Proceedings of the 2018 IEEE
International Conference on Robotics and Biomimetics
December 12-15, 2018, Kuala Lumpur, Malaysia

Road Boundary Estimation for Mobile Robot using Deep Learning and Particle Filter

Kazuki Mano[1], Hiroaki Masuzawa[2], Jun Miura[3] and Igi Ardiyanto[4]

Abstract— **This research aims to develop a method of estimating road boundaries by deep learning. Existing methods detect boundaries using specifically designed features, and if such features are not available, it is difficult to estimate road boundaries. On the other hand, estimation by deep learning does not require designing features beforehand because it can learn features by itself, and it could estimate boundaries for a more diverse set of roads. In this research, we propose a method of estimating road boundaries by a combination of deep learning and particle filter. By performing a temporal filtering with a particle filter, it is possible to deal with occasional failures in road boundary recognition by deep learning.**

I. Introduction

Outdoor mobile robots need to have a function of estimating the boundary of a road or a traversable region for safe navigation. There are many features that can be used for road boundary estimation, but the features that are available in a certain environment may not be available in different environments. Therefore, it is undesirable to estimate road boundaries using only pre-defined specific features.

We have developed methods of estimating road boundaries using multiple visual features with flexible road models and particle filter [1], [2]. The use of multiple features makes it possible to estimate road boundaries robustly in various environments. If it is difficult to obtain any pre-defined features, however, estimation will fail. Therefore, we needed to prepare a vast variety of features in advance.

In this paper, we apply deep learning to solving this problem. Deep learning does not require features in advance, but automatically obtains them by learning from various data. Therefore, a road boundary estimation with deep learning is expected to be able to deal with various types of road boundaries without pre-defined features. However, such an estimation is not always correct, and suffers from temporary failures. In this research, a stable road boundary estimation is realized using temporal information integration of framewise estimation results by particle filter.

Fig. 1 shows the block diagram of the proposed method. First, road boundary regions are detected by semantic segmentation from RGB images using a deep neural network. The states to estimate by the particle filter are road boundary parameters and the output of the network is used as a likelihood.

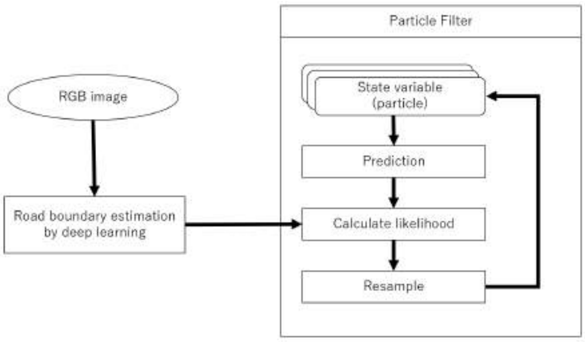

Particle Filter
RGB image
State variable (particle)
Prediction
Road boundary estimation by deep learning
Calculate likelihood
Resample

Fig. 1: Outline of proposed method.

The rest of the paper is organized as follows. Section II describes related work. Section III describes road boundary estimation by semantic segmentation. Section IV describes temporal information integration using particle filter for road parameter estimation. Section V describes experiments of the proposed method. Section VI describes conclusions and future work.

II. Related work

A. Statistical road boundary estimation by multiple features

Matsushita and Miura [1] extract three types of road boundary features, height difference from a laser range finder and intensity and color gradient from a camera, and estimate road parameters and the ego-motion using particle filter. Chiku and Miura [2] apply this approach to a stereo-based road boundary detection. By combining multiple features, they can robustly estimate road boundaries in various road scenes. In these works, a pre-defined set of features are used.

