Monocular-Based Drivable Area Segmentation by Fusing 3-D and Texture Information

Takehito Ogata
Center for Technology Innovation-Controls, R&D Group, Hitachi, Ltd., Hitachi, Japan
Email: takehito.ogata.bs@hitachi.com

***Abstract*—In this paper, a monocular camera based drivable area segmentation algorithm is described. The feature-point based motion stereo algorithm is a well-known method to measure 3D environment by using Monocular-camera. However, one of the disadvantages of this algorithm is that it is not suitable to measure 3D information of the area around the traveling direction and non-texture area (e.g. road surface). It is important to know, 3D measurement of the traveling direction and road surface are critical for the driving assistance system. In this paper, we propose the unique drivable area segmentation algorithm. One of its uniqueness is that it combines 3D information of feature points calculated from motion stereo, and segmentation based on similarity of grid-based texture feature. We implement this algorithm in automotive embedded SoC and evaluate various situations.**

***Index Terms*—motion stereo, texture-based feature vector, k-nearest neighbor, occupancy grid map, wide-angle camera**

I. Introduction

It is no doubt that the invention of automobiles has changed our lives more comfortable and convenient. However, it also has brought one of the serious social problems to solve - the traffic accidents. The Advanced Driver Assistance System (ADAS), or the Autonomous Driving (AD) system, is expected to reduce fatalities and injuries caused by traffic accidents by substituting or supporting drivers' role of sense, think and act. One of the essential issues to realize such system is how the system recognizes the surrounding environment. Vision based sensors such as cameras are widely used as front sensing cameras to prevent collision, or rear-view cameras to support drivers' sight while parking, because they are known as low-cost yet high-resolution sensors.

The motion stereo method is one of the vision-based 3D calculation methods and used to detect obstacles around the vehicle. One of the motion stereo methods, which is classical and well-known, consists of two steps. Firstly, the feature point detection method, such as Harris operator, is applied to detect feature points from the images acquired from vehicle-mounted front facing camera. Secondly, the feature point tracking is performed, and trajectories of feature points are used to calculate 3D information. Then, sparse 3D information is obtained. However, one of the disadvantages of this algorithm is that it is not suitable to measure 3D information of the area around traveling direction and non-texture area (e.g. road surface). It is important to know, 3D measurement of the traveling direction and road surface are critical for the driving assistance system.

In this paper, we propose the unique drivable area segmentation algorithm. One of its uniqueness is that it combines 3D information of feature points calculated from motion stereo, and segmentation based on similarity of grid-based texture feature. We implement this algorithm in automotive embedded SoC and evaluate various situations.

II. Related Works

Regarding to the drivable area detection / segmentation algorithms in recent years, vision based Deep Learning approaches are gaining attention [1], [2]. LiDAR based drivable area detection approaches are also widely used for Autonomous Driving applications [3]. However, the cost of these technologies is usually high, because these technologies require high-performance processors or high-cost sensors. One of objectives of this research is to develop affordable algorithm which can be implemented to existing ADAS controllers, so that the algorithm can be used for many ADAS applications.

Compare to above, although dense motion stereo approach [4] is still computationally expensive, sparse motion stereo approach [5] can be implemented in current ADAS controller. However, it is difficult to measure 3D information of the area around traveling direction of the host vehicle since the observed optical flows from the area are relatively short. This is because the change of the image around this area is small since the traveling direction of the host vehicle in the image is almost same as the epipole. This is principle weakness of the algorithm.

III. Proposed Method

As described above, it is not suitable to calculate 3D information from the obstacles around the epipole. In contrast, obstacles far from the epipole, which are located at periphery of the image, are suitable to calculate 3D information. The concept of our approach is to correspond texture feature vector and motion stereo-based 3D information obtained from periphery area and

Manuscript received March 17, 2021; revised July 13, 2021.

doi: 10.18178/joig.9.4.140-145

use it to estimate 3D information around epipole based on texture similarity. Fig. 1 shows the entire process flow of the algorithm and Fig. 2 shows the abstract of the algorithm.

