



On-line road boundary modeling with multiple sensory features, flexible road model, and particle filter

Y. Matsushita, J. Miura*

Department of Computer Science and Engineering, Toyohashi University of Technology, Toyohashi, Aichi 441-8580, Japan

ARTICLE INFO

Article history:

Available online 5 March 2011

Keywords:

Outdoor navigation
Mobile robot
Road boundary modeling
Particle filter

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 with cameras and a laser range finder. To show that the proposed method is applicable to other sensors, preliminary results of using stereo instead of the laser range finder are also described.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Autonomous outdoor navigation has been one of the active research areas in robotics, from Navlab [1] to Grand Challenge [2]. 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., [3,4]). 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 [5,6], while others first detect road regions using, for example, color information to determine the road boundaries [7,8].

Range sensing is also popular in road boundary detection [9–11]. 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 [2]. Stereo vision can also be used for extracting road region as a planar region [12]. 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 [13,14]. Some works use range information for refining the image-based road detection process [5,15,2].

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 [6] developed a Kalman filter-based method which estimates the 3D road parameters and the vehicle ego-motion. Apostoloff and Zelinsky [16] proposed a particle filter-based lane detector using vision with a simple road model. Kim [17] proposed a robust lane detection and tracking method based on explicit lane marking detection and particle filtering. Danescu and Nedevchi [18] proposed a similar method for integrating lane and curb information obtained by stereo vision. 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. [19] applied a particle filter for detecting lane marking in each image not for tracking them.

Coping with various road types is also an important issue. Many previous methods deal with only unbranched roads where detecting a pair of (mostly parallel) road boundaries is the task. In local roads, however, we need to cope with frequent road branches.

We have been developing a particle filter-based road boundary detection method. 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 integrated via specially-designed likelihood functions. In addition, the method uses flexible road models which can represent both unbranched and branching roads and controls the transition from one model to another using the trends of evidence. Such an integration

* Corresponding author. Tel.: +81 532 44 6773; fax: +81 532 44 6757.

E-mail address: jun@ics.tut.ac.jp (J. Miura).

URL: <http://www.aisl.cs.tut.ac.jp/> (J. Miura).

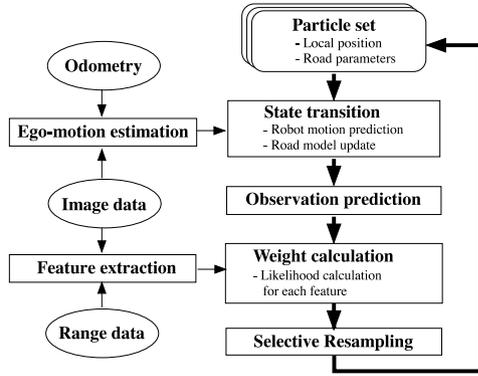


Fig. 1. Overview of the proposed method.

of multiple sensory features and flexible road model transitions are naturally implemented in a particle filter-based framework. Previous particle filter-based methods [16–18] use more simple models for roads and their transitions or rely on the results of extracting specific features, while our method uses more flexible models and does not explicitly extract image features, thereby being more robust to a variety of road scenes.

Another feature of the proposed method lies in its generality in the sense that it can deal with various road scenes by appropriately selecting likelihood functions and road models. To show this, we also describe preliminary results of using stereo instead of a laser range finder to extract shape features.

The rest of the paper is organized as follows. Section 2 describes an overview of the proposed method. Section 3 explains the state vector and the road models for unbranched and branching roads. Section 4 explains the image and the range data processing for the importance weight calculation. Section 5 describes the state transition step in the particle filter which includes robot motion prediction and road model update. Section 6 shows experimental results in various road scenes. Section 7 describes an attempt to use stereo vision instead of a laser range finder in the proposed framework. Section 8 concludes the paper and discusses future work.

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

- (1) *The 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) *The observation prediction* step predicts the next observation from the robot position and the road parameters.
- (3) *The 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) *The 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 [21].