B. Semantic Segmentation

Semantic segmentation is a task of classifying each pixel into one of the predefined set of classes. FCN, SegNet, and ResNet are popular network models in semantic segmentation using Deep Learning. FCN [3] uses end-to-end convolution networks for semantic segmentation. SegNet [4] is efficiently designed in terms of memory and calculation time by using an encoder-decoder model with a smaller number of trainable parameters. The encoder-decoder model is also used for many networks such as U-NET [7]. ResNet [5] trains deeper networks using learning of a residual

[1] Kazuki Mano is with the Departure of Computer Science and Engineering, Toyohashi University of Technology, Japan mano@aisl.cs.tut.ac.jp

[2] Hiroaki Masuzawa is with the Department of Computer Science and Engineering, Toyohashi University of Technology, Japan masuzawa@aisl.cs.tut.ac.jp

[3] Jun Miura is with the Department of Computer Science and Engineering, Toyohashi University of Technology, Japan jun.miura@tut.jp

[4] Igi Ardiyanto is with the Department of Electrical Engineering and Information Technology, Universitas Gadjah Mada, Indonesia igi@ugm.ac.id

function. ResNet is the basis of many networks such as CASENet [8].

C. *Semantic edge detection using deep learning*

Z. Yu et al. [8] proposed a method of extracting boundaries between semantic regions using deep learning. The method is evaluated using SBD [9], a standard dataset for benchmarking semantic edge detection, and Cityscapse [10], a popular semantic segmentation dataset. However, this method requires a large computing power because the network is based on ResNet101 [5], and could be difficult to apply to mobile robots where real-time processing is indispensable.

III. Road boundary detection by deep learning

We formulate the road boundary detection as a three-class segmentation problem where each pixel is classified into road, road boundary, and others. Fig. 2 shows an input image and the corresponding label image. In Fig. 2(b), red, blue, and, green regions represent road boundary, road, and other regions, respectively.

A. *Network*

We use U-NET [7] structure shown in Fig. 3 as the network. The input is a 256 × 256 RGB image of a road scene, and the output is the result of segmentation into the three classes, represented also by a 256 × 256 RGB image.

B. *Dataset*

We use two datasets. One is ICCV09DATA [6], published in ICCV 2009 and includes 715 data, classified into eight classes: sky, tree, road, grass, water, building, mountain, and foreground object. We divide these labels into road class (the blue region in center of Fig. 4) and the others, and then add a new class between them as road boundaries class. Fig. 4 shows examples of input images, original labeled ones, and generated three-class labeled ones. Among those 715 images, 615 images are randomly chosen as training data, and the remaining 100 images are used as test data.

The other dataset was created from images taken at Toyohashi University of Technology. We took ten images as the robot moves on a road. We used five of them for training and the rest for testing. We manually labeled the images (see Fig. 2 for an example).

TABLE I: Evaluation using real data.

	Recall	Precision	Accuracy	F-measure
Best Recall(Fig.5)	0.793	0.840	0.957	0.816
Average	0.552	0.609	0.955	0.571
Standard Deviation	0.204	0.201	0.024	0.200

Combining the two datasets, we have 620 training data and 105 test data. Furthermore, the data were augmented by trimming, left/right inversion, and two types of gamma value change, and we finally have 22,746 training and 4,200 test data.

C. *Detection Results*

1) Evaluation using real data: We evaluate the method using real data in terms of accuracy, recall, precision, and F-measure. The accuracy is the ratio of correctly-identified pixels. The precision is the ratio of the number of road boundary pixels to that of pixels classified as road boundary. The recall is the ratio of the number of pixels classified as road boundary to that of pixels of road boundary. The F-measure is the harmonic mean of the recall and the precision.

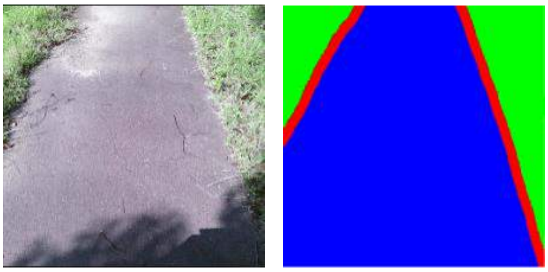
The accuracy focuses on the entire estimation result, and it is not desirable to use it as the evaluation criterion in this problem where most pixels are not road boundaries. Since for autonomous navigation, detecting road boundaries reliably is the most important, we use the recall of the boundary as the primary measure of evaluation.

