

# On-Line Road Boundary Modeling with Multiple Sensory Features, Flexible Road Model, and Particle Filter

Yoshiteru Matsushita      Jun Miura

*Department of Information and Computer Sciences  
Toyohashi University of Technology, Toyohashi, Japan*

**Abstract**— This paper describes a method of robustly modeling road boundaries on-line for autonomous navigation. Since sensory evidence for road boundaries might change from place to place, we cannot depend on a single cue but have to use multiple sensory features. It is also necessary to cope with various road shapes and road type changes. These requirements are naturally met in the proposed particle filter-based method, which makes use of multiple features with the corresponding likelihood functions and keeps multiple road hypotheses as particles. The proposed method has been successfully applied to various road scenes.

**Index Terms**— Outdoor navigation, Mobile robot, Road boundary modeling, Particle filter.

## I. INTRODUCTION

Autonomous outdoor navigation has been one of the active research areas in robotics, from Navlab [22] to Grand Challenge [21]. For a fully autonomous navigation, the robot has to have many functions such as route planning, localization, road detection and following, and obstacle avoidance. This paper focuses on the road (or traversable region) detection.

GPS systems, combined with an accurate map, can provide reliable location information for outdoor navigation (e.g., [15, 19]). But for safe navigation, local information on road boundary, such as curbs and lanes, should be extracted and utilized on-line.

Vision has been widely used for road boundary detection. Some methods detect road and lane boundaries directly [2, 5], while others first detect road regions using, for example, color information to determine the road boundaries [4, 18].

Range sensing is also popular in road boundary detection [3, 10, 23]. If we use a 2D scanner, however, specific geometric features such as guardrails and clear curbs should exist. Using multiple range sensors makes it possible to detect traversable regions by themselves [21]. Stereo vision can also be used for extracting road region as a planar region [14]. Using only geometric information, however, might not be enough in some roads like a small trail among low grasses.

One issue in road boundary detection is how to cope with the variety of road scenes. Effective sensory information for road boundary detection varies from place to place and multiple sensory features thus need to be utilized. Fusion of range and image data has been investigated, but mainly for obstacle detection [11, 12]. Some works use range information for refining the image-based road detection process [2, 8, 21].

Another issue is occasional sensing failures or missing effective features (e.g., a discontinuity of curbs). Road boundary detection only from the latest observation might be vulnerable and, therefore, model-based filtering approaches are effective. Dickmanns and Mysliwetz [5] developed a Kalman filter-based method which estimates the 3D road parameters and the vehicle ego-motion. Apostoloff and Zelinsky [1] proposed a particle filter-based lane detector using vision with a simple road model. Kim [9] proposed a robust lane detection and tracking method based on explicit lane marking detection and particle filtering. The filtering-based approach is also effective for reducing the sensing cost because only a part of sensor data (e.g., some image regions) need to be processed in many cases. Sehestedt et al. [17] applied a particle filter for detecting lane marking in each image not for tracking them.

We have been developing a particle filter-based road boundary detection method [13], similarly to [1]. To cope with various road scenes, the method uses multiple sensory features obtained by cameras and a laser range finder. Evidence from multiple features is naturally integrated via specially-designed likelihood functions. The method dealt with only unbranched roads. In this paper, we extend the method so that it uses more flexible road models with branches. Various road models are adaptively utilized in the particle filter-based framework.

The rest of the paper is organized as follows. Sec. II describes an overview of the proposed method. Sec. III explains the state vector and the road models for unbranched and branching roads. Sec. IV briefly explains the image and the range data processing for the importance weight calculation. Sec. V describes the state transition step in the particle filter which includes robot motion prediction and road model update. Sec. VI shows experimental results in various road scenes. Sec. VII concludes the paper and discusses future work.

## II. OVERVIEW OF THE METHOD

The proposed method adopts a particle filter [20] for integrating multiple sensory information and for managing the road shape and type changes. Fig. 1 shows an overview of the proposed method. The right-hand side of the figure indicates the iteration of particle filter-based estimation. The left-hand side indicates the sensor data processing.

Each particle keeps both the road parameters and the robot

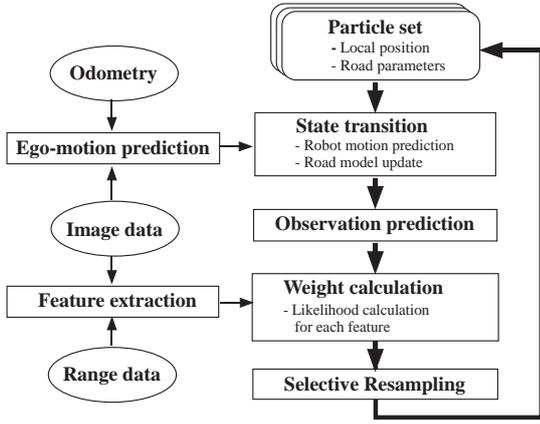


Fig. 1. Overview of the proposed method.

position with respect to the current origin, which is actually the previous robot pose. There are four steps in the iteration:

(1) *State transition* step generates a new particle set. There are mainly two operations in this step. One is the transition due to the motion of the robot, which is predicted from the odometry and image data. The other is road shape and type change which occurs as a new part of the road becomes visible.