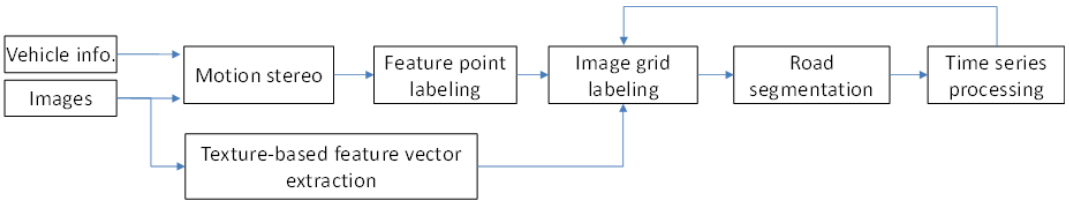


Figure 1. Construction of proposed algorithm.

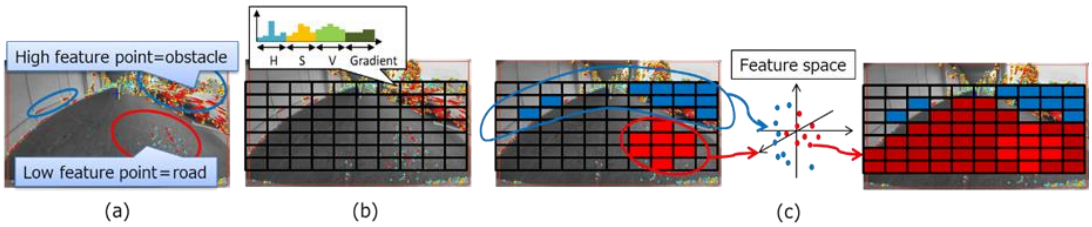


Figure 2. Abstract of proposed algorithm.

A. Feature Point Labeling

Firstly, feature point-based motion stereo method is performed and 3D coordinates $\{X_i, Y_i, Z_i\}$ are calculated for all feature points P_i. Then, each feature point is labelled as "road" or "obstacle" according to its coordinate. This process is shown in Fig. 2(a).

In this paper, since the functions of Harris operator and Lucas-Kanade Optical flow are already available in some ADAS controllers, we adopt them and calculate 3D coordinates by using vehicle's dead reckoning calculated from vehicle's internal information. Then, we use height Z_i to give label C_i^P by following equation. Here, Z_{road} is the predefined threshold of height.

$$C_i^P = \begin{cases} 1 & \text{if } Z_i < Z_{road} \\ -1 & \text{otherwise} \end{cases} \quad (1)$$

B. Texture-Based Feature Vector Extraction

As shown in Fig. 2(b), obtained images are divided into several image grids $\{G_1, G_2, \dots, G_N\}$ and the texture-based feature vectors $\{\tau_1, \tau_2, \dots, \tau_N\}$ are calculated from each image grid.

In this paper, we adopt histograms of HSV and oriented gradients as a feature vector. Each histogram is calculated within each image gird and normalized, then four histograms are connected into one vector. Here, when HSV value in the position of x in the image is represented as $H(x), S(x), V(x)$ (the value range is [0.0,1.0]), HSV histograms in the image grid G_n, which are represented as $h_n = \{h_n^{(1)}, \dots, h_n^{(B_h)}\}$, $s_n = \{s_n^{(1)}, \dots, s_n^{(B_s)}\}$ and $v_n = \{v_n^{(1)}, \dots, v_n^{(B_v)}\}$, are calculated from following equations.

$$h_n^{(u)} = \frac{1}{\text{area}(G_n)} \sum_{x \in G_n} \delta[\text{rnd}(B_h H(x)) - u] \quad (2)$$

$$s_n^{(u)} = \frac{1}{\text{area}(G_n)} \sum_{x \in G_n} \delta[\text{rnd}(B_s S(x)) - u] \quad (3)$$

$$v_n^{(u)} = \frac{1}{\text{area}(G_n)} \sum_{x \in G_n} \delta[\text{rnd}(B_v V(x)) - u] \quad (4)$$

here, $\text{area}(G_n)$ means area of image grid G_n, rnd() means the round function, $\delta[]$ means Kronecker Delta function and B_h, B_s, B_v mean each bin size of HSV histogram, respectively. Likewise, when gradient value and angle in the position of x in the image are represented as $\text{M}(x), \Theta(x)$ (the value range [0.0, 1.0]), the histograms of oriented gradients are obtained from following equation.

$$\theta_n^{(u)} = \frac{1}{\sum_{x \in G_n} M(x)} \sum_{x \in G_n} M(x) \delta\left[\text{rnd}\left(B_g(\Theta(x) - \beta_n)\right) - u\right] \quad (5)$$

here, β_n is vertical angle correction value. Since we use wide-angle camera, vertical angle in the image vary according to the position in the image. Therefore, each β_n is set to each image grid G_n so that calculated gradient angle $\Theta(x)$ of vertical objects is corrected as 0 [deg].