Fig. 5 shows the result for which the best recall is obtained. Table I summaries the statistics for all test data. These results show that the performance of the method is sufficient to be used for temporal information integration.

2) Evaluation using simulator: We evaluate the method using the Gazebo simulator on ROS [11]. The simulator can build various environments for evaluating the method. Fig. 6 shows the results of road boundary estimation and Table II is the evaluation of these figures. The results show that the proposed method is effected in various environment.

D. *Comparison between 2-class classification and 3-class classification*

We detect road boundary regions based on a 3-class classification (road boundary, road, and others). A 2-class



(a) Input image

(b) Label image

Fig. 2: Training of road boundary estimation by Deep Learning.

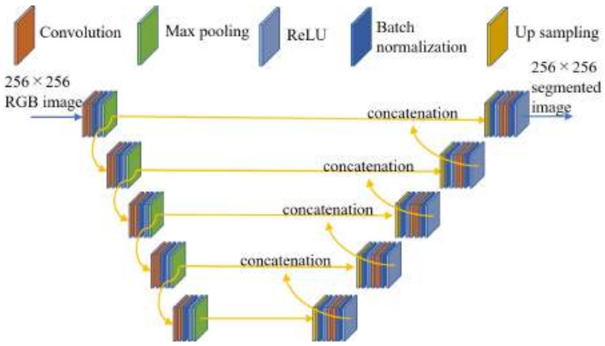


Fig. 3: The structure of U-Net.

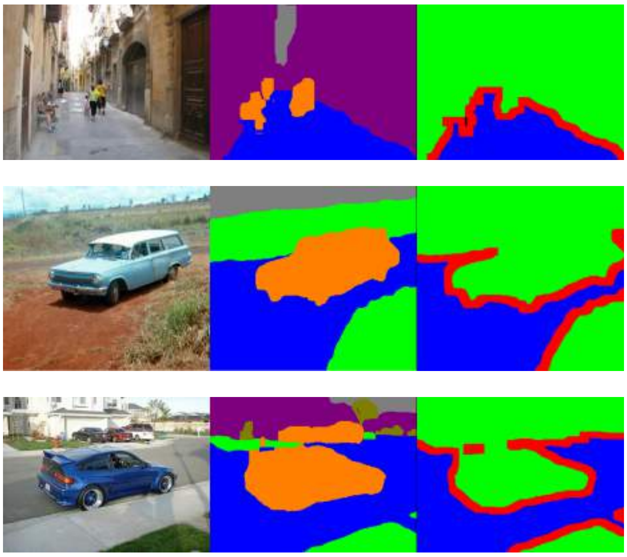
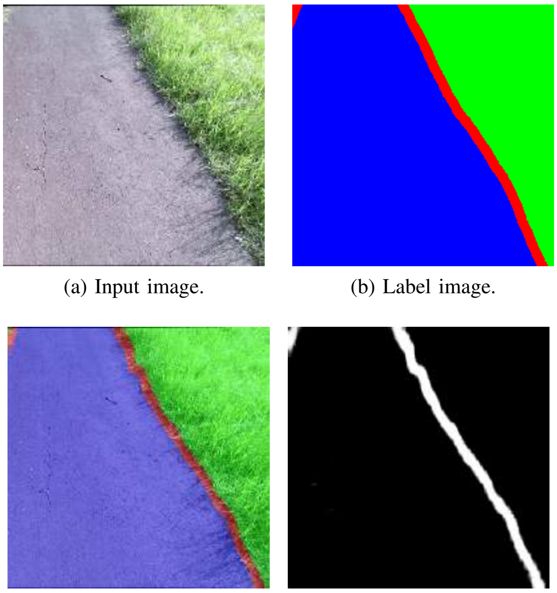


Fig. 4: Data created from ICCV09DATA. Each row shows input image, segmentation result, and three-labeled image from left to right.



