2012 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 7-12, 2012. Vilamoura, Algarve, Portugal

On-line Road Boundary Estimation by Switching Multiple Road Models using Visual Features from a Stereo Camera

Takeshi Chiku and Jun Miura
Department of Computer Science and Engineering
Toyohashi University of Technology

***Abstract*— This paper describes a road boundary estimation method for autonomous navigation. We consider navigation in a campus environment, where roads (or traversable regions) are not necessarily modeled by a typical road model with a pair of parallel lines, but have a variety of shapes. We therefore use a set of flexible road models with model transition mechanisms for a robust road boundary estimation. This new modeling is incorporated into our multiple sensory feature-based road boundary estimation framework using a particle filter. The proposed method has been successfully applied to various scenes in our campus to realize autonomous navigation.**

***Index Terms*— Outdoor navigation, Mobile robot, Road boundary estimation, Stereo vision, Particle filter.**

I. INTRODUCTION

Personal service robots are expected to help people in various scenes of their everyday life. Since activities in outdoor is an important part of life and the autonomous outdoor navigation capability is therefore indispensable for such a robot. A fully autonomous navigation entails many functions such as route planning, localization, road detection and following, and obstacle avoidance. This paper focuses on road (or traversable region) detection.

Vision has been widely used for road boundary detection and estimation. Some methods detect road and lane boundaries directly [3], while others first detect road regions using color information [14]. Most of these works rely on a single or a few predefined visual features. Range sensors [15], [7] or stereo cameras [11] are also popular for detecting roads or traversable regions; they assume geometric discontinuity (e.g., curbs) at road boundaries, and might not be effective in some roads like a small trail among low grasses.

It is also necessary to cope with occasional sensing failures or missing effective features (e.g., a discontinuity of curbs). Many model-based filtering approaches using Kalman filters [6], particle filters [1], [5], or swarm-based optimization [12] have been shown to be effective.

One important issue in road boundary estimation, especially for personal service robots, is how to cope with a variety of road scenes; effective sensory information for road boundary detection may vary from place to place and the road shape may be complex (see Fig. 1). Use of multiple visual/sensory features is effective for increasing the robustness to the variety of boundary appearances; for example, filter-based methods are suitable for integrating multiple cues [1], [9]. Concerning estimation of complex road shape, Bai et al. [2] deal with the parametric estimation of road boundaries composed of a sequence of connected arcs using multiple particle filters. Few methods, however, have dealt with various road shapes as shown in Fig. 1.

(a) various road appearances.

(b) various road shapes.

Fig. 1. Examples of road scenes.

We have been developing a particle filter-based road boundary estimation method [10], [4]. By using flexible road models with branches, the method can recognize branching roads, which most of existing works have not dealt with. In this paper, we greatly extend the road models to cope with a much larger variety of road scenes. Thanks to our approach which handles road model changes at the state transition step in particle filtering, we can achieve the extension in a principled manner.

The rest of the paper is organized as follows. Sec. II explains our road boundary estimation framework. Sec. III describes road models and their parameter update. Sec. IV describes the transition between road models to cope with road type changes. Sec. V shows experimental results. Sec. VI concludes the paper and discusses future work.

II. ROAD BOUNDARY ESTIMATION FRAMEWORK

This section provides basic ideas of our multi-sensory road estimation framework [10], [4]; Fig. 2 shows an overview. The right-hand side of the figure indicates the iteration of a particle filter, while the left-hand side indicates the sensor data processing. We here use the simplest road model, that is, the *unbranched road* model as an introductory example.