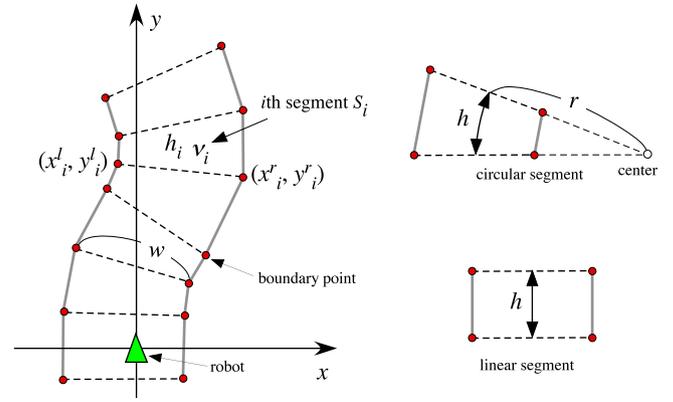


Fig. 2. A piecewise-linear road model (unbranched road model).

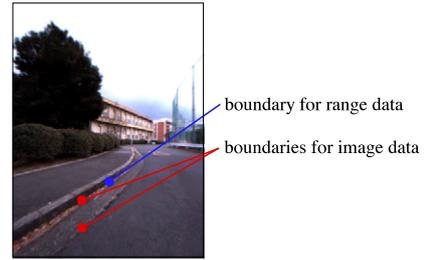


Fig. 3. Gaps between road boundaries for range and image data.

3. Road model and state vector

This section explains our road models and state vector representations. In the field of road shape design, straight lines, circular curves, and transition spirals such as clothoids [22] are usually used. Most previous works also use straight lines and parametrized curves such as circular ones. Since a greater variety of road shapes may exist in local environments, however, 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 Section 5.2).

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

3.1. 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:

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

where (x_i^l, y_i^l) and (x_i^r, y_i^r) are the left and the right boundary point positions, h_i is the segment length, and $v_i = 1/r_i$ 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].

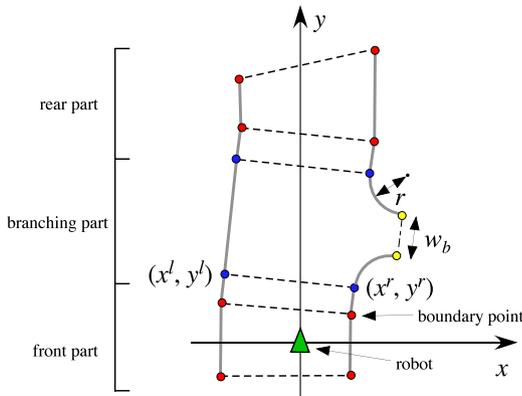


Fig. 4. A branching road model (right T-branch).

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

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

3.2. 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:

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

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 model for left T-branch has the same form, while that for crossing has the following form for the branching part:

$$S^c = [x^l, y^l, x^r, y^r, w_b^l, w_b^r, r^l, r^r] \quad (5)$$

4. 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 explains the image and the range data processing as well as the likelihood and weight calculation.

4.1. Range data processing

A SICK laser range finder (LRF) is set at the height of 0.45 [m] looking downward by 5 [degree] (see Fig. 6). 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 value is determined by the local angle using a normal distribution with the mean being 90 degrees and the standard deviation being 30 degrees, and calculated at each 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 almost undetectable 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