These four histograms are connected into one vector, called feature vector τ_n.

$$\tau_n = \{h_n, s_n, v_n, \theta_n\} \quad (6)$$

here, we do not normalize feature vector τ_n since each histogram is already normalized.

C. Image Grid Labeling

According to the obtained label C_i^P of the feature point P_i at position x^P in the image, the tentative labels $\{\Lambda_0^G, \Lambda_1^G, \dots, \Lambda_N^G\}$ are given to the image grids $\{G_1, G_2, \dots, G_N\}$ by following equation.

$$\Lambda_n^G = \begin{cases} C_i^P & \text{if } x^P \in G_n \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

here, if the image grid includes both feature points of road labeled and obstacle labeled, the tentative label of the grid is set to 0. Then, the formal label C_n^G is given according to the tentative label Λ_n^G. C_n^G is set to Λ_n^G, if Λ_n^G is not 0. If Λ_n^G is 0, C_n^G is given based on the similarity of correspond feature vector τ_n and set of feature vectors τ_m whose Λ_m^G is not 0. An example of this process is shown in Fig. 2(c).

In this paper, since dimensions of feature vector is about just hundreds and number of feature vectors are also about just hundreds, we adopt k-nearest neighbor method for this process. Therefore, a set of k image grids G_{knn} is given by similarity calculation between the feature vector τ_n and all $\tau_m(\Lambda_m^G \neq 0)$. Then, the formal label C_n^G is given by following equation.

$$C_n^G = \begin{cases} \Lambda_n^G & \text{if } \Lambda_n^G \neq 0 \\ \underset{l\in\{1,-1\}}{\operatorname{argmax}} \sum_{n\in G_{knn}} \delta[l - \Lambda_n^G] & \text{otherwise} \end{cases} \quad (8)$$

here, as described in Section I, very few feature points are usually detected from road surface. Therefore, we also set tentative label Λ_n^G to 1, if the image grid G_n is inside of the previous segmentation result which is calculated from the time series processing. The detail of the time series processing is described later section. However, the image grid which satisfy above condition but also include any feature point whose label is -1, the tentative label of this image grid is set to undefined ($\Lambda_n^G = 0$).

D. Road Area Segmentation

According to the obtained results from above process, road area segmentation is performed. Since the image grids located at the most bottom row are mostly classified as road, the segmentation process starts from the most bottom image grids. It searches upper grid and checks the label of the image grid. The search stops if the upper grid is classified as obstacle. This process is performed for each column of the image grid.

Since border of the image grid defines the resolution of the road, it is desirable to set size of the image grid to smaller. However, if we set image grid smaller, it will increase processing time because the processing time of the k-nearest neighbor method is related to the number of image grids. Therefore, we adopt the Coarse to Fine approach. Firstly, normal image grids are used, and the road area is segmented according to above process. Then, smaller image grids are set only around the border of the road and obstacle grids. Then the smaller image grids are classified based on the feature vectors of smaller image grids, and these results are applied to the road area segmentation to make the border smoother.

E. Time Series Processing

As described above, resolution of the road segmentation is related to the size of image grids. Although we adopt Coarse to Fine approach, farther area gets lower resolution. Therefore, we adopt world coordinate-based Occupancy Grid Map (OGM) [6] to reduce this disadvantage. This time series processing will also benefit to reduce affects from the temporal miss segmentation.

Image coordinate-based segmentation result is converted into world coordinate-based results and accumulated to the OGM. We define the OGM in the vehicle centered 2D coordinate system. The map alignment between current segmentation results and the previous OGM is performed by using the dead reckoning which is calculated previously in Section III.A.