A. Road model and state vector

In the field of road shape design, straight lines, circular curves, and transition spirals such as clothoids [13] are usually used. Many previous road boundary estimation methods use straight and/or parametric curve models. Since a

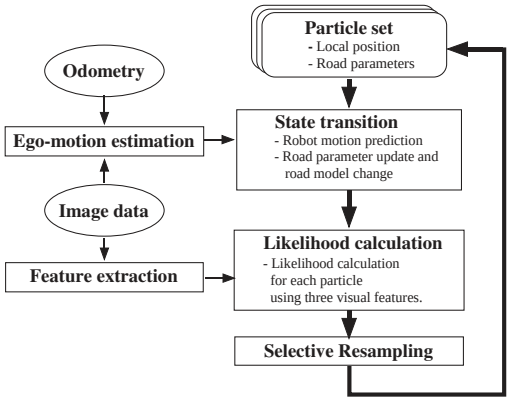
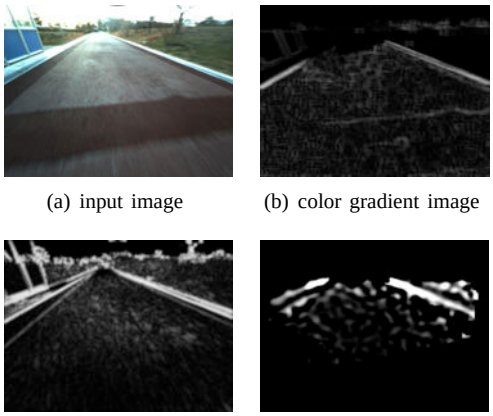


Fig. 2. Overview of road boundary estimation process.



(a) input image (b) color gradient image

(c) intensity gradient image (d) height gradient image

Fig. 4. Input and gradient images for road boundary features.

greater variety of road shape may exist in local environments, however, and since precise reconstruction of road shape is not the purpose of the method, we use piecewise-linear road models to represent a road shape in a local region visible from the robot.

An unbranched road is the one which does not have any branches and has almost constant widths along the road. This type of road is modeled by a sequence of *segments*, as shown in Fig. 3 (left). Each segment S_i has as parameters two boundary points on both sides ((x_i^r, y_i^r) and (x_i^l, y_i^l)), local curvature ν_i, and length h_i. This road type also has road width w, which is common to the current set of segments but is estimated on-line, and gap parameter $G = [g_e^r, g_e^l, g_h^r, g_h^l]$, which is to deal with the case where different boundaries are supported by different visual features. Refer to [4] for the effect of the gap parameter.

The proposed method estimates the robot pose and the road parameters simultaneously. So we use the following state vector:

$$X_u = [U, w, G, S_1, S_2, \ldots, S_6], \quad (1)$$

$$U = [\Delta x, \Delta y, \Delta\theta], \quad (2)$$

where U is the robot pose. The robot pose and the road parameters are defined in the previous robot coordinates. The number of segments is fixed to six for unbranched roads.

B. State transition

The state transition step in the particle filter transforms a set of particles to another set by a robot ego-motion estimate and road parameter updates. The former is calculated from visual odometry [10]. The latter 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. 3 (right). The curvature of the newly added segment is determined to follow a normal distribution; their mean and variance are set to that of the rear-most segment to which the new one is attached and $0.04\,[1/m]$, respectively.

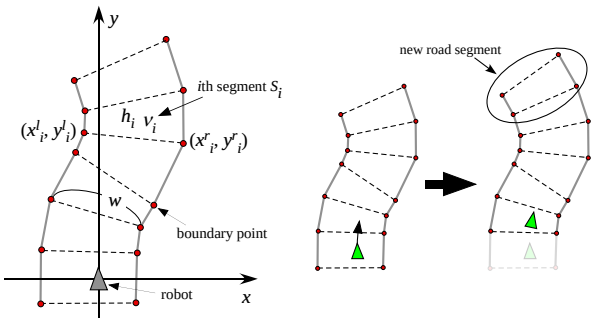


Fig. 3. Unbranched road model (left) and road parameter update (right).

Road model changes, the other type of transitions such as the one from an unbranched to a branching, will be described in Sec. IV.