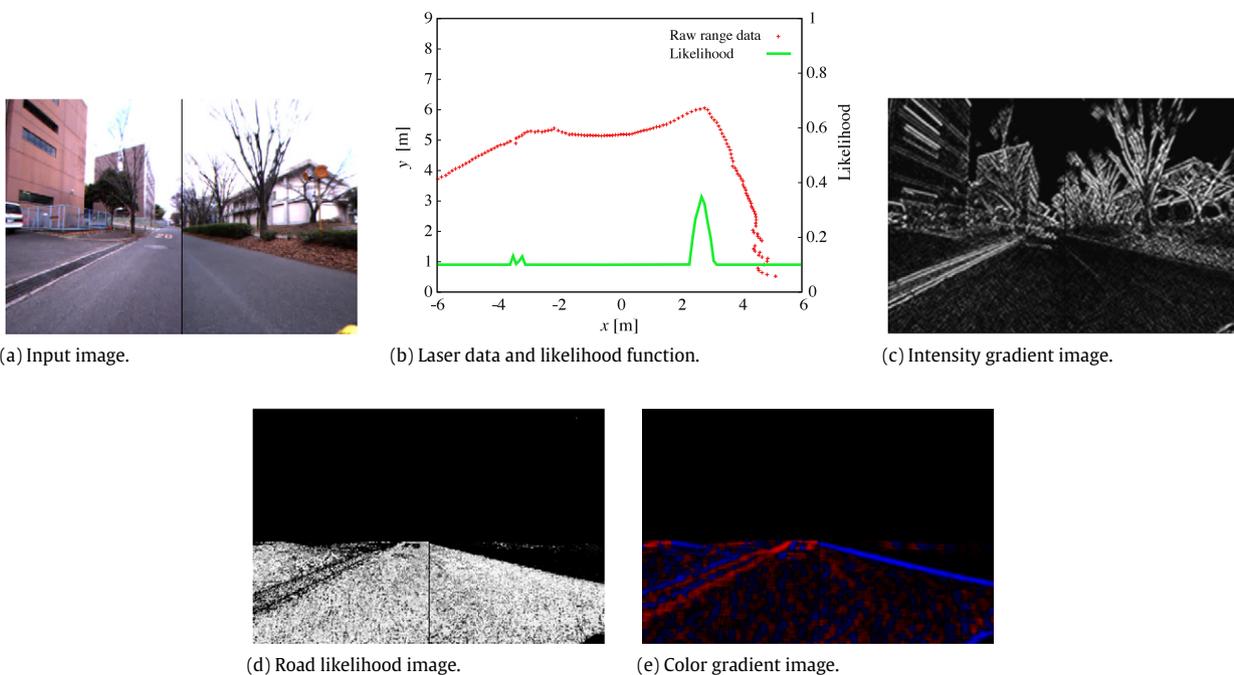


Fig. 5. Likelihood calculation for range and image data.

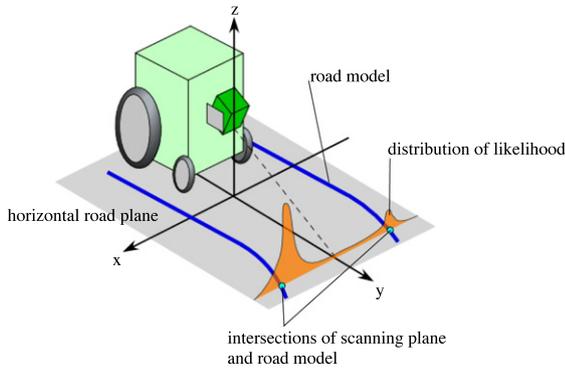
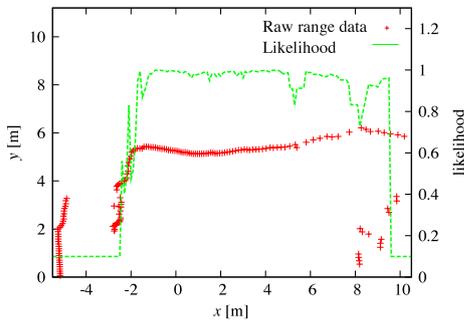


Fig. 6. Likelihood calculation for a particle and the laser data.



(a) Input image.



(b) Laser data and flatness likelihood function.

Fig. 7. Flatness likelihood calculation for range data.

angle is nearer to 180 degrees. So we use a normal distribution with the mean being 180 degrees for calculating the likelihood values. Fig. 7 shows an example likelihood function for flatness.

4.2. 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 [degrees]. We use two visual cues: road boundary edges and road surface color.

Road boundary edges are effective if strong edge segments of lane markers or shadow boundaries caused by curbs are available. We first apply a 3×3 median filter, followed by a Sobel filter with 11×11 Gaussian smoothing to calculate the intensity gradient. Fig. 5(c) shows the intensity gradient image for the input image (Fig. 5(a)).

To use the color cue, we first estimate the road surface color. The input image is transformed to a CIE $L^*a^*b^*$ image and the a^*b^* 2D space is used for representing the color. We assume a single color model and describe the surface color using a 2-D normal distribution in the 2D color space. The mean vector and the covariance matrix are estimated on-line using 500 samples on the estimated road region in the latest five frames (i.e., 100 samples from one frame). Using the estimated color model, the likelihood

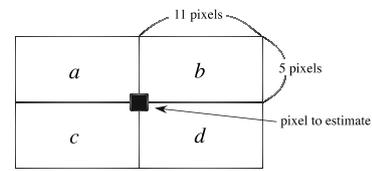


Fig. 8. Windows for calculating the color gradient.

of each pixel in the current image is calculated. Fig. 5(d) shows the road likelihood image.

