) The first method uses a monocular static camera. This method detects obstacles based on the optical flows which are inconsistent with the main movement direction of vehicles [2,3]. This method needs huge calculation and it is sensitive to vehicle motion. It cannot detect static obstacles. It can only be used to detect moving obstacles. (ii) The second method uses a monocular moving camera. This method detects obstacles based on searching for such features as shape [4,5] or symmetry [6]. This method can only be used to detect one kind of specific object, such as pedestrian detection or vehicle detection. (iii) The last method is 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 calculations and is sensible to vehicle motion.

These existing methods 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. 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 order to make up for these inadequacies, 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 analyzes the captured images to realize obstacles detection. The output of the proposed method is the shape of moving as well as static obstacles. After having obtained the obstacle information, drivers can react quickly and make the corresponding actions accurately to prevent car accidents. Here *true obstacles* are defined as arbitrary objects which 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 *false obstacles*.

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.

Outline of the paper is given in the following: Given a video image sequence taken by a moving camera, in Section 3, we employ Gaussian Mixture Model for reconstructing the background. In Section 4, we detect the road region using Support Vector Machines. Section 5 describes region classification using the result of road region detection and extraction of obstacles on the road. Experimental results are shown in Section 6. Finally, the paper is concluded in Section 7.

2. **Outline of the Proposed Method.** 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 and is regarded as the background. In this condition, the foreground image which is obtained from the background model contains obstacles on the road and the objects outside of the road. In order to extract the shape of the obstacles in the foreground image, the following operations are

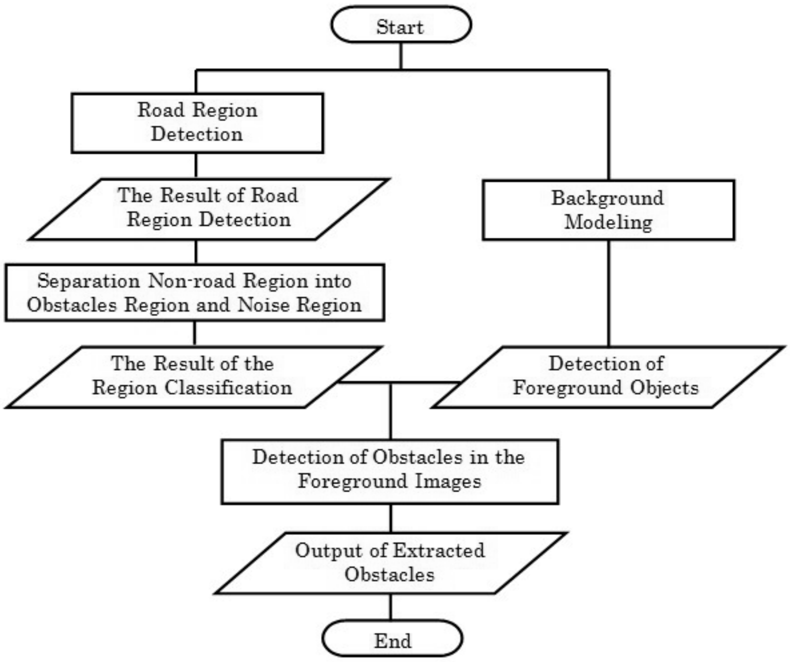


Figure 1. Flowchart of the proposed method

employed. First, the road region is detected using Support Vector Machines. Second, 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. All the objects inside the noise region in the foreground image are considered as noises and deleted. Road marks in the foreground image are considered as false obstacles inside the road region and deleted using the road region. Finally, the shape of the obstacles (e.g., pedestrians, boxes) in the foreground image is extracted using the obstacles region. Figure 1 shows the flowchart of the proposed method.

3. **Background Modeling.** A typical method of detecting obstacles is background modeling, by which we can get the shape of an object directly. Numerous approaches concerning this problem differ in the types of used background models and the procedures 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. Among adaptive methods, Christopher et al. [9] used a single Gaussian distribution to model the values of a particular pixel and to get the background model. When this method is applied in an indoor scene, the output is satisfactory; but not good for outdoor scenes. Rather than modeling the values of one pixel by one Gaussian distribution, Stauffer et al. [10,11] modeled the values of a particular pixel as mixture of K Gaussian distributions. This is method is called Gaussian Mixture Model (GMM). Different Gaussians are assumed to represent different gray values. Based on the mean value and the variance of each Gaussian in the mixture, they determine which Gaussian corresponds to the present background. Pixels that do not fit the background distributions are considered as foreground. To allow the model adapt to change in illumination and run in real-time, an update algorithm was applied.