C. Likelihood calculation

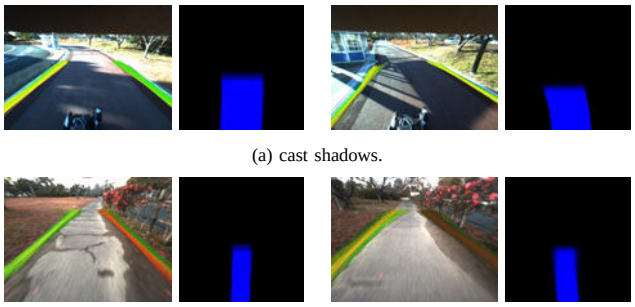
This step calculates the likelihood of each particle using stereo image data. We extract three types of image information, color, edge, and height, as evidence for the existence of road boundaries, and suppose that at least one type of the information exhibits a large change at boundary positions.

We do not explicitly extract road boundaries. Instead, we calculate *gradient images* for the three types of information and calculate the weight of a particle by projecting the road boundary given by the road parameters of the particle and by summing up the gradient values at the mapped boundary positions. The summed value is converted to a likelihood using some sigmoid function. Fig. 4 shows an example set of three gradient images.

The likelihood of a particle is calculated by multiplying the six likelihood values for every combination of the three features (color, edge, and height) and the two side (left and right). When the values for a combination become very small *for any particles* due to, for example, a discontinuity of curb or strong cast shadows, that combination is not used for preventing the likelihood values for all particle becoming very small.

D. Resampling step

This step performs a resampling for obtaining a new set of particles. To avoid particle deprivation, we do the resampling only when needed. If the so-called effective number of particles is less than the half of the number of particles, the resampling is performed [8].



(a) cast shadows.

(b) unclear boundaries.

Fig. 5. Estimation results for an unbranched road.

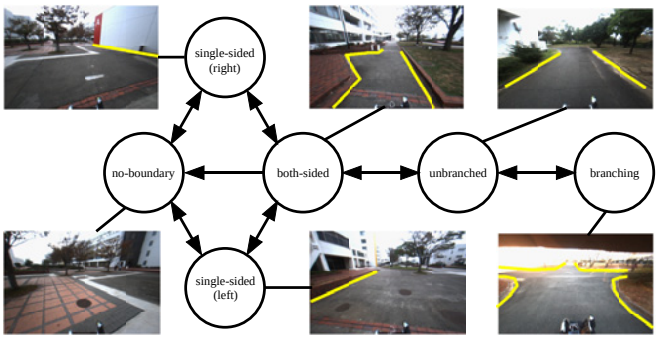


Fig. 6. Road models and transitions.

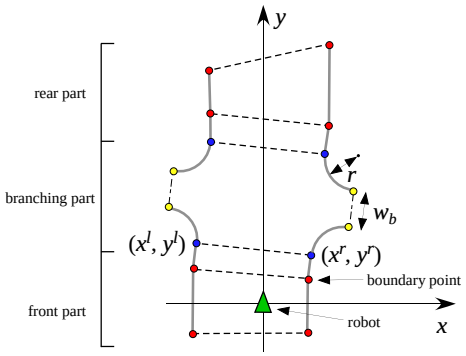


Fig. 7. Branching road model.

E. Estimation result

Fig. 5 shows results of road boundary estimation. For each pair of images, the left one 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 height information, respectively. If a boundary is yellow, for example, it shows that both color and edge information are dominant for the corresponding particle.

The right image is 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.

These figures show that road boundaries are estimated reasonably well for a variety of scenes.

III. Road Models

Our road boundary estimation framework deals with various road scenes by preparing multiple road models. Our previous works [10], [4] used two road models for unbranched and branching roads. We here extend the road models to use six models: unbranched, branching, both-sided, left-sided, right-sided, and no-boundary. Examples of road scenes corresponding to these models are shown in Fig. 6.

A. Branching model

A branching model (see Fig. 7) has a branching part with two road parts on its front and rear sides. The branching part specifies its entry position (x^l, y^l) and (x^r, y^r) , the width w_b of branching road(s), and the radius r of the curved parts. We have three branching road types, crossing, right-T, and left-T, but all types have the same representation; two branching road on the left and the right direction of a crossing are assumed to have the same width.