The gradient of the road likelihood is then calculated as follows. Since the direction of the road boundary in the image varies according to the relative pose between the robot and the road, we use the windows shown in Fig. 8 to calculate the gradient as follows. When calculating the gradient for the right boundary, since the likelihood of the left-hand side is larger, we use:

$$\max(C_b - C_a, C_b - C_c, C_d - C_a, C_d - C_c) \quad (6)$$

as the gradient value, while for the left boundary, we use:

$$\max(C_a - C_b, C_a - C_d, C_c - C_b, C_c - C_d), \quad (7)$$

where C_* is the likelihood value of the corresponding region in Fig. 8. Fig. 5(e) is the resultant color gradient image, where the magnitude of red and blue indicate the degree of being on the left and the right boundary, respectively.

The intensity and the color gradient image are normalized to the range [0, 255] and used for likelihood calculation. The likelihood of a particle for an image feature is calculated as follows. The road model with respect to the robot pose is mapped onto the image and the gradient values under the mapped boundary are collected and averaged. This averaged value is then transformed to a likelihood value ranging [0, 1] using a sigmoid function.

4.3. 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.

5. 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 an ego-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.

5.1. Robot motion prediction from image data and odometry

Robot motion is visually estimated using the eight-point algorithm [23] 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

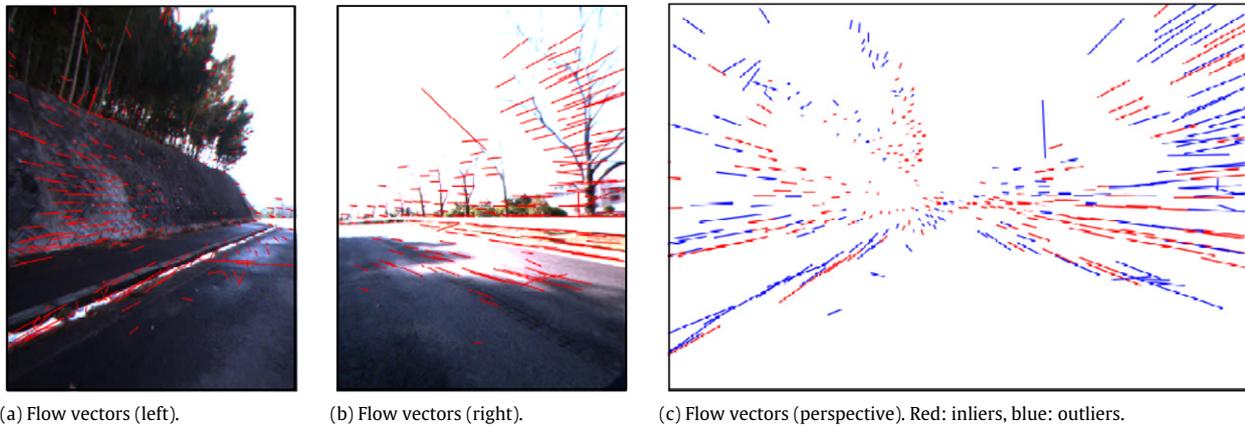


Fig. 9. Flow vectors for motion estimation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

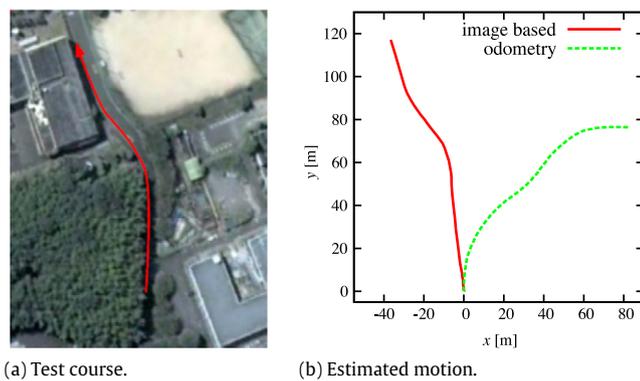


Fig. 10. Motion estimation results.

right forward and then the eight-point algorithm with RANSAC is applied to the mapped points to calculate the fundamental matrix (F matrix). Fig. 9(a)–(b) show the extracted flow vectors in the left and the right image. Fig. 9(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. 10 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.

5.2. 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. 11.