The GMM is robust when employed in a fixed camera case. However, in this research, the employed camera is moving since it is mounted on a car. In order to employ the GMM in a moving camera case, we need to construct a virtual scene based on a realistic scene. We then 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.

3.2. **Gaussian mixture model.** We consider the values of a particular pixel in the image sequence over time as *pixel process*. The pixel process is a time series of pixel values. At any time t ($t = 1, 2, \ldots, T$), the history of a particular pixel (x_0, y_0) is given by

$$\{X_1, \ldots, X_T\} = \{I(x_0, y_0, t) : 1 \le t \le T\} \tag{1}$$

where I is the gray value of pixel (x_0, y_0).

The recent history of each pixel, $\{X_1, \ldots, X_T\}$, is modeled by a mixture of K Gaussian distributions. The probability of observing the current pixel value X_t is

$$P(X_t) = \sum_{k=1}^{K} w_{k,t} * \eta(X_t, \mu_{k,t}, \sigma_{k,t}^2) \tag{2}$$

where K is the number of Gaussian distributions in the mixture; $w_{k,t}$ is the weight of the k^{th} Gaussian in the mixture at time t; $\mu_{k,t}$ is the mean value of the k^{th} Gaussian in the mixture at time t; $\sigma_{k,t}^2$ is the covariance of the k^{th} Gaussian in the mixture at time t; η is a Gaussian probability density function defined by

$$\eta(X_t, \mu_{k,t}, \sigma_{k,t}^2) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2}\frac{(X_t - \mu_{k,t})^2}{\sigma_{k,t}^2}\right\} \tag{3}$$

The first frame in the video is used for the initialization of the GMM. A new distribution is created for each pixel with the current pixel value as its mean value, an initially high variance, and a low prior weight.

The updating algorithm of GMM goes as follows:

1) Every new pixel value, X_t, is checked against the existing k ($1 \le k \le K$) Gaussian distributions to find if it matches one of those distributions. The match is defined by

$$\frac{|X_t - \mu_{k,t-1}|}{\sigma_{t-1}} \le T_{gauss} \tag{4}$$

Here T_{gauss} is a threshold.

2) The weights of the k^{th} distributions at time t, $w_{k,t}$, are adjusted as follows:

$$w_{k,t} = (1 - \alpha) w_{k,t-1} + \alpha (M_{k,t}) \tag{5}$$

where α is the learning rate and $M_{k,t}$ is 1 for the model which matched and 0 for the remaining models. After this approximation, the weights are renormalized.

3) If none of the K distributions match the current pixel value X_t, the least probable distribution is replaced with a distribution with the current value X_t as its mean value, an initially high variance, and a low prior weight. When $k \leq K$, create a new distribution with the current value X_t as its mean value, an initially high variance, and low prior weight.

4) If there are distributions which match the current pixel value X_t in the K distributions, the Gaussian Model k that matches best with the current pixel value will be updated as follows:

$$\mu_{k,t} = (1-\rho)\mu_{k,t-1} + \rho X_t \tag{6}$$

$$\sigma_{k,t}^2 = (1-\rho)\sigma_{k,t-1}^2 + \rho(X_t - \mu_{k,t})^T(X_t - \mu_{k,t}) \tag{7}$$

where $\rho \approx \alpha / w_{k,t}$. The parameters μ and σ^2 for unmatched distributions remain unchanged.

5) The existing Gaussians are ordered by the value of $w_{k,t}/\sigma_{k,t}^2$. After ordering, the most likely background distributions remain on the top and the less probable transient background distributions on the bottom and are eventually replaced by new distributions. Then the first B distributions are chosen as the background model, where

$$B = \arg\min_b \left(\sum_{k=1}^{b} w_k > T_B \right) \tag{8}$$

where T_B is a measure of the minimum portion of the data that should be accounted for by the background.

6) Every pixel value, X_t, is checked against the B Gaussian distributions; if it matches one of them, this pixel is a background pixel, otherwise a foreground pixel.

4. **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 paper, we detect a road region using the Support Vector Machines (SVM) [12-14]. This method includes two steps: a training step and a test step.

In the first frame of the input video, a small sample of pixels is labeled as a road class or a non-road class. (This is done manually at the moment in experiments.) With each pixel of these samples, we extract a feature vector by feature extraction (described later). These feature vectors are considered as the training data of the SVM.