The representation of branching part is:

$$S_b = \left[x^l, y^l, x^r, y^r, w_b, r\right]. \tag{3}$$

The state vector for the branching road model is given by:

$$X_b = \left[U, w, G, S_1^f, S_2^f, \ldots, S_b, S_1^r, S_2^r, \ldots\right], \tag{4}$$

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 width w_b of the branch. U, w, and G are the same as in eq. (1).

B. Both-sided, single-sided, and no-boundary models

A both-sided road, which is different from unbranched road, is composed of a pair of boundaries with a varying distance between them; no explicit constraints like parallelism exist on the shapes and the positions of the left and the right boundary. Both boundaries are tracked independently.

When the boundary on one side does not exist (i.e., single-sided road), we keep searching for any boundary which may will appear on that side and, at the same time, keep tracking the boundary on the other side. When no boundary exist in the field of view (i.e., open space), we independently search for boundaries on both sides.

We prepare three models, both-sided model, single-sided model, and no-boundary model, for these three road types. To realize tracking or search operations mentioned above, however, all of the three models use the same road representation shown in Fig. 8. Each model has a pair of this single-sided representation but sets the mode of each boundary to the *tracking* mode or the *search* mode depending on the model.

The single-sided representation is composed of a sequence of segments, each of which has front position (x, y), orientation θ, and length h. The number of segments and the segment length depend on the robot speed, the road width, and the field of view and may have a larger distance along the segments than unbranched roads. We currently set the number of segments to nine and the length h is fixed to $1.0\,[m]$ for every segment. The state vector for the single-sided road model is then given by:

$$X_s = [U, G, S_1, S_2, \ldots, S_9], \tag{5}$$

$$S_i = [x_i, y_i, \theta_i], \tag{6}$$

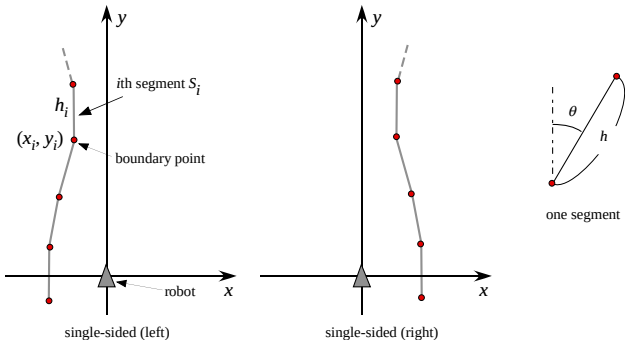


Fig. 8. Single-sided road representation.

where U and G are the same as in eq. (1).

The parameter update is carried out when the robot is judged to pass by the front-most segment; that segment is deleted and a new one is added. In the search mode, the variety of parameters (i.e., the variances of road parameter distributions) are set larger than in the tracking mode. In addition, a part of the particle set (currently, $30\,\%$) are given random parameters within a predetermined range to catch newly-appearing boundaries because they may not have any relationships with the past ones.

IV. Road Model Changes

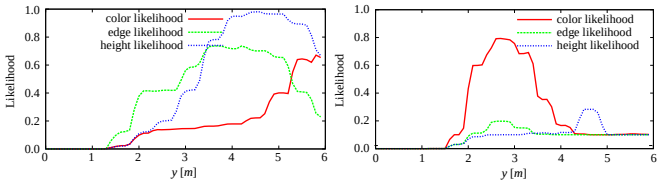
A road model change is a transition from one road model to another. Each road model change is defined by a pair of the transition condition and the transition operation. In a particle filter framework, it could be possible to always test hypotheses for all road models and to see how each of them fits to the current situation. That approach will be, however, costly because for each model, we need to generate multiple hypotheses with many different parameter values (i.e., many particles) for a reliable estimation. We therefore try to find ***signs*** of road model changes and if they exist, we make particles for possible road models those signs suggest. Regardless of the sign being true or false, correct road models are expected to ***survive*** in the subsequent resampling steps.