(2) *Observation prediction* step predicts the next observation from the robot position and the road parameters.

(3) *Weight calculation* step first determines the likelihood functions from the extracted range and image features and then calculates the importance weight of each particle.

(4) *Selective resampling* step performs resampling only when needed. If the so-called effective number of particles is less than the half of the number of particles, resampling is performed [6].

### III. ROAD MODEL AND STATE VECTOR

This section explains our new road models and state vector representations. In the field of road shape design, straight lines, circular curves, and transition spirals such as clothoids [16] are usually used. Our previous work [13] also used a straight line and a set of circular curves with fixed curvatures. Since a greater variety of road shapes may exist in local environments, we use piecewise-linear road models to represent a local region visible from the robot. The models are continuously updated as the robot moves (see Sec. V-B).

A state vector includes both the robot position and the road parameters, with respect to the previous robot local coordinates, for their simultaneous estimation [13]. The robot position is equivalent to the ego-motion from the previous position, which is represented by 2D translation and the rotation.

#### A. Model for unbranched roads

Fig. 2 shows the model for unbranched roads. The model consists of a set of *road segments*, each of which is either of circular or linear type. The  $i$ th segment  $S_i$  is represented by:

$$S_i = [x_i^l, y_i^l, x_i^r, y_i^r, h_i, \nu_i]^T, \quad (1)$$

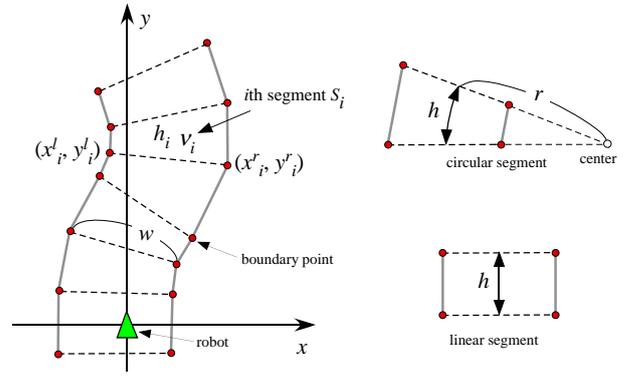


Fig. 2. A piecewise-linear road model (unbranched road model).



Fig. 3. Gaps between road boundaries for range and image data.

where  $(x_i^l, y_i^l)$  and  $(x_i^r, y_i^r)$  are the left and the right boundary point position,  $h_i$  is the segment length, and  $\nu_i = 1/r_i$  is the curvature. The set of segments for each particle has a single width parameter  $w$ , which is also estimated on-line. A gap  $g$  between detected boundary positions by the image and the range sensor is also estimated (see Fig. 3).

This piecewise-linear model can represent arbitrary road shapes by using as many number of segments as necessary, but this may increase the computational cost and decrease the robustness to sensor uncertainties. So we currently fix the number of segments to six and  $h$  to 1.0 [m].

When the robot moves on an unbranched road, we use the following state vector:

$$X = [\Delta x, \Delta y, \Delta \theta, g^l, g^r, S_1, S_2, \dots, S_6]^T. \quad (2)$$

#### B. Model for branching roads

The model for a branching road additionally includes the shape parameters of the branching part. Fig. 4 shows the model for the road with the right branch, consisting of the three parts: front, branching, and rear. The front and the rear part have the same representation as the unbranched road model.

The branching part  $S^b$  has two more parameters than ordinary segments:  $w_b$  for the width of the branch and the radius  $r$  of the branching point.  $S^b$  is represented by:

$$S^b = [x^l, y^l, x^r, y^r, w_b, r]^T. \quad (3)$$

When the robot moves on a branching road, we use the following state vector:

$$X = [\Delta x, \Delta y, \Delta \theta, g^l, g^r, S_1^f, S_2^f, \dots, S^b, S_1^r, S_2^r, \dots]^T, \quad (4)$$

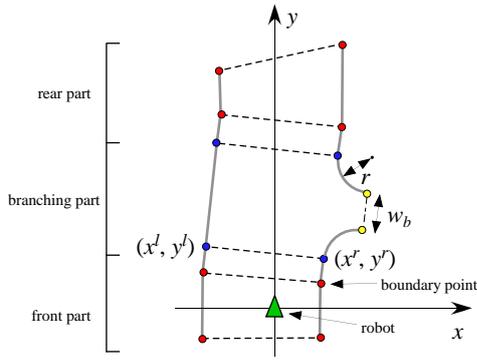


Fig. 4. A branching road model.

where  $S_i^f$  and  $S_i^r$  are the segments for the front and the rear part; the number of these segments varies according to the width  $w_b$  of the branch. The models for left T-branch and crossing are defined similarly.