In other frames of the input video, a feature vector is extracted for each pixel using the same feature extraction method. These feature vectors are considered as the test data. Then, each pixel in the input video is classified as a road pixel or a non-road pixel using the trained SVM classifier.

4.1. **Feature extraction.** In this paper, the features we use are color features and texture features. For color features, three features in HSV color space are used. For texture features, five Haralick statistical features [15] are used as follows:

$$Energy = \sum_u \sum_v \{p(u,v)\}^2 \tag{9}$$

$$Entropy = -\sum_u \sum_v p(u,v) \log\{p(u,v)\} \tag{10}$$

$$Contrast = \sum_{n=0}^{N_g-1} n^2 \left\{ \sum_{u=1}^{N_g} \sum_{v=1}^{N_g} p(u,v) \right\}, \quad |u-v| = n \tag{11}$$

$$Inverse\ difference\ moment = \sum_u \sum_v \frac{1}{1+(u-v)^2} p(u,v) \tag{12}$$

$$Correlation = \frac{\sum_u \sum_v (uv)p(u,v) - \mu_x\mu_y}{\sigma_x\sigma_y} \tag{13}$$

Here $p(u,v)$ is an element of a Gray Level Co-occurrence Matrix (GLCM). μ_x, μ_y and σ_x, σ_y are the mean values and covariance values calculated from GLCM, respectively.

The algorithm of texture features calculation goes as follows:

1) The gray levels of original images are 256. We reduce the number of gray levels from 256 to 8.

2) The pixels in the small square window of size $5 * 5$ centered at the current pixel are used to calculate a GLCM. The size of the GLCM is the same as the number of gray levels of the image, so, in the proposed method, the GLCM is an $8 * 8$ matrix. The definition of the GLCM is given below.

The GLCM is a tabulation of how often different combinations of pixel brightness values (gray levels) occur in an image. GLCM texture considers the relation between two pixels at a time, called the reference and the neighbor pixel. $p_\delta(i,j)$ $(i,j = 0,1,2,\dots,7)$ is the frequency of the reference pixel with the value i and the neighbor pixel with the value j which satisfy a given offset $\delta(D_X, D_Y)$ within the window. $p_\delta(i,j)$ is considered as the value of element (i,j) in GLCM p_δ.

3) According to Equations (9)-(13), we calculate five texture features.

These five texture features and three color features are combined to form an eight-element feature vector as follows:

$$F_{i,j} = [f_{t_1(i,j)}, f_{t_2(i,j)}, f_{t_3(i,j)}, f_{t_4(i,j)}, f_{t_5(i,j)}, f_{c_1(i,j)}, f_{c_2(i,j)}, f_{c_3(i,j)}] \tag{14}$$

where $f_{t_n(i,j)}$ is the n^{th} Haralick statistical feature at the point (i,j) and $f_{c_n(i,j)}$ is the n^{th} color feature at the point (i,j) in the HSV color space.

4.2. **Training database initialization.** The first frame in the input video is used as the training image of an SVM; the other frames in the input video are used as test images.

In the first frame, the training data is selected and labeled by a human. Two rectangle windows are used by a supervisor to select the training data on the image as shown in Figure 2. A lower green window is placed in the road region. The pixels located in this green window are labeled as positive samples. An upper red window is placed outside the road region. The pixels located in this green window are labeled as negative samples. These two samples constitute the training database.

FIGURE 2. Training data selection

5. **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.

5.1. **Region classification.** Here we want to divide the non-road region into obstacles region and noise region. Figure 3(a) and 3(b) show the result of road region detection and the corresponding road region template image, respectively.

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. Figure 3(c) shows the result of the region classification. The obstacle region is indicated by gray pixels.

5.2. **Classification of foreground objects.** 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 noise. In order to extract the shape of obstacles, we should change the black pixels which represent the road marks and noise 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

(a) (b) (c)

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

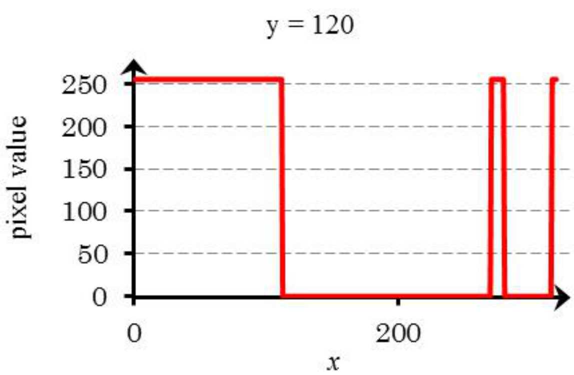


FIGURE 4. Pixel values distribution of the 120^{th} row in the road region image

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. 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 erosion operation and regional expansion as the post-processing.