Edges in the graph shown in Fig. 6 indicate possible transitions between the road models. The transitions are classified into five types. Each transition type will be explained below.

A. Transition by modification

This type of transition happens when an unbranched model changes to a branching or a both-sided model. In such changes, as the robot moves, a branching part or some deviation of road boundaries will gradually visible. We always check how well the current unbranched model fits to the data, and if we find a sign of such changes, we add particles for both models into a particle set.

The sign is detected by examining the ***trends*** of the likelihood values for the three features along the direction of the robot. We calculate their averaged values for all particles and describe them as functions of the distance from the robot. If we find a region where all averaged values are less than some threshold (currently, 0.25 for color and edge features and 0.35 for height features) and whose length is larger than another threshold (currently, $1\,[m]$), we randomly choose a certain number of particles for unbranched roads from the current particle set and modify them to generate branching or both-sided models.



(a) left boundary. (b) right boundary.

Fig. 9. An example of likelihood trend

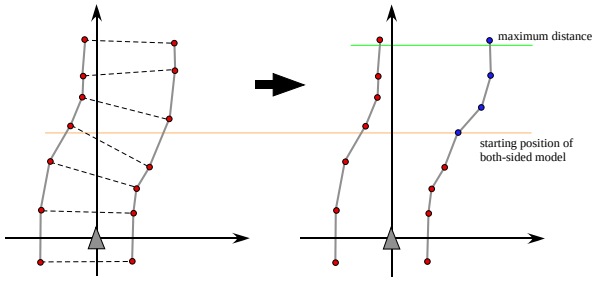


Fig. 10. Transition from unbranched to both-sided.

Fig. 9 shows an example of likelihood trend. Since there is a region with low likelihood values on the right beyond $5\,[m]$, particles for right-T branching and both-sided models are generated. Fig. 10 shows a procedure of modifying an unbranched model into a both-sided one for the a low-likelihood region on the right; a sequence of segments are added beyond the starting position of the region.

B. Transition by temporal consequence

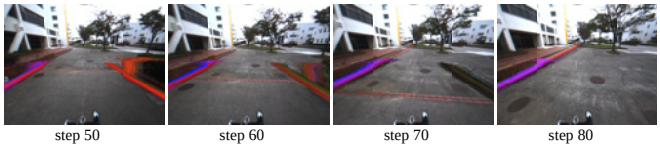
This type of transitions is for those from branching to unbranched models. When the robot is at a crossing, for example, its branching part (see Fig. 7) will soon disappear in the image as the robot moves, and unbranched road segments will then be added, as in the case of Fig. 3(right), and the model will naturally be an unbranched one.

C. Transition by composition

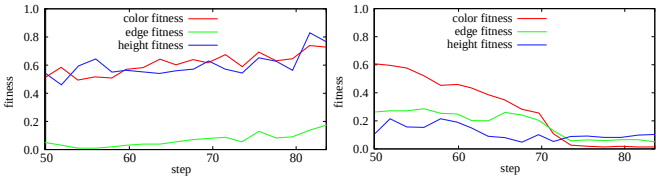
This type of transitions is for those from both-sided to unbranched models. This happens when two single-sided models on both sides are considered almost in parallel. We examine each particle to see if its single-sided models (see Fig. 8) are almost linear and in parallel; a boundary is considered linear if the sum of absolute relative orientations between segments is less than a threshold (currently, $18.0\,[deg.]$). If this condition holds, that particle is changed to an unbranched model and the boundaries on both sides are estimated simultaneously in subsequent steps.

D. Transition by disappearance

This type of transition occurs when a boundary (or a pair of boundaries) disappears by getting out of the field of view, and is for transitions from a both-sided model to a left-sided, a right-sided, or a no-boundary model. Since in the both-sided model, the left and the right boundary are



(a) image sequence.



left boundary. right boundary.

(b) change of fitness values.

Fig. 11. Transition from both-sided to single-sided.

independently estimated as described before, the operation of this transition is to just switch the mode of a single-sided model from *tracking* to *search*.