(a) Input image.

(b) Label image.

(c) Result of estimation.

(d) Boundary extracted image.

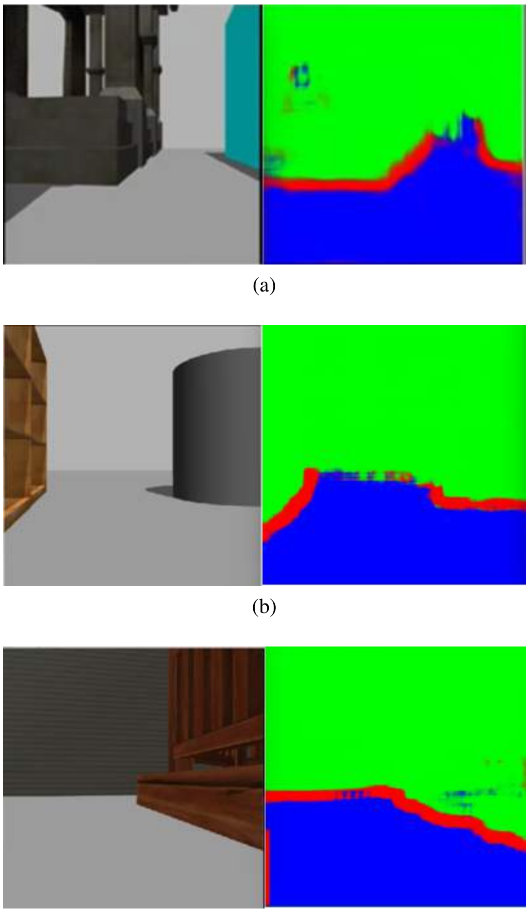
Fig. 5: A classifier result for the real dataset.

classification (road boundary and others) might be, however, enough for our purpose. We therefore compare their performances. Both methods use the same network except the output layers, which is due to the difference in the number of classes.

Table III shows the comparison results in terms of recall, precision, accuracy, and F-measure. Recall, our primary measure, is better in 3-class classification than in 2-class one, probably because 3-class classification utilizes geometrical relationships among three classes, which could be more informative than 2-class ones.

TABLE II: Evaluation in simulator.

	Recall	Precision	Accuracy	F-measure
Fig.6(a)	0.805	0.771	0.981	0.787
Fig.6(b)	0.826	0.686	0.983	0.750
Fig.6(c)	0.704	0.693	0.967	0.698



(a)

(b)

(c)

Fig. 6: Classification results for the simulated dataset. Each row shows input image and segmentation result.

TABLE III: Comparison between 2-class classification and 3-class classification.

		Recall	Precision	Accuracy	F-measure
Average	2-class	0.379	0.675	0.973	0.469
	3-class	0.552	0.609	0.955	0.571
Std. Deviation	2-class	0.185	0.184	0.016	0.192
	3-class	0.204	0.201	0.024	0.200

IV. Temporal Information Integration using Particle Filter

A. *Road model*

There are many types of road, such as an unbranched road and an intersection. In this paper, we deal with the simplest shape, that is, an unbranched road with a right and a left straight boundary. The road model can be represented by four points, namely, the start point and the end point of the right and the left boundary. To cope with the deviation from

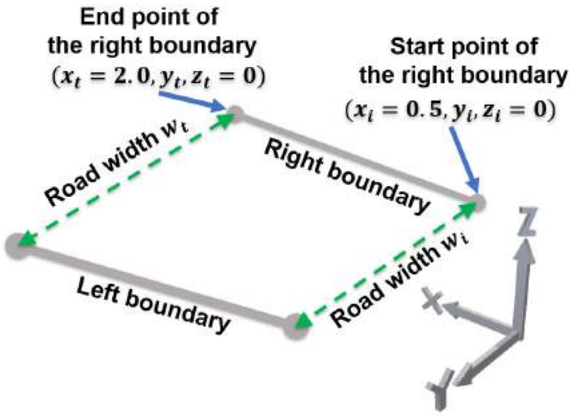


Fig. 7: The road model and state variables.

the road model, such as a loosely curved road, we estimate the positions statistically.