6. Experimental Results.

6.1. Detection of foreground images. The camera, which is fixed in front of the vice driver's seat, records the road images in front of the car when the car moves forward. 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) are considered as false obstacles and objects outside the road region is noise. According to this definition, correct objects in video 1 is two pedestrians and a box; a correct object in video 2 is a pedestrian; correct objects in video 3 are two pedestrians and a box.

In the first place, we reconstruct the background model using a Gaussian mixture model in the input images. Figure 5 shows the input images and the corresponding foreground images. Because the camera is moving, the foreground images as shown in Figure 5 contain a lot of noise. This noise is mostly caused by the objects outside the road region. In order to delete these noises, we need to detect the road region.

In the second place, we detect the road region in the input images using the Support Vector Machines. Figure 6 shows the results of two road region detection methods: the method using SVM (described in Section 3) and the method using motion compensation [16-19]. In these resultant images, the purple color region means the road region. The detection method using motion compensation needs camera calibration [20] and must calculate the motion parameters of the car. Due to the error of these calculations, the detection results are worse than the detection method using SVM.

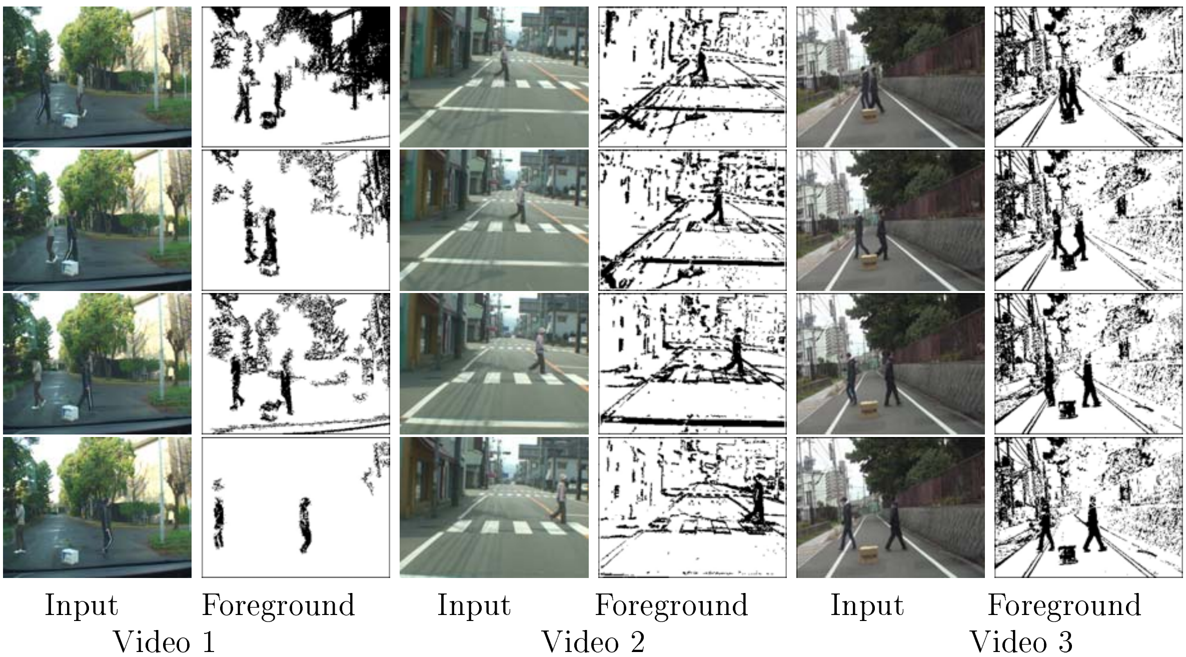
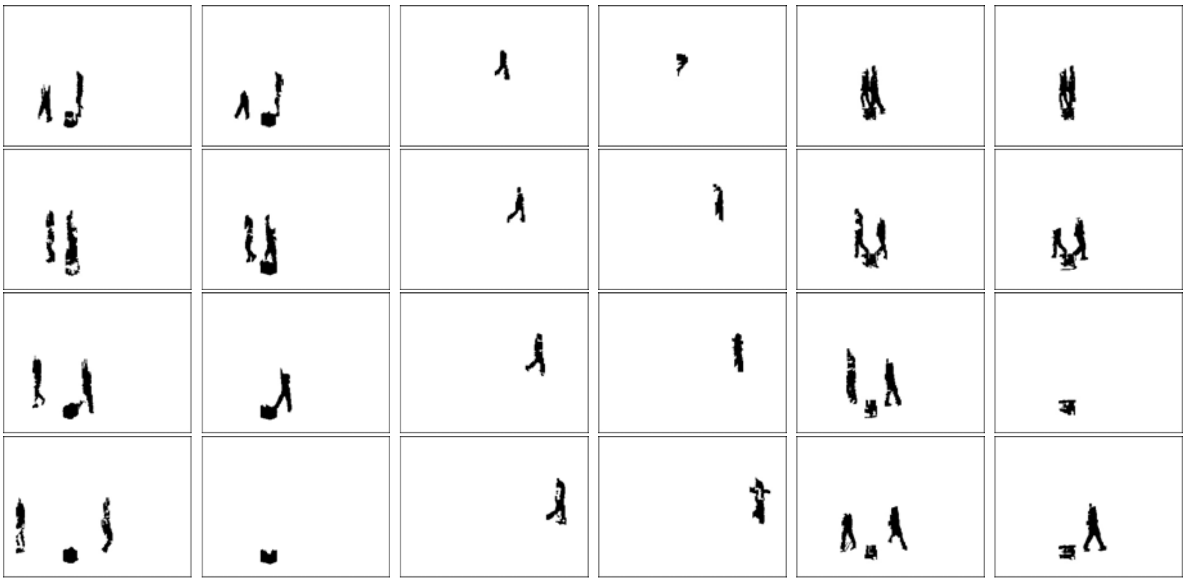