It is not necessary to always consider this type of transition. We therefore try to find a sign of disappearance, as in the case of *transition by modification*. In transition by modification, we need to analyze the trend to determine the position on the boundary where a branching part begins. In transition by disappearance, however, we need to only judge if a boundary still exists or not. For this purpose, we define a fitness criterion, which indicates how much portion of a road boundary of a particle fits to the current situation. We check every pixel on the mapped boundary in the image to see if the likelihood value for that pixel is effective (i.e., above a threshold) or not. We use the same thresholds for the three likelihood values as above (i.e., 0.25 for color and edge features and 0.35 for height features). The fitness is then defined as the ratio of effective pixels. Finally, if the average of the fitness values of all particles is less than a threshold (currently, 0.2), the current model is considered unfitted, and a mode switching from *tracking* to *search* occurs.

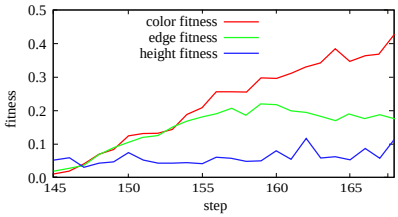
Fig. 11(a) shows a sequence of images for a transition from both-sided to single-sided; we can see that the majority of the particle set changes gradually from both-sided to single-sided. Fig. 11(b) shows changes of the averaged fitness values for the left and the right boundaries. The model is switched from both-sided to single-sided at step 72.

E. Transition by appearance

This type of transition occurs when a boundary on either side appears as it enters the field of view of the camera, and is for transitions from a no-boundary model to a single-sided or a single-sided model to a both-sided. This is an inverse transition of that by disappearance, and if the averaged fitness values for at least one feature becomes larger than a threshold (currently, 0.4), the operation is to switch the mode of a single-sided model from *search* to *tracking*.

Fig. 12(a) shows a sequence of images for a transition from a single-sided model to a both-sided one. At each step, the left image is the estimation result of single-sided models in *track* mode, while the right one in *search* mode. We

(a) image sequence.



(b) change of fitness values.

Fig. 12. Transition from single-sided to both-sided.

can see that distinctive (i.e., brighter) boundaries gradually change from *search* mode to *track* mode, as the robot moves. Fig. 12(c) shows the changes of the averaged fitness values for the right boundary. The model is switched from single-sided to both-sided at step 168.

V. Experimental Results

Fig. 13 shows some of experimental courses in our campus, which have a large variety of shape and appearance. Fig. 14 shows the estimation results for a crossing in course 1. It is shown that both branching and both-sided models are generated at the entry and the former ones then gradually gain the majority.

Fig. 15 shows the results for a place where the road type frequently changes (course 2). From steps 60 to 75, the right boundary got out of the field of view and the transition from both-sided to left single-sided occurred. From steps 105 to 115, the right boundary re-appeared and both-sided models gained the majority. From steps 530 to 575, since the boundaries on both sides became almost in parallel, the transition from both-sided to unbranched occurred. From step 660 to 685, the left boundary disappeared and the transition from both-sided to right single-sided occurred. Fig. 16 shows the change of the portions of models for this scene.

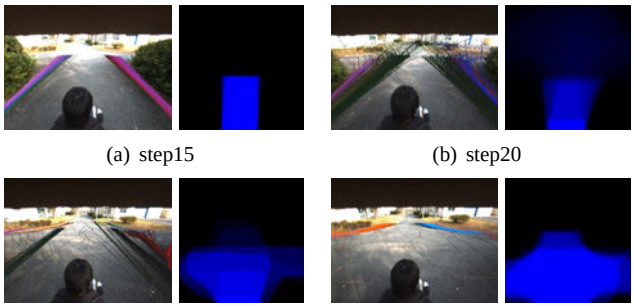
The processing speed is about two frames per second for 600 particles by a Corei7 (2.80GHz, 3GB) laptop PC with a Bumblebee2 stereo camera (Point Grey Co.). We have already realized an autonomous navigation with on-line road boundary estimation in a part of experimental site [4].