5.2.1. 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

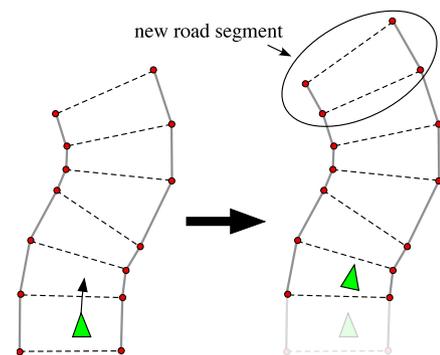


Fig. 11. Road model update.

distribution of mean 1 [1/m] and standard deviation 0.04 [1/m] is used as a proposal distribution.

5.2.2. 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 may waste particles. 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. 18(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 a 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.

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. 12). 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

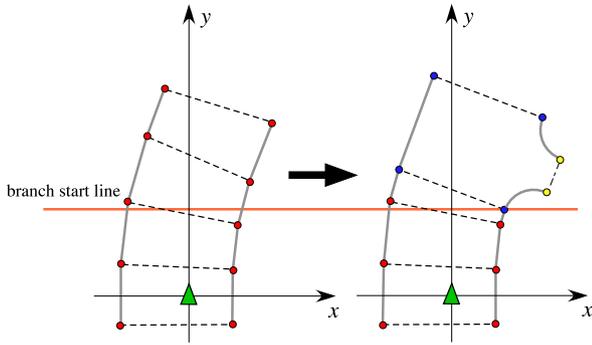


Fig. 12. Generating a branching road model.



(a) Course 1.

(b) Course 3.

Fig. 13. Two out of three test courses. The other is shown in Fig. 10(a).

branching part, a uniform distribution between 1 and 5 [m] for the radius r , and a uniform distribution between 3 and 7 [m] for the width of the branch w_b .

5.2.3. 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.

6. Experimental results

6.1. Results for unbranched road models

This subsection describes results for unbranched road models. We used three courses. Fig. 13 shows courses 1 and 3 and Fig. 10(a) indicates course 2.

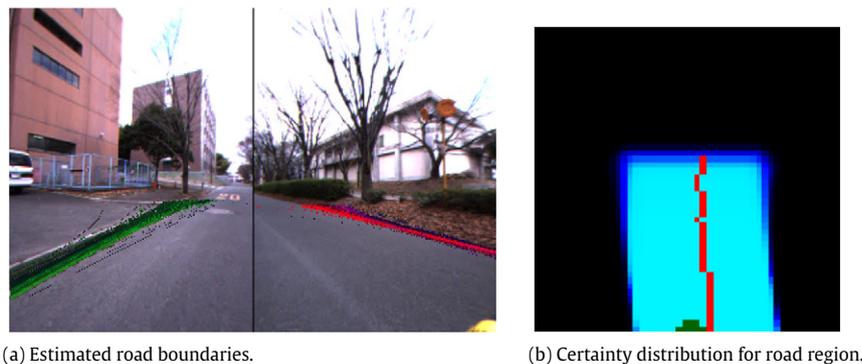
6.1.1. Estimation results

Let us consider Fig. 5(a). 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. 14 shows the estimation result. Fig. 14(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. 14(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. 14(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 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.

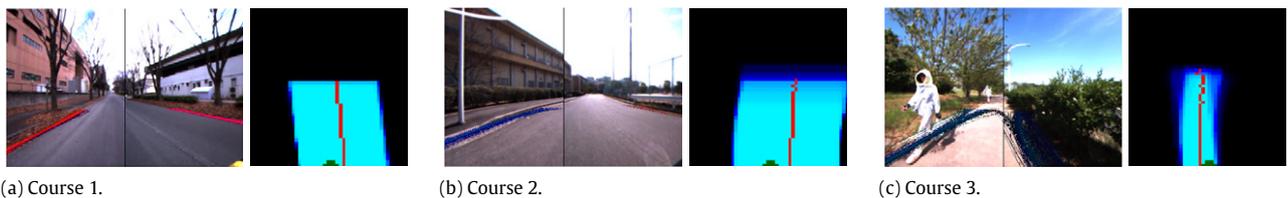
Fig. 15 shows other estimation results for three different roads. In the course 3 result, due to less distinct features and an occlusion by a person, the estimate of a far part of the road is uncertain, that is, the particle set has a large variety of road parameters. It is, however, possible for the robot to follow the road because the estimate of that part becomes clearer as it approaches to the robot.



(a) Estimated road boundaries.

(b) Certainty distribution for road region.

Fig. 14. Estimation result for the data shown in Fig. 5. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) Course 1.

(b) Course 2.

(c) Course 3.

Fig. 15. Estimation results using unbranched road models for other scenes.