#### IV. IMAGE AND RANGE DATA PROCESSING FOR IMPORTANCE WEIGHT CALCULATION

The importance weights of particles are calculated using the image and the range data. We do not explicitly extract road boundaries but use likelihood functions for model-based weight calculation. This section briefly explains the image and the range data processing as well as the weight calculation.

##### A. Range data processing

A SICK laser range finder (LRF) is set at the height of  $0.45 [m]$  looking downward by  $5 [deg]$ . If there is a height gap at the road boundary (e.g., at a curb position), the sequence of 3D points forms an L-shape. The nearer the local angle at each point of the sequence is to 90 degrees, the more likely the point is on the boundary. The likelihood is defined as a function of the horizontal position ( $x$ ). Fig. 5(b) shows an example 3D point sequence and the corresponding likelihood function obtained in the scene shown in Fig. 5(a). The right boundary is apparent for LRF thanks to the bank on the right side, while the left one is not visible for LRF.

Fig. 6 illustrates the likelihood calculation for a particle. The road model is mapped on to the road plane and the product of the two likelihood values at the intersection positions is used.

For branching roads, we also evaluate the “flatness” of the road surface at the entrance of the branch (i.e., the intersection between the laser scanning plane and the line connecting two yellow points in Fig. 4). The likelihood of flatness becomes higher when the local angle is nearer to 180 degrees. Fig. 7 shows an example likelihood function for flatness.

##### B. Image data processing

We use a LadyBug2 (Pointgrey Research Inc.) omnidirectional camera system. Two CCD cameras among five are currently used to cover the field of view of about  $144 [deg]$ .

We use two visual cues: road boundary edges and road surface color. We use the intensity and the color gradient image

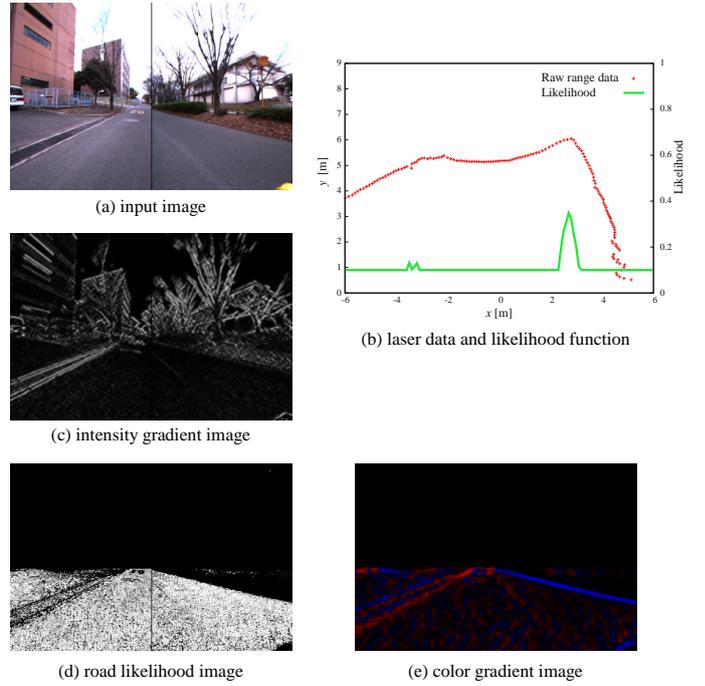


Fig. 5. Likelihood calculation for range and image data.

for the first and the second cues, respectively. Fig. 5(c) is the intensity gradient image for the input image (Fig. 5(a)). To calculate the color gradient image, a road color model is estimated on-line from the latest five images. Fig. 5(d) is the road likelihood image and Fig. 5(e) is the color gradient image; the blue colors and the red colors indicate the likelihood of

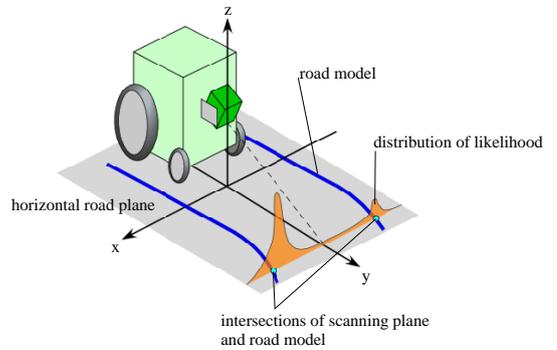


Fig. 6. Likelihood calculation for a particle and the laser data.