An example of calculation is shown in Fig. 3. Red region shown in Fig. 3(a) is the segmentation results obtained from the process described in Section III.D. This road segmentation result is converted to world coordinate system by using camera intrinsic and extrinsic parameters and accumulated to the OGM (Fig. 3(b)). Then, high-probability area is extracted from the OGM (Fig. 3(c)). The back-projected road segmentation result obtained from the OGM is shown in Fig. 3(d). Compare to Fig. 3(a), the border of the road segmentation looks smoother.

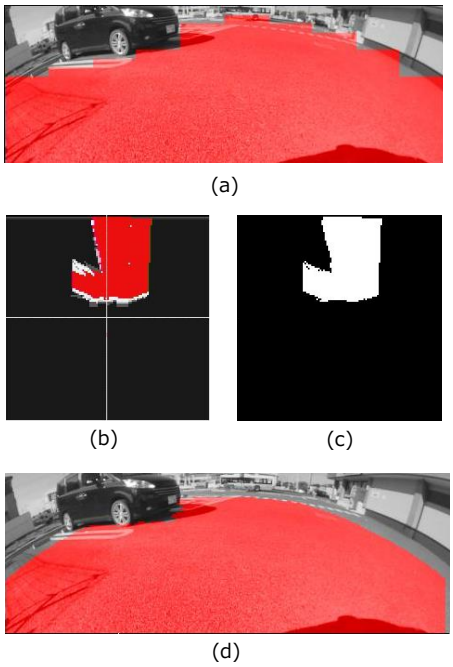


Figure 3. Summary of occupancy grid map.

IV. Principle Confirmation of the Algorithm

The proposed algorithm is valid if the similarity of the texture feature vectors obtained from road area is higher than other area (e.g. obstacle). We perform principle confirmation for this hypothesis.

We apply Principal Component Analysis (PCA) to the texture feature vectors extracted from road area and obstacle area and visualize them in 2D map. Fig. 4 shows results. The left side of the figure shows 2D map of PCA. Red points are the texture feature vectors of road area, whereas blue points are the texture feature vectors of obstacle area. The right side of the figure shows the image used to calculate the texture feature vectors.

In the case of upper image in Fig. 4 (the road area has weak texture), the distribution of the road texture feature vectors is small. It means the within-class variance is small. This is suitable for classification. In contrast, in the case of lower image in Fig. 4 (the road area has strong texture) the distribution of the road texture feature vectors is bigger. It means the within-class variance is bigger. However, since there are lots of texture in the road surface, many feature points of the road area are extracted compare to the upper case. Since we adopt k-nearest

neighbor method, this situation is also suitable for proposed method.

According to above confirmation, our proposed method is valid under following situations.

1) The road area has weak texture (the within-class variance is small).
2) The road area has strong texture and many feature points (the within-class variance is big however many feature points are extracted).

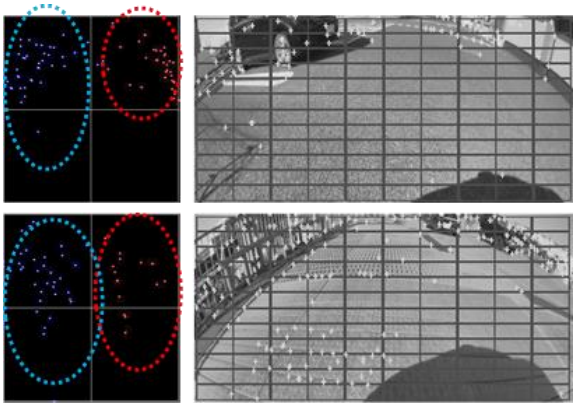


Figure 4. Examples of visualization of texture vector distributions.

V. Implementation

We implemented the proposed method to the camera-based parking assist system shown in Fig. 5. This system has four wide angle cameras and the proposed method is implemented to use whether front or rear camera according to driving direction. It uses the front camera when the host vehicle moves forward whereas it uses the rear camera when the host vehicle moves backward. We implemented to R-Car V2H from Renesas Electronics, Ltd.

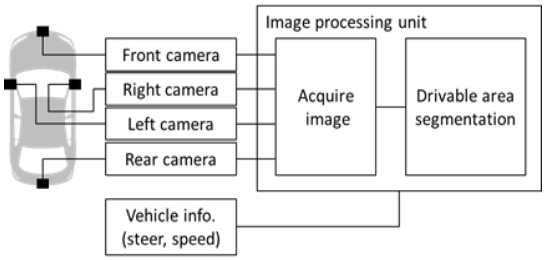


Figure 5. System implementation.