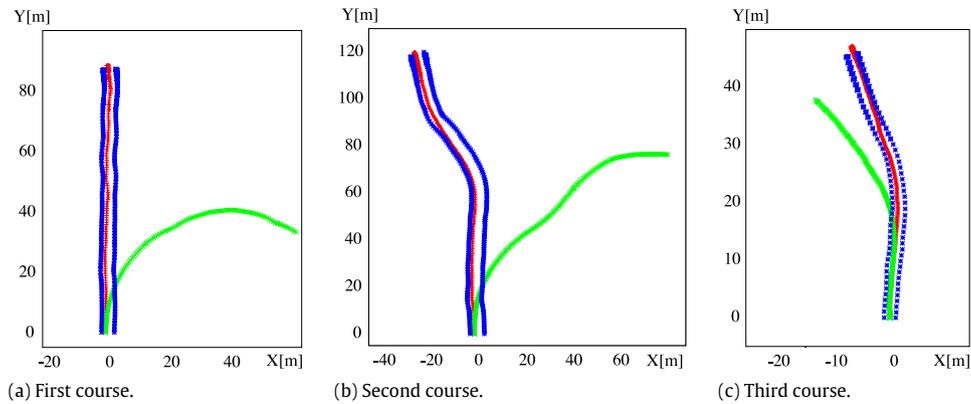


Fig. 16. Road shape reconstruction results. Blue: road boundaries, Red: estimated motion, Green: motion estimation by odometry. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

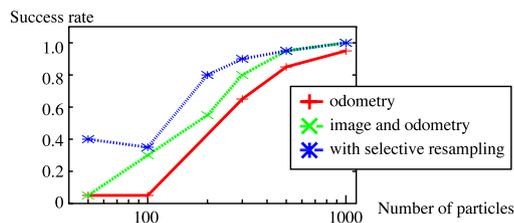


Fig. 17. Effect of image-based proposal and selective sampling.

6.1.2. Reconstruction of the road shape

Each particle has its parent, from which that particle was generated at the state transition step. By tracing back from the final set of particles, we can reconstruct the global road shape and the motion history. Fig. 16 shows the reconstructed road shapes with the estimated motion histories obtained by the proposed method and by odometry only. Although the proposed method is intended for local road boundary modeling, the global road shapes are reconstructed reasonably well.

6.1.3. 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. 17 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 [21], for course 3 in Fig. 15(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 of the resultant road boundaries mapped onto the image sequence. The figure shows that both the image-based proposal distribution and the selective resampling are effective.

6.2. Results for branching road models

Fig. 18 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 possibility of a 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 (a 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. 18(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 leaves the left side with no branches (see Fig. 5(a)).

Fig. 19 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 not diverge. Around step 40, only particles for branching roads exist, because the right T-branch is correctly recognized.

Fig. 20 shows the results for branching model estimation for course 2 in Fig. 15(b). The estimation results are mostly acceptable but wrong models also survive. This is mainly due to poorly extracted features.

Fig. 21 shows the reconstructed road shapes with the estimated motion histories using branching road models for courses 1 and 2. The right T-branch in Fig. 21(a) corresponds to the one detected in Fig. 18. Three branches in Fig. 21(b) (from bottom to top) correspond to the results shown in Fig. 20 (from left to right), respectively. Note that the first branch in Fig. 21(b) is recognized as a crossing; this is because some particles corresponding to crossings, which are shown in Fig. 20(a), happened to survive in the end, although most of the particles recognized the branch as a right T-branch when the robot was there.

Fig. 22 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 roads 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. Further analysis of such cases and improvement of the method will be necessary.

6.3. 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]. To navigate the robot on unbranched roads, a line is fitted to the skeleton of the road region (see Fig. 14(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. Fig. 23 shows snapshots of an autonomous driving in course 3.

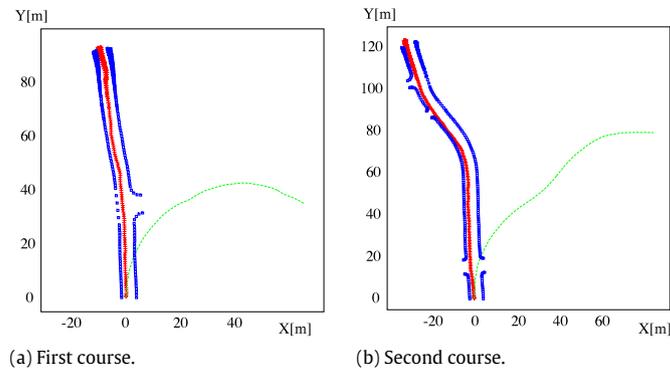


Fig. 21. Road shape reconstruction results for branching roads. Blue: road boundaries, Red: estimated motion, Green: motion estimation by odometry. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



Fig. 22. An incorrect estimate case.