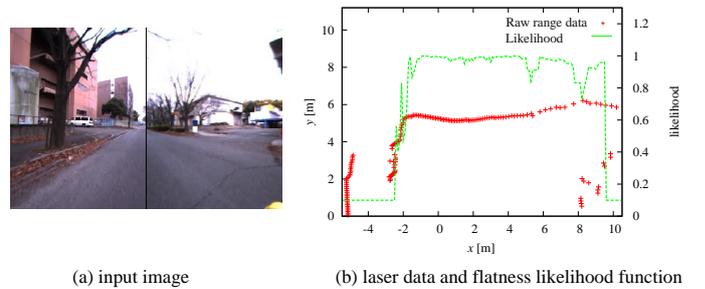


Fig. 7. Flatness likelihood calculation for range data.

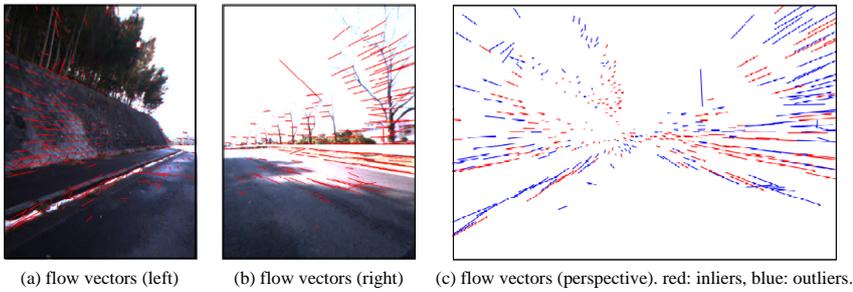


Fig. 8. Flow vectors for motion estimation.

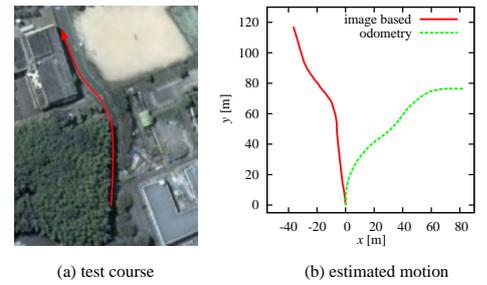


Fig. 9. Motion estimation results.

being on the right and the left boundary point, respectively.

The likelihood of a particle for an image feature is calculated as the averaged likelihood values on the boundaries which are to be obtained by mapping the road boundaries of the particle onto the image (see [13] for more details).

### C. Importance weight calculation

Six likelihood values are calculated for every combination of the three features (laser, edge, and color) and the two sides (left and right). The importance weight of a particle is given by the product of all likelihood values. In some cases, however, the likelihood values for a feature on one side become very small *for any particles* due to, for example, a discontinuity of curb or strong cast shadows. In such a case, the weights for all particles become very small and, as a result, many promising particles might be deleted. To avoid this, if the maximum likelihood for a feature on one side is less than a threshold (currently, 0.3), the combination of the feature and the side is considered not to be effective and is not used.

## V. STATE TRANSITION

The state transition step transforms a set of particles to another set by robot motion prediction and road model update. The former is carried out by a robot motion estimation and a probabilistic sampling. The latter is a key of the proposed method, which adaptively generates new particles to cope with road type changes. Since the robot position and road parameters are represented in the previous robot local coordinates, a coordinate transformation is also performed in this step.

### A. Robot motion prediction from image data and odometry

Robot motion is visually estimated using the eight-point algorithm [7] and odometry. Harris corners are first extracted as feature points in the two images (from the two cameras) and their correspondences are determined between consecutive images. The features are mapped onto a virtual image plane facing right forward and then the eight-point algorithm with RANSAC is applied to the mapped points to calculate the fundamental matrix ( $F$  matrix). Fig. 8(a)-(b) show the extracted flow vectors in the left and the right image. Fig. 8(c) shows the mapped flow vectors; red ones are inliers used for  $F$  matrix calculation and blue ones are outliers.

From  $F$  matrix we can recover the robot motion up to scale, which is given by the odometry. When the estimated

motion is largely different from the odometry, we use the odometry value. Fig. 9 shows an example of motion estimation by image and odometry. Image data usually give better results; through the experiments we have done, about 4% of the image-based motion estimate were rejected as unreliable. A proposal distribution is defined from the estimated ego-motion and empirically-determined uncertainty estimates.

### B. Road model update

As the robot moves, a new part of the road becomes visible. Since the shape of the new part is unknown, we make a set of hypotheses for it. In the particle filter framework, this hypotheses generation (called *road model update*) is realized by generating particles with various road models. The road model update takes place when the robot is judged to enter a new road segment. The previous segment where the robot was is deleted and a new one is attached as shown in Fig. 10.

1) *Update for unbranched road*: In the case of unbranched road, one usual road segment is attached. For each particle which should be updated, the curvature of the attached segment is chosen by sampling. A normal distribution of mean 1 [1/m] and standard deviation 0.04 [1/m] is used as a proposal distribution.