FIGURE 5. The results of background modeling with three scenes

SVM Motion Compensation SVM Motion Compensation SVM Motion Compensation

Video 1 Video 2 Video 3

FIGURE 6. The results of road region detection

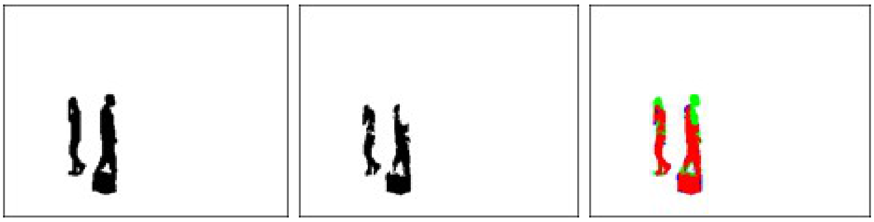


(a) (b) (a) (b) (a) (b)

Video 1 Video 2 Video 3

FIGURE 7. The results of obstacles detection

In the third place, we carry out region classification in the road region template image, and then delete road marks and noises in the foreground image using the result of the region classification. In Figure 7, (a) shows the results of obstacles detection in which SVM is used for the road detection, whereas (b) gives the results of obstacles detection in which motion compensation is employed for the road detection. Since the accuracy of the road region detection has been improved by the employment of the SVM, results



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

FIGURE 8. Images employed for evaluation

of the obstacles detection are improved. It is noted that, in these resultant images, the obstacles are represented by shape.

6.2. **Detection of obstacles.** In order to evaluate the effectiveness of the proposed obstacles detection method, we compare the result of obstacles detection with the Ground Truth (shown in Figure 8(a) and 8(b)). In the resultant image of comparison (shown in Figure 8(c)), the red area means the overlap part of 8(a) and 8(b), and this part is called *True Positive*; blue means the part which is included in 8(b) but not in 8(a), and this part is called *False Positive*; green means the part which is included in 8(a) but not in 8(b), and this part is called *False Negative*. We calculate *recall* using the following formula:

$$recall = \frac{TP}{GT} \times 100[\%] \tag{15}$$

Here 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:

$$Recall = \frac{N_{TP}}{N_{GT}} \times 100[\%] \tag{16}$$

$$Precision = \frac{N_{TP}}{N_{TP} + N_{FP}} \times 100[\%] \tag{17}$$

where N_{TP} is the number of correct objects in the resultant images; N_{GT} is the number of objects in the ground truth images; N_{FP} is the number of incorrect objects in the resultant images. The results of evaluation are shown in Figures 9-11. (a) is the evaluation of the results of obstacles detection in the case of using SVM, whereas (b) is the evaluation of the results of obstacles detection in the case of using motion compensation.