Fig. 23. Snapshots of autonomous driving.

information are more effective. Fig. 25 shows the estimation results for three different locations.

8. 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 using cameras and a laser range finder. 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. This has been shown, to some extent, by preliminary results using a stereo camera, instead of the laser range finder, for obtaining shape information of road boundaries.

To cope with a greater variety of road scenes including open spaces and slopes, we are planning to extend the road models and

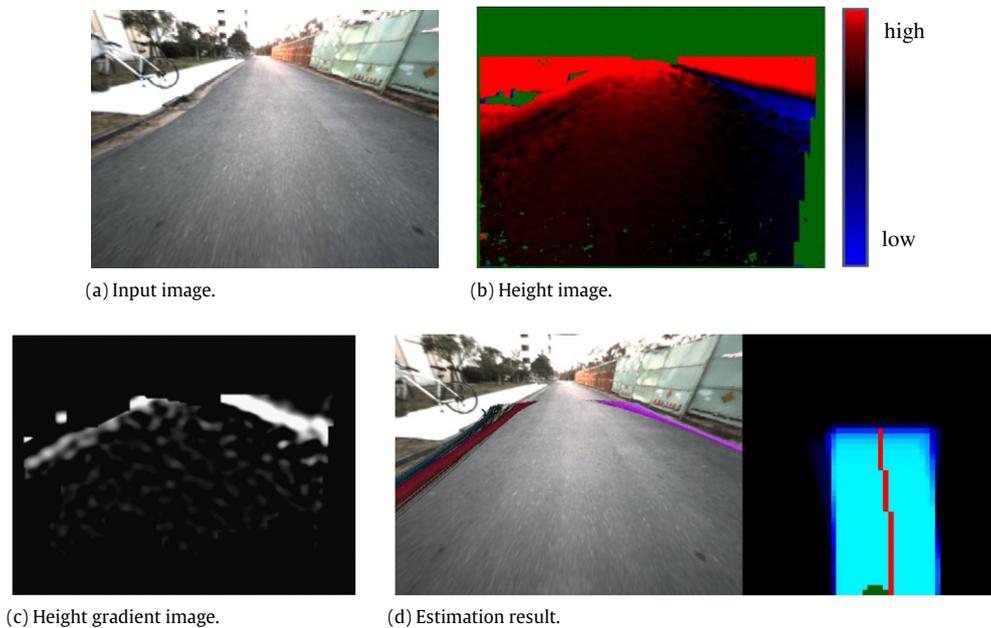


Fig. 24. Stereo-based road boundary modeling.



Fig. 25. Estimation results using stereo.

their update strategies. Combining the proposed road boundary modeling with a global localization (e.g., [24]) for realizing an autonomous navigation in outdoor environments is also future work.

Acknowledgements

The authors would like to thank Mr. Takeshi Chiku for implementing the stereo-based road boundary detection. This work is supported in part by Grant-in-Aid for Scientific Research (No. 21300075) from JSPS.