2) *Update for branching road*: The branching parts of a road gradually become visible as the robot moves, similarly to the case of unbranched roads. It is therefore possible to always make hypotheses of branching roads when new road segments are attached. Since the number of branching parts is much smaller than ordinary road segments, however, such a hypothesis generation will be inefficient. We thus add branching road models only when they are likely to be approaching.

For this purpose, we examine the trends of the likelihood values for the intensity gradient, the color gradient, and the flatness of the road along the direction of the road; we calculate their averaged values for all particles and describe them as functions of the distance from the robot along the road (see Fig. 15(c), for example). If the first two values are below a threshold (currently, 0.2) and if the last value is above another threshold (currently, 0.4) on a sufficiently large part (more than 1 [m] long) in the trends, then particles are generated which have branching part starting at the front end of that part. If such a part exists on the left (right) boundary, left (right) T-branch models are generated. If such parts exist on both boundaries, two types of T-branch and crossing models are all generated.

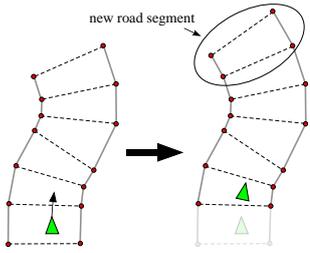


Fig. 10. Road model update.

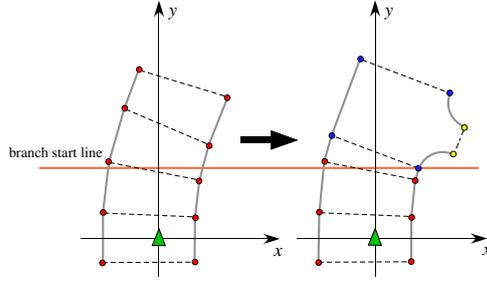


Fig. 11. Generating a branching road model.



Fig. 12. Estimation result for the data shown in Fig. 5.

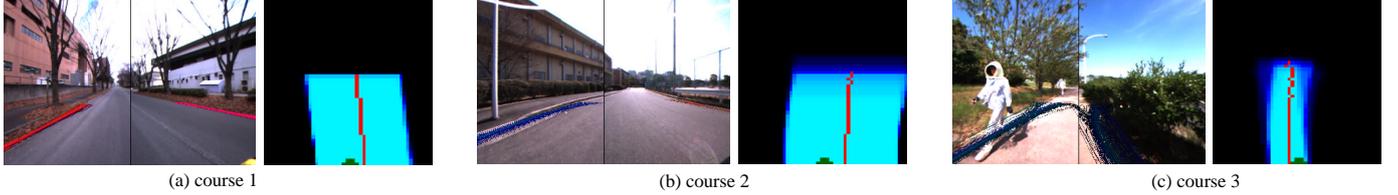


Fig. 13. Estimation results using unbranched road models for other scenes.

The process of generating branching road models is as follows. We first sample an existing unbranched model and replace the part beyond the branch starting point by an appropriate branching part model (see Fig. 11). The branching part is generated using the following proposal distributions: a normal distribution with the estimated starting point of the part being the mean and  $3.0 [m]$  being the standard deviation for the starting position of the branching part, a uniform distribution between  $1 \sim 5 [m]$  for the radius  $r$ , and a uniform distribution between  $3 \sim 7 [m]$  for the width of the branch  $w_b$ .

3) *Number of particles*: We usually keep 500 particles when all models are unbranched roads. When branching road models are included in the particle set, we increase the number to 750. When generating branching road models, we add 50 particles for each model.

## VI. EXPERIMENTAL RESULTS

### A. Results for unbranched road models

1) *Estimation results*: Let us consider Fig. 5. There is a parking space on the left and no curb exists there. There is a bank on the right. Range data is thus effective only for the right road boundary. Concerning image data, the edge information is more effective on the left, while the color information is more effective on the right.

Fig. 12 shows the estimation result. Fig. 12(a) indicates road boundaries obtained from the particle set superimposed on the input image. To see which feature is effective, we assign the three primary colors, red, green, and blue, to color, edge, and range information, respectively. A purple line, for example, indicates that color and range information support the line. In Fig. 12(a), green is dominant on the left boundary because edge information is effective, while red or purple are dominant on the right because range and color information are effective.

Fig. 12(b) shows a kind of certainty distribution of road regions in the robot local coordinates, obtained by voting road regions coming from the current set of particles. Brighter

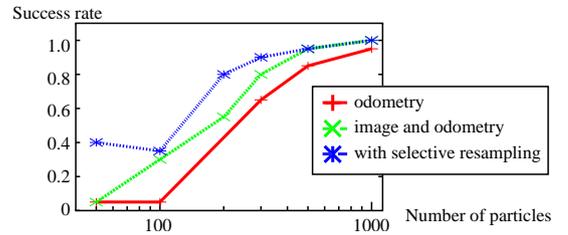


Fig. 14. Effect of image-based proposal and selective sampling.

pixels indicate higher certainties. The green semicircle and the red line in the figure indicate the robot pose and the center position (i.e., skeleton) of the road, respectively. The red line could be a guide for controlling the robot motion. Fig. 13 shows other estimation results for three different roads.