Fig.7 illustrates our road model. Let (x, y, z) be the coordinate value in the robot coordinate system. The x axis is the front direction of the robot, the y axis is the left direction of the robot, and the z axis is the upward direction of the robot. Then, let the state variables be $[y_{ri}, y_{rt}, w_i, w_t]$. y_i, y_t are the y coordinate values of the start and the end point of the right boundary, respectively. w_i and w_t are the road widths at the start point and the end point, respectively. Since the road width never becomes zero or less, we also have a constraint of $w* > 0$. In this model, x and z values are treated as constant and only the y component as a variable.

The position of the start point of the right boundary is $(0.5, y_{ri}, 0)$ and that of the end point is $(1.5, y_{rt}, 0)$. The start and the end point position of the left boundary are expressed as $(0.5, y_{li} = y_{ri} + w_i, 0)$ and $(1.5, y_{lt} = y_{rt} + w_t, 0)$. Note that the unit is meter.

B. Prediction step

There are two causes of uncertainty to be considered in the state transition. One is the uncertainty of robot motion; it does not exactly follow the road. The other is a gradual shape change of the road itself. Both causes simultaneously affect the road boundaries with respect to the robot pose. We therefore add noise to the road boundary parameters $(y_{ri}, y_{rt}, w_i, w_t)$ using a normal distribution with zero mean and a standard deviation of 0.2 [m].

C. likelihood calculation

We project the left and the right road boundary in the scene on the output image of the network, and then examine how the projected boundaries and the semantic segmentation results match for calculating the likelihood. The likelihood is defined by:

$$likelihood = \frac{1}{2}\frac{1}{N}\sum_{i}^{N}\frac{p_i}{255} + \frac{1}{2}\frac{1}{M}\sum_{j}^{M}\frac{p_j}{255}, \qquad (1)$$

where p_i and p_j are the values of the output image for the ith and the jth pixel of the right and the left projected boundary, respectively, and N and M are the respective total number of pixels.

TABLE IV: Execution environment of estimation road boundary by Deep Learning

OS	Ubuntu 16.04
CPU	Intel(R) Core(TM)i7-7700HQ 2.80GHz (8 core)
GPU	NVIDIA GeForce GTX 1050 Ti
DNN framework	Chianer3.3.0

V. Experiment

A. Experiment at Toyohashi University of Technology

The first experiment of estimating road boundary was conducted by using the data taken at Toyohashi University of Technology. The data were acquired using a Kinect v2 put on Mercury Mega robot (Rivest Co., Ltd). We used rosbag, a tool of ROS, for recording and playing back RGB and depth images.

Fig. 8 shows the road boundary estimation results at steps 74, 84, 106, 107 and 245. The road boundary estimation result at each frame is the weighted average of the parameters of all particles. In addition, we compare our method with a one-shot estimation by Hough transform, which is applied to the output of deep learning without any temporal information integration.