VI. Experiment

A. Basic Evaluation

Firstly, we evaluated the algorithm in the simple situation. The situation is that the host vehicle is approaching to the parked vehicles (Fig. 6). The results are shown in Fig. 7. In Fig. 7, the red area is the projected result of the segmented road area. Although the motion stereo algorithm did not detect the vehicle in front of the host vehicle at all time, the proposed algorithm could segment drivable area according to the distance between the host vehicle and the parked vehicles.

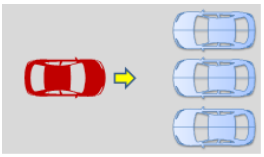


Figure 6. A scene of basic experiment.

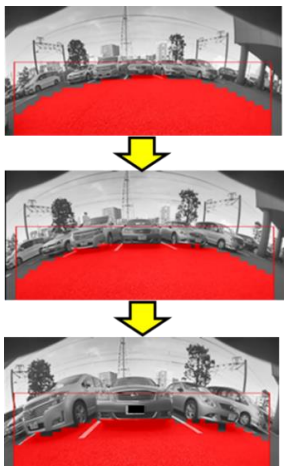


Figure 7. Examples of basic experiment result.

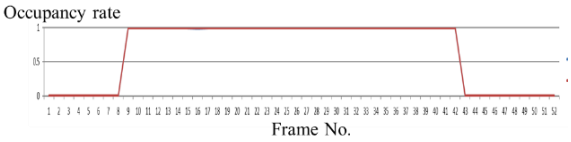
To evaluate this results in detail, we labelled road area of each frame image manually as a ground truth and compared ground truth and the segmentation results. Here, if we just compare the whole segmented road in the image, it is not good evaluation because farther area (usually poor result tend to be obtained) is smaller, whereas closer area (good result tend to be obtained) is bigger in the image. Therefore, we divide detection are into 4 evaluation areas (A, B, C and D, shown in Fig. 8), and compared in each evaluation area individually. For evaluation, we use occupancy rate, which is calculated from number of pixels of segmented road area divided by defined evaluation area. Fig. 9 shows the results. (a), (b), (c) and (d) are the results of area A, B, C and D, respectively. The vertical axis means occupancy rate and the horizontal axis means time frame. Blue line means occupancy rate of the manual result and red line means occupancy rate of the system output. According to this result, although some differences are observed, the system output almost follows the change of manual result. Difference shown in area D is bigger than difference shown in area A. We consider this is cause by the size difference between the area A and D. The size of the image grid in area D is relatively bigger than the size of the image grid in area A.

We measured processing time of the whole algorithm (exclude motion stereo part) shown in Fig. 1. The processing time was about 25-30[ms], which is practical processing time for several ADAS applications.

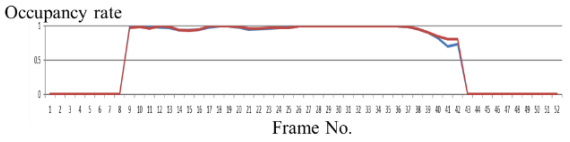
Note that the embedded processor for ADAS is not enough to process conventional DNN-based drivable area segmentation algorithms. Although many DNN-based algorithms work with practical processing time by using high-specification PCs with many GPU resources which are often used for Autonomous Driving developments, this system will contribute only few number of car

accident reduction at far future. It is thought that the algorithm that works in the current embedded processor is more valuable because it will contribute large numbers of car accident reduction by implementing the algorithm to many vehicles.

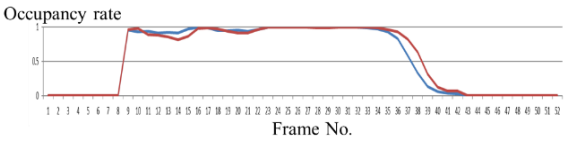
Figure 8. Definition of performance evaluation area.