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

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

In the part of road region detection, we compared two road region detection methods. According to the result of the detection, the detection method using the SVM works better than the method using motion compensation. This was shown by the experiments (shown in Figures 9-11). The drawback of the method using motion compensation may have been caused by low precision of camera calibration and motion parameters calculation.

In the road region detection method using the SVM, we extracted a feature vector for each pixel in the input image. This feature vector is combined by five texture features and

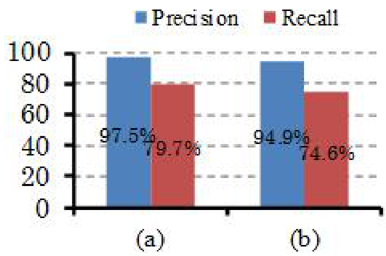


FIGURE 9. The result of evaluation (video 1)

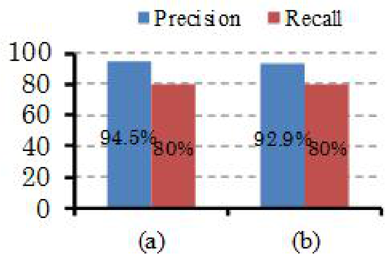


FIGURE 10. The result of evaluation (video 2)

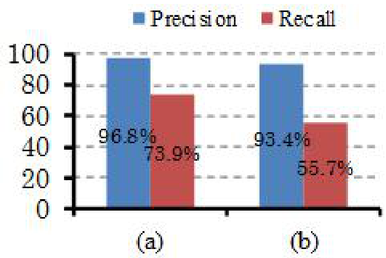


FIGURE 11. The result of evaluation (video 3)

three color features. According to experiments, this composite feature vector contributed sufficiently to detecting a road region from a video. It is noted that the composite feature vector was better than the feature vector which only uses texture features or only uses color features.

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 including both static objects and moving object. 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 third place, the output of the proposed method is the shape of obstacles. Most of the existing obstacles detection methods only indicate the location of an obstacle by a rectangle frame which surrounds it. We understand that extraction of the shape of an obstacle is important for obstacle recognition. If the detected obstacle is recognized as a pedestrian from its shape, we can foresee his/her next action.

The proposed method is now under improvement so that it may be applicable to a slightly curved road.

Acknowledgment. This work was supported by JSPS KAKENHI (22510177) which is greatly acknowledged.

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] W. R. Christopher, A. Azarbayejani, T. Darrell and A. Pentland, Pfinder: Real-time tracking of the human body, *Proc. of IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.2, pp.780-785, 1997.
[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] P. KaewTraKulPong and R. Bowden, An improved adaptive background mixture model for real-time tracking with shadow detection, *Proc. of the 2nd European Workshop on Advanced Video Based Surveillance Systems, Computer Vision and Distributed Processing*, 2001.
[12] S. Zhou, J. Gong, G. Xiong, H. Chen and K. Iagnemma, Road detection using support vector machine based on online learning and evaluation, *Proc. of IEEE Intelligent Vehicles Symposium*, pp.21-24, 2010.
[13] S. Zhou and K. Iagnemma, Self-supervised learning method for unstructured road detection using fuzzy support vector machines, *Proc. of ICRA*, pp.1183-1189, 2010.
[14] W. Hsu, C. Chang and J. Lin, A practical guide to support vector classification, *Proc. of IEEE Intelligent Vehicles Symposium*, vol.1, pp.1-16, 2010.
[15] R. M. Haralick, Statistical and structural approaches to texture, *Proc. of IEEE*, vol.67, pp.786-804, 1979.
[16] K. Yamaguchi, A. Watanabe and T. Naito, Road region estimation using a sequence of monocular images, *Proc. of International Conference on Pattern Recognition*, pp.1-4, 2008.
[17] K. Yamaguchi, T. Kato and Y. Ninomiya, Vehicle ego-motion estimation and moving object detection using a monocular camera, *Proc. of International Conference on Pattern Recognition*, pp.610-613, 2006.
[18] K. Yamaguchi, T. Kato and Y. Ninomiya, Ego-motion estimation using a vehicle mounted monocular camera, *The Institute of Electrical Engineers of Japan*, vol.129, no.12, pp.2213-2220, 2009.
[19] S. Qian, J. K. Tan, H. Kim, S. Ishikawa and T. Morie, Obstacles extraction from a video taken by a moving camera, *Proc. of the International Conference on Connected Vehicles & Expo*, 2012.
[20] Z. Zhang, A flexible new technique for camera calibration, *Proc. of IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp.1330-1334, 2000.