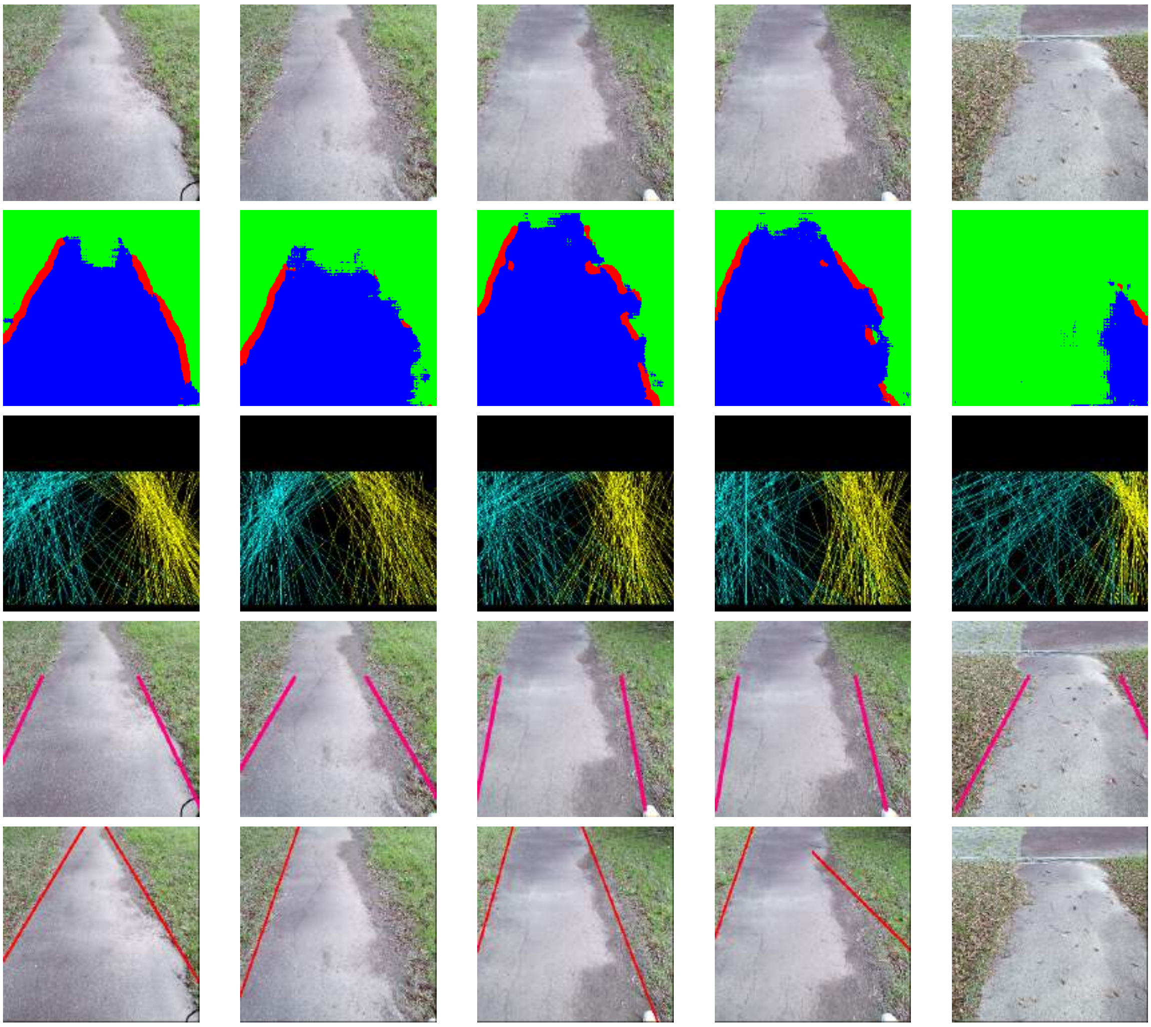
We determine the threshold for votes in Hough transform as one third of the vertical size of the image. The result of road boundary estimation by Hough transform is given by the maximum voted line on the deep learning result.