2) *Necessary number of particles*: The number of particles affects the estimation performance. By using a better proposal distribution, it is expected to be able to reduce the number of particles. Fig. 14 illustrates the relationships between the number of particles and the success rate for the three methods: proposal only by odometry, proposal by image data and odometry, and proposal by image data and odometry with selective resampling [6], for course 3 in Fig. 13(c). We ran each method 20 times for each number of particles and calculated the success rate. A run is considered successful if the estimation is judged to be correct until the end by human visual inspection. The figure shows that both the image-based proposal distribution and the selective resampling are effective.

### B. Results for branching road models

Fig. 15 shows a sequence of the estimated branching road models. Graphs in the figure show the likelihood trends for the three features and the parts with a possibility of branching.

At step 33 ((a)~(d)), a branching part candidate appears on the left, although no branches actually exist. A small possibil-

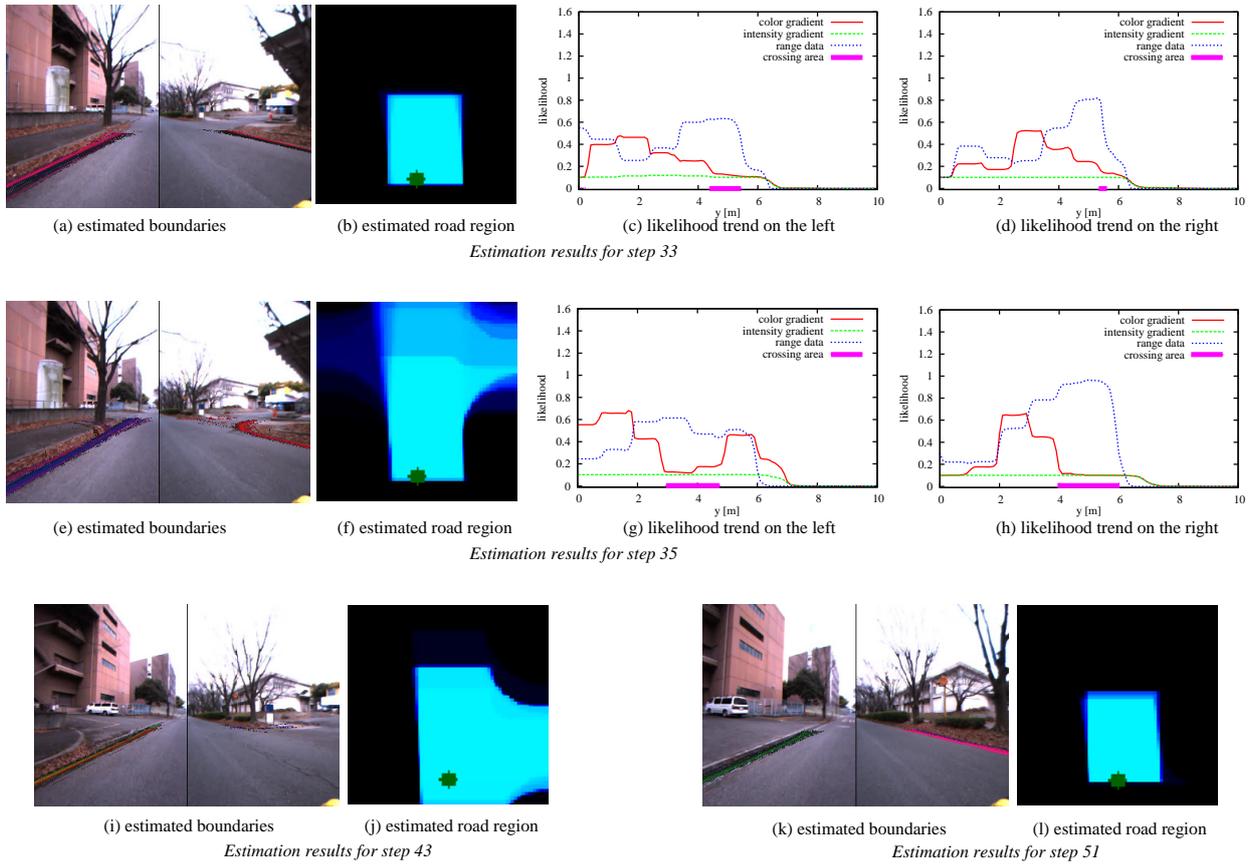


Fig. 15. A sequence of the estimated branching road models.