References

- [1] R. Wallece, K. Matsuzaki, Y. Goto, J. Crisman, J. Webb, T. Kanade, Progress in robot road-following, in: Proceedings of 1986 IEEE Int. Conf. on Robotics and Automation, 1986, pp. 1615–1621.
- [2] S. Thrun, et al., Stanley: the robot that won the DARPA grand challenge, *Journal of Field Robotics* 23 (9) (2006) 661–692.
- [3] S. Panzneri, F. Pascucci, 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, 2001, pp. 1346–1351.
- [4] R. Thrapp, C. Westbrook, D. Subramanian, Robust localization algorithms for an autonomous campus tour guide, in: Proceedings of 2001 IEEE Int. Conf. on Robotics and Automation, 2001, pp. 2065–2071.
- [5] M. Beauvais, S. Lakshmanan, CLARK: a heterogeneous sensor fusion method for finding lanes and obstacles, *Image and Vision Computing* 18 (2000) 397–413.
- [6] E.D. Dickmanns, B.D. Mysliwetz, Recursive 3-D road and relative ego-state recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 14 (2) (1992) 199–213.
- [7] J.D. Crisman, C.E. Thorpe, SCARF: A color vision system that tracks roads and intersections, *IEEE Transactions on Robotics and Automation* 9 (1) (1993) 49–58.
- [8] M.A. Sotelo, F.J. Rodriguez, L. Magdalena, L.M. Bergasa, L. Boquete, A color vision-based lane tracking system for autonomous driving on unmarked roads, *Autonomous Robots* 16 (2004) 95–116.
- [9] H. Cramer, G. Wanielik, Road border detection and tracking in non cooperative areas with a laser radar system, in: Proceedings of German Radar Symposium, 2002.
- [10] A. Kirchner, T. Heinrich, Model-based detection of road boundaries with a laser scanner, in: Proceedings of IEEE Int. Symp. on Intelligent Vehicles, 1998, pp. 93–98.
- [11] W.S. Wijesoma, K.R.S. Kodagoda, A.P. Balasuriya, Road boundary detection and tracking using ladar sensing, *IEEE Transactions on Robotics and Automation* 20 (3) (2004) 456–464.
- [12] M. Okutomi, K. Nakano, J. Matsuyama, 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, 2002, pp. 3321–3327.
- [13] R. Labayrade, C. Royere, D. Gruyer, D. Aubert, Cooperative fusion for multi-obstacles detection with use of stereovision and laser scanner, *Autonomous Robots* 19 (2005) 117–140.
- [14] D. Langer, T. Jochem, Fusing radar and vision for detecting, classifying and avoiding roadway obstacles, in: Proceedings of IEEE Int. Symp. on Intelligent Vehicles, 1996, pp. 333–338.
- [15] A.S. Huang, D. Moore, M. Antone, E. Olson, S. Teller, Multi-sensor lane finding in urban road networks, in: *Robotics: Science and Systems IV*, 2008.
- [16] N. Apostoloff, A. Zelinsky, Robust vision based lane tracking using multiple cues and particle filtering, in: Proceedings of 2003 IEEE Intelligent Vehicles, 2003, pp. 558–563.
- [17] Z. Kim, Robust lane detection and tracking in challenging scenarios, *IEEE Transactions on Intelligent Transportation Systems* 9 (1) (2008) 16–26.
- [18] R. Danescu, S. Nedevschi, Probabilistic lane tracking in difficult road scenarios using stereovision, *IEEE Transactions on Intelligent Transportation Systems* 10 (2) (2009) 272–282.
- [19] S. Sehestedt, S. Kodagoda, A. Alempijic, G. Dissanayake, Robust lane detection in urban environments, in: Proceedings of 2007 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2007, pp. 123–128.
- [20] S. Thrun, W. Burgard, D. Fox, *Probabilistic Robotics*, The MIT Press, 2005.
- [21] G. Grisetti, C. Stachniss, 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, 2005, pp. 2432–2437.
- [22] V.S.R. Sasipalli, G.S. Sasipalli, K. Harada, Single spirals in highway design and bounds for their scaling, *IEICE Transactions on Information and Systems* E80-D (11) (1997) 1084–1091.
- [23] R. Hartley, A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed., Cambridge University Press, 2003.

- [24] J. Miura, K. Yamamoto, Robust view matching-based markov localization in outdoor environments, in: Proceedings of 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, 2008, pp. 2970–2976.



Y. Matsushita received the B.Eng. degree and the M.Eng. degree in mechanical Engineering in 2007 and 2009, respectively, from Osaka University. He had been working on outdoor mobile robots at Osaka University.



J. Miura received his B.Eng. degree in mechanical engineering in 1984, his M.Eng. and his Dr.Eng. degrees in information engineering in 1986 and 1989, respectively, all from the University of Tokyo, Tokyo, Japan. In 1989, he joined Department of Computer-Controlled Mechanical Systems, Osaka University, Suita, Japan. Since April 2007, he has been a Professor at Department of Information and Computer Sciences, Toyohashi University of Technology, Toyohashi, Japan. From March 1994 to February 1995, he was a Visiting Scientist at the Computer Science Department, Carnegie Mellon University, Pittsburgh, PA. He received the Best Paper Award from the Robotics Society of Japan in 1997. He was also selected as one of the six finalists for the Best Paper Award at 1995 IEEE International Conference on Robotics and Automation. Prof. Miura has published over 100 papers in international journals and conferences in the areas of intelligent robotics, computer vision, and artificial intelligence.