At step 74, using the likelihood values as weights, the particles converges to some extent and, therefore, the estimation becomes certain. At step 84, the result of one-shot estimation has only left boundary because the right boundary is not be detected by deep learning. However, our method succeeds in estimating it by temporal information integration. At steps 106 and 107, the one-shot estimation is successful at the former but is not at the latter due to the failure of detecting the right boundary by deep learning. At step 245, the deep learning outputs very bad result. Even in these case, our method succeeds thanks to temporal information integration.

The execution time in the environment shown in Table IV was 0.32 sec per cycle with 200 particles and 0.30 sec per cycle with 100 particles.

B. Experiment at Universitas Gadjah Mada

We also conducted experiments at another place, Universitas Gadjah Mada (UGM). For training the network, we added 23 data taken at UGM to the original dataset and applied the data augmentation as well. Fig. 9 shows the results of the road boundary estimation at UGM. The proposed method also works well for different road scenes, while the one-shot detection does not. Especially, at step 99, although the deep learning result produces lots of fake boundary regions, the proposed method keeps a reasonable detection thanks to a model-based filtering.



(a) step74 (b) step84 (c) step106 (d) step107 (e) step245

Fig. 8: Result of the Experiment in Toyohashi University of Technology. From top to bottom of each columns are input image, output of the network, particle filter output, road boundary estimation by using our method and road boundary estimation by using Hough transform.

VI. Conclusions and Future work

In this paper, we have described a road boundary estimation method for a mobile robot using deep learning and particle filter. We use U-Net for boundary extraction and showed that the 3-class classification (road boundary, road, others) is better than the 2-class classification (road boundary, others). To cope with occasional boundary detection failures, we use particle filter for a stable road boundary detection. We define a state vector to represent the road shape and use the output of the trained network for calculating the likelihood values. We applied the proposed method to simple unbranched roads and showed its effectiveness.

We currently deal with only unbranched roads with a right and a left boundary. It is future work to extend the model to a more variety of road shapes such as intersections and acutely-curved roads. It is also necessary to increase the variations of road appearances in the dataset so that the proposed method will be applied to various road scenes.

Acknowledgement

The authors would like to thank Dr. Shuji Oishi for fruitful discussion.