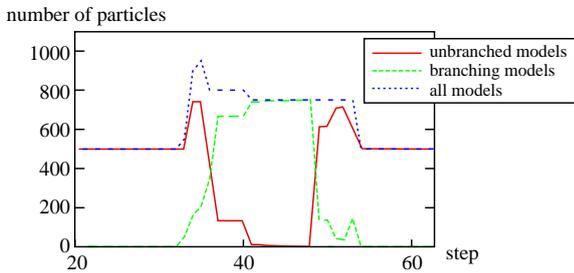


Fig. 16. Change of the number of particles for the results in Fig. 15.

ity of branch also arises on the right, but it is still too small to be considered. Only left T-branch models are generated at this moment. At step 35 ((e)~(h)), all branching models (crossing and two T-branches) are generated and evaluated. The right T-branch models, which are correct, are more highly evaluated as shown in Fig. 15(f). At step 43, the robot is at the position of the branch and almost only the right T-branch models remain. After the robot passed the branch at step 51, only unbranched models exist. There is a strong edge segment on the left, which makes the left side be with no branches (see Fig. 5(a)).

Fig. 16 shows the change of the number of particles for each type. From around step 30, the number of particles increases because branching road models are added. The number sometimes exceeds 750 when the resampling is not performed due to the selective resampling strategy, but it does

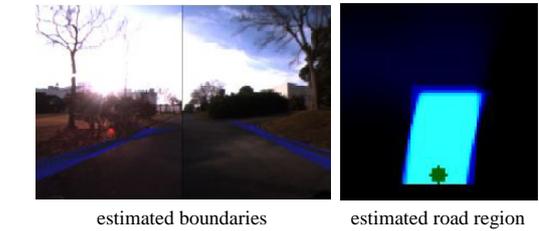


Fig. 18. An incorrect estimate case.

not diverge. Around step 40, only particles for branching roads exists, because the right T-branch is correctly recognized.

Fig. 17 shows the results for branching model estimation for course 2 in Fig. 13(b). The estimation results are mostly acceptable but wrong models also survive. This is mainly due to poorly extracted features. Fig. 18 shows a failure case for a right T-branch. A strong backlight prevented a reliable detection of image features and therefore only range data features were used. Since the LRF detects only one point on each boundary, many right-curved unbranched road survived in this case. This then resulted in examining inappropriate regions (i.e., regions where the actual road boundary does not exist) on the road for left boundaries thereby increasing the weights for false left T-branches. Similar cases may happen when the radius at the entry point of a branch is much larger than expected.

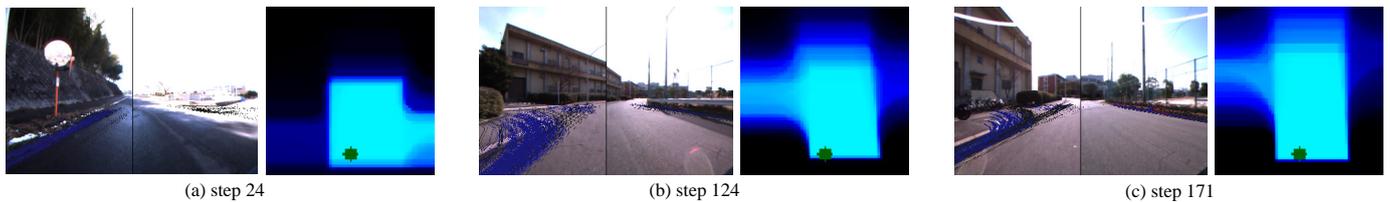


Fig. 17. Estimation results using branching road models for course 2.

### C. On-line navigation

The proposed method was implemented on a mobile robot for an autonomous navigation with on-line road boundary modeling. The cycle time is about 0.6 [s], among which the particle filtering part takes less than 0.1 [s]. For navigating the robot in unbranched roads, a line is fitted to the skeleton of the road region (see Fig. 12(b)) and a turning radius is selected to follow the line. When entering a branching part, the robot stops at the center of that part, rotates by dead reckoning to face a branch to proceed, and restarts to follow the branch as a new unbranched road.

## VII. CONCLUSIONS AND FUTURE WORK

This paper has described a method of robustly modeling road boundaries on-line. Multiple sensory features and flexible road models are effectively integrated in the particle filter framework. The method has been successfully applied to various actual road scenes. The method provides a general approach to road boundary modeling; it is basically applicable to any roads and sensors if we have appropriate road models and likelihood functions. A future work is to extend the road models and their update strategies to cope with a greater variety of road scenes including open spaces and slopes. It is also worth considering to extract and integrate other sensory features such as stereo for improving the reliability of road boundary modeling.