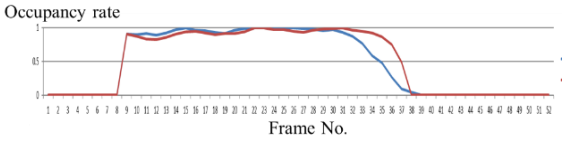
(a) Area A



(b) Area B



(c) Area C



(d) Area D

Figure 9. Evaluation results.

B. Experiment in Various Situations

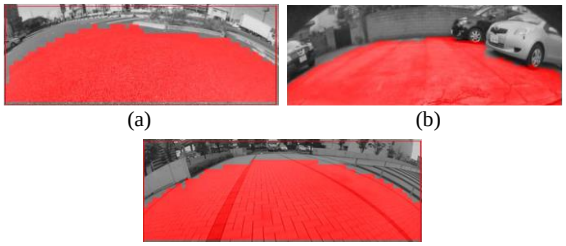
We have tested the proposed method to various scenes. Results are shown in Fig. 10, Fig. 11 and Fig. 12. The proposed method worked well as we expect in the scene shown in Fig. 10. As described in Section IV, the proposed method works effectively if the road has (1) weak texture or (2) strong texture with many feature points. Fig. 10(a) is a simple asphalt road so it belongs to (1). On the other hand, (b) and (c) belong to (2), since (b) is a concrete road with many cracks and (c) is a brick road scene. Fig. 11 shows the scene with shadows on the road. As shown left side of Fig. 11, when the host vehicle is far from the shadow, the proposed method cannot segment shade area as a road. However, once the host vehicle got closer to the shadow area, it detects several feature points in the shadow area, accordingly the proposed method can segment shadow area as a road.

In contrast, as we expect, we got unstable results in night scene or rainy scene. They are shown in Fig. 12(a) and (b). It is well known that the contrast in the image is relatively low in the night. It is also well known that the road reflects the texture of obstacles in rainy situation. These make the texture similarity higher between road and obstacles. Fig. 12(c) and (d) are the 2D map of (a) and (b), calculated by PCA, whose calculation is same as described in Section IV. The meaning of color is same as Fig. 4. The distances between road (red points) and texture (blue points) shown in Fig. 12(c) and Fig. 12(d) are closer than the distance shown in Fig. 4. Therefore, the proposed method does not work well when feature vectors calculated from the road texture and obstacle texture are similar.

C. Calculation of Reliability

Generally, ADAS or AD applications should work in various situations. In order to realize this, it is not usual to develop a single recognition algorithm which performs perfectly in all situations. Instead of do so, the commonly used approach is to fuse results of several algorithms which have individual strengths and weaknesses. To fuse their results better, the calculation of the reliability or confidence of recognition result is important. Therefore, we considered reliability calculation for the proposed method.

Since the proposed algorithm calculates the feature vectors which have tentative label $\{\Lambda_0^G, \Lambda_1^G, \ldots, \Lambda_N^G\}$ during the calculation, it is thought that their between-class variance and within-class variance can be the candidate of the reliability. To confirm this, we performed the LDA to the feature vectors with tentative labels and calculated the between-class variance and within-class variance of road class in the first principal dimension of the LDA. Fig. 13 shows the results. Fig. 13(a), (b) and (c) are the same scenes as Fig. 10(a), Fig. 10(c) and Fig. 12(a), respectively. The vertical axis is variance and the horizontal axis is time frame. The red line means between-class variance and the blue line means within-class variance of the road class. According to these results, within-class variance of the road class does not have useful information. In contrast, between-class variance has useful because it becomes higher when the segmentation results are good whereas it becomes lower when the segmentation results are bad. Therefore, we found that the between-class variance can be the appropriate indicator of the reliability of the algorithm.



(a) (b)

(c)

Figure 10. Segmentation results (1).