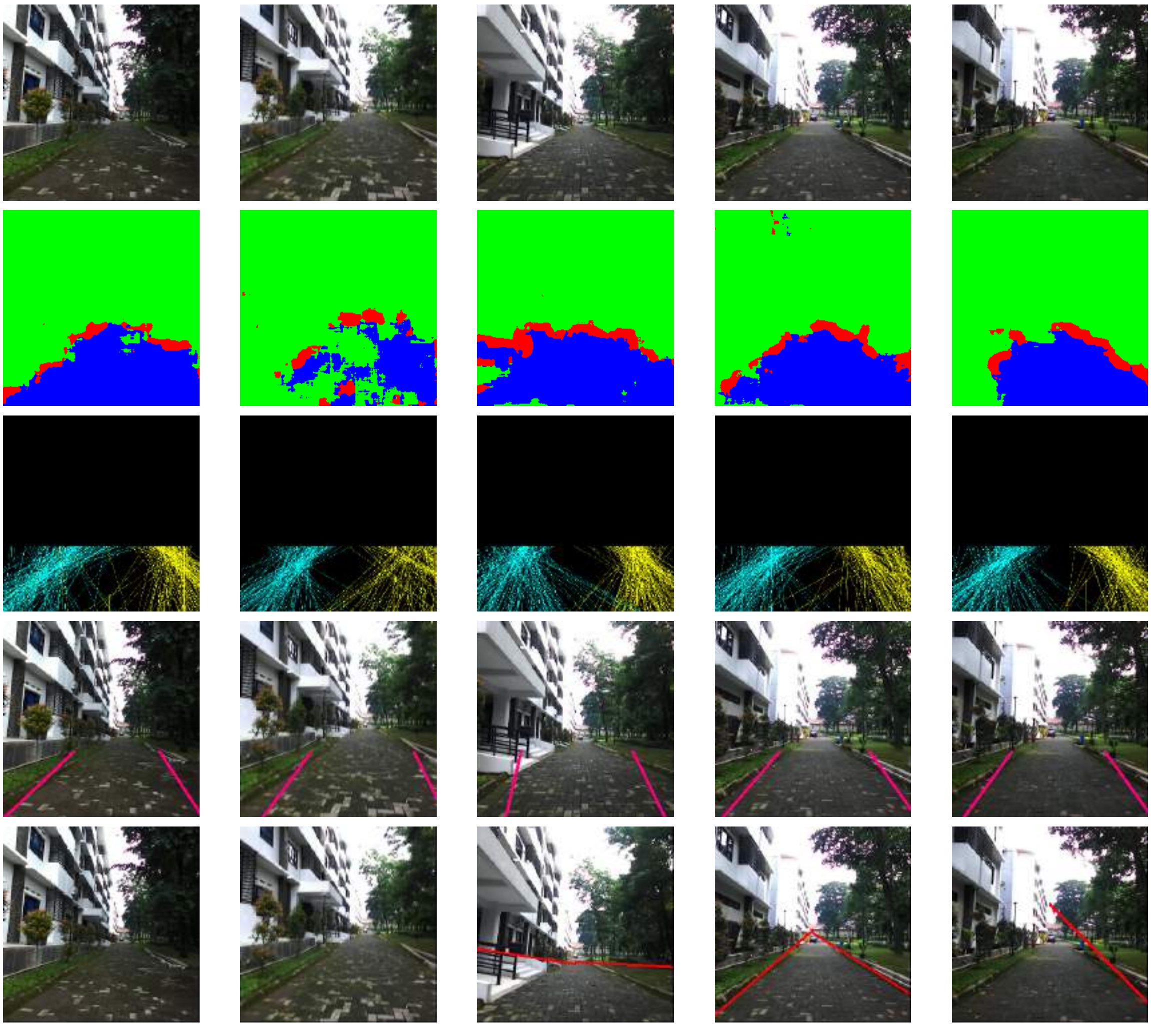
References

[1] Y. Matsushita and J. Miura, "On-Line Road Boundary Modeling with Multiple Sensory Features, Flexible Road Model, and Particle Filter", Robotics and Autonomous Systems, Vol. 59, No. 5, pp. 274-284, 2011.

[2] T. Chiku and J. Miura, "On-line Road Boundary Estimation by Switching Multiple Road Models using Visual Features from a Stereo Camera", IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 4939-4944, 2012.

[3] J. Long, E. Shelhamer and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation", IEEE Conference on Computer Vison and Pattern Recognition, 2015.

[4] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation", IEEE Conference on Computer Vison and Pattern Recognition, 2015.



(a) step28 (b) step65 (c) step99 (d) step206 (e) step215

Fig. 9: Result of the Experiment in UGM. From top to bottom of each columns are input image, output of the network, particle filter output, road boundary estimation by using our method and road boundary estimation by using Hough transform.

[5] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition”, International Conference on Computer Vision, 2016.

[6] S. Gould, R, Fulton and D. Koller, “Decomposing a Scene into Geometric and Semantically Consistent Regions”, IEEE 12th International Conference on Computer Vision, pp. 1-8, 2009.

[7] O. Ronneberger, P. Fischer and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, Lecture Notes in Computer Science, Vol. 9351, pp. 234-241, 2015.

[8] Z. Yu, C. Feng, M. Liu and S. Ramalingem, “CASENet: Deep Category-Aware Semantic Edge Detection”, IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[9] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik, “Semantic contours from inverse detectors”, International Conference on Computer Vision, 2011.

[10] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The Cityscapes Dataset for Semantic Urban Scene Understanding”, IEEE Conferenceon Computer Vision and Pattern Recognition, 2016.

[11] Gazebo. Open Source Robotics Foundation. http://gazebosim.org. [12 Apr. 2018].