### REFERENCES

- [1] N. Apostoloff and A. Zelinsky. Robust Vision based Lane Tracking using Multiple Cues and Particle Filtering. In *Proceedings of 2003 IEEE Intelligent Vehicles*, pp. 558–563, 2003.
- [2] M. Beauvais and S. Lakshmanan. CLARK: a Heterogeneous Sensor Fusion Method for Finding Lanes and Obstacles. *Image and Vision Computing*, Vol. 18, pp. 397–413, 2000.
- [3] H. Cramer and G. Wanielik. Road Border Detection and Tracking in Non Cooperative Areas with a Laser Radar System. In *Proceedings of German Radar Symposium*, 2002.
- [4] J.D. Crisman and C.E. Thorpe. SCARF: A Color Vision System that Tracks Roads and Intersections. *IEEE Trans. on Robotics and Automat.*, Vol. 9, No. 1, pp. 49–58, 1993.
- [5] E.D. Dickmanns and B.D. Mysliwetz. Recursive 3-D Road and Relative Ego-State Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 199–213, 1992.
- [6] G. Grisetti, C. Stachniss, and W. Burgard. Improving Grid-based SLAM with Rao-Blackwellized Particle Filters by Adaptive Proposals and Selective Resampling. In *Proceedings of 2005 IEEE Int. Conf. on Robotics and Automation*, pp. 2432–2437, 2005.
- [7] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003.
- [8] A.S. Huang, D. Moore, M. Antone, E. Olson, and S. Teller. Multi-Sensor Lane Finding in Urban Road Networks. In *Robotics: Science and Systems IV*, 2008.
- [9] Z. Kim. Robust Lane Detection and Tracking in Challenging Scenarios. *IEEE Trans. on Intelligent Transportation Systems*, Vol. 9, No. 1, pp. 16–26, 2008.
- [10] A. Kirchner and T. Heinrich. Model-Based Detection of Road Boundaries with a Laser Scanner. In *Proceedings of IEEE Int. Symp. on Intelligent Vehicles*, pp. 93–98, 1998.
- [11] R. Labayrade, C. Royere, D. Gruyer, and D. Aubert. Cooperative Fusion for Multi-Obstacles Detection with Use of Stereovision and Laser Scanner. *Autonomous Robots*, Vol. 19, pp. 117–140, 2005.
- [12] D. Langer and T. Jochem. Fusing Radar and Vision for Detecting, Classifying and Avoiding Roadway Obstacles. In *Proceedings of IEEE Int. Symp. on Intelligent Vehicles*, pp. 333–338, 1996.
- [13] Y. Matsushita and J. Miura. Simultaneous Estimation of Road Region and Ego-Motion with Multiple Road Models. In *Proceedings of 2008 IEEE Int. Conf. on Multisensor Fusion and Integration for Intelligent Systems*, pp. 526–532, 2008.
- [14] M. Okutomi, K. Nakano, J. Matsuyama, and T. Hara. Robust Estimation of Planar Regions for Visual Navigation Using Sequential Stereo Images. In *Proceedings of 2002 IEEE Int. Conf. on Robotics and Automation*, pp. 3321–3327, 2002.
- [15] S. Panzneri, F. Pascucci, and G. Ulivi. An Outdoor Navigation System Using GPS and Inertial Platform. In *Proceedings of 2001 IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics*, Vol. 2, pp. 1346–1351, 2001.
- [16] V.S.R. Sasipalli, G.S. Sasipalli, and K. Harada. Single Spirals in Highway Design and Bounds for Their Scaling. *IEICE Trans. on Information and Systems*, Vol. E80-D, No. 11, pp. 1084–1091, 1997.
- [17] S. Sehestedt, S. Kodagoda, A. Alempijiv, and G. Dissanayake. Robust Lane Detection in Urban Environments. In *Proceedings of 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 123–128, 2007.
- [18] M.A. Sotelo, F.J. Rodriguez, L. Magdalena, L.M. Bergasa, and L. Boquete. A Color Vision-Based Lane Tracking System for Autonomous Driving on Unmarked Roads. *Autonomous Robots*, Vol. 16, pp. 95–116, 2004.
- [19] R. Thrapp, C. Westbrook, and D. Subramanian. Robust Localization Algorithms for an Autonomous Campus Tour Guide. In *Proceedings of 2001 IEEE Int. Conf. on Robotics and Automation*, pp. 2065–2071, 2001.
- [20] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. The MIT Press, 2005.
- [21] S. Thrun and et al. Stanley: The Robot that Won the DARPA Grand Challenge. *J. of Field Robotics*, Vol. 23, No. 9, pp. 661–692, 2006.
- [22] R. Wallece, K. Matsuzaki, Y. Goto, J. Crisman, J. Webb, and T. Kanade. Progress in Robot Road-Following. In *Proceedings of 1986 IEEE Int. Conf. on Robotics and Automation*, pp. 1615–1621, 1986.
- [23] W.S. Wijesoma, K.R.S. Kodagoda, and A.P. Balasuriya. Road Boundary Detection and Tracking Using Ladar Sensing. *IEEE Trans. on Robotics and Automation*, Vol. 20, No. 3, pp. 456–464, 2004.