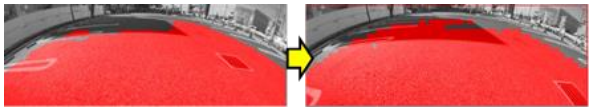
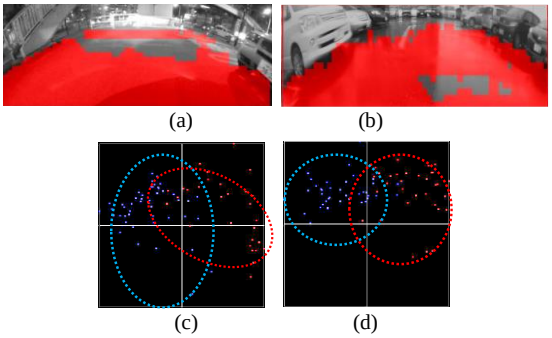
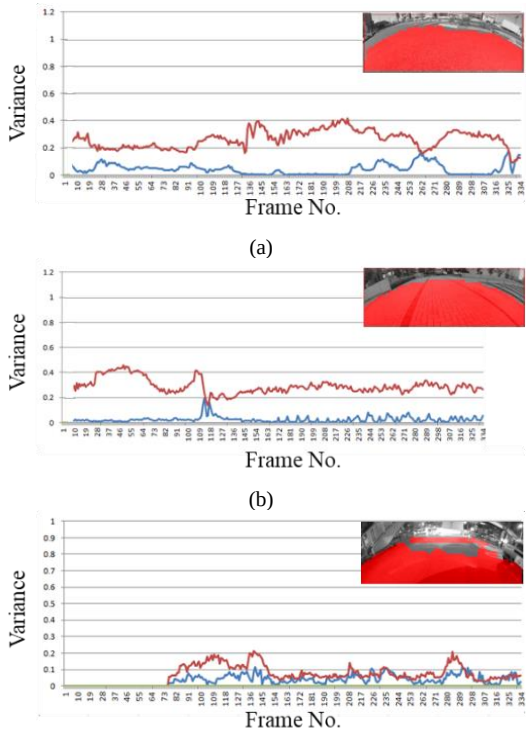


Figure 11. Segmentation results (2).



(a) (b) (c) (d)

Figure 12. Segmentation results (3).



(a) (b) (c)

Figure 13. Calculation results of between-class variances and within-class variances.

VII. Conclusion

In this paper, we proposed a unique monocular camera based drivable area segmentation method. One of its uniqueness is that it combines 3D information of feature points calculated from motion stereo, and segmentation based on similarity of grid-based texture feature. Especially, it uses machine learning approach for grid labeling however it does not require pre-training or pre-defined training data. It uses labels defined by 3D information of motion stereo as pre-defined data and classify unknown feature vectors by using them. The contribution of this paper is following:

- We developed the affordable drivable area segmentation method which uses monocular camera and works in embedded processor.
- We implemented and evaluated the proposed method and clarified its strength scene and weakness scene. The proposed method is suitable when the road has (1) weak texture or (2) strong texture with many feature points.
- We found the between-class variance of tentatively labeled feature vectors can be the appropriate indicator of the reliability of the algorithm. This is usable when we consider multi-sensor or multi-algorithm fusion to improve robustness of the sensing.

Conflict of Interest

The authors declare no conflict of interest.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. on PAMI*, vol. 39, pp. 2481-2495, Dec. 2017.

[2] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. of CVPR*, 2017.

[3] J. Byun, K. Na, B. Seo, and M. Roh, "Drivable road detection with 3D point clouds based on the MRF for intelligent vehicle," in *Proc. of Conf. on Field and Service Robotics*, 2015, pp. 49-60.

[4] J. K. Suhr, H. G. Jung, K. Bae, and J. Kim, "Automatic free parking space detection by using motion stereo-based 3D reconstruction," *Machine Vision Applications*, vol. 21, pp. 163-176, Feb. 2010.

[5] A. Seki and R. Okada, "Monocular-camera based obstacle detection with measurement error estimation," in *Proc. 20th ITS World Congress*, 2013.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT Press, 2005.

Takehito Ogata was born on January 20th, 1981. He obtained B.E. M.E. and Ph.D. in the field of Control Engineering from Kyushu Institute of Technology, Kitakyushu, Japan, at 2003, 2004 and 2007, respectively. From 2005 to 2006, he was a visiting researcher at the Centre for Vision, Speech and Signal Processing in the University of Surrey, Guildford, England.

He is currently a Senior Researcher and Unit Leader of the Transportation System Control Research Department, Center for Technology Innovation – Controls, R&D Group, Hitachi, Ltd. He has started his career in Hitachi since 2007, and he has been working on several projects for the development of Advanced Driver Assistance System, Autonomous Parking System and Autonomous Driving System. His research interests include computer vision, sensor fusion, machine learning.