VI. Conclusions and Future Work

This paper has described a road boundary estimation method using a stereo camera and multiple road models. By appropriately switching road models, it can robustly and reliably estimate road boundaries in various road scenes including unbranched road, branches, and open spaces. Multi-

(a) course 1 (b) course 2

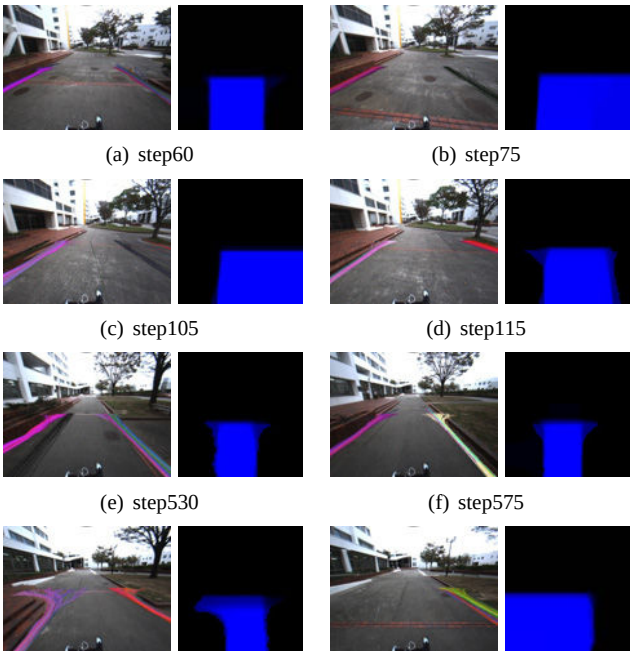
Fig. 13. Experimental courses.



(a) step15 (b) step20

(c) step25 (d) step030

Fig. 14. Estimation of a crossing on course 1.



(a) step60 (b) step75

(c) step105 (d) step115

(e) step530 (f) step575

(g) step660 (h) step685

Fig. 15. Estimation results for varying roads on course 2.

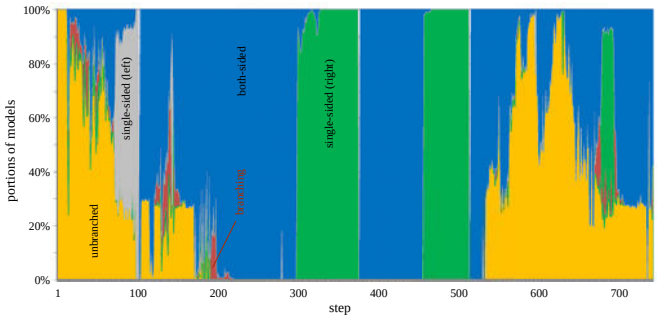


Fig. 16. Changes of portions of models. Colors indicate road models as follows. Yellow: unbranched, blue: both-sided, gray: single-sided (left), green: single-sided (right), and brown: branching.

sensory fusion and road model switching are naturally incorporated in a particle filter framework. We implemented the method on our mobile robot and have done several on-line road boundary estimation and navigation experiments in our university campus where not only well-structured roads but also other spaces to walk through exist.

Although the result is very promising, we need to do more extensive navigation experiments. We also plan to extend the method to cover most of traversable regions in our campus including, for example, slopes. Combination with global localization capabilities such as GPS or view-based localization methods is another future work.

Acknowledgment: The authors would like to thank Dr. Junji Satake for his help in the experimental setup. This work is supported in part by Grant-in-Aid for Scientific Research (No. 21300075) from JSPS.

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] L. Bai, Y. Wang, and M. Fairhurst. Multiple Condensation Filters for Road Detection and Tracking. *Pattern Analysis and Applications*, Vol. 13, No. 3, pp. 251–262, 2010.

[3] 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.

[4] T. Chiku, J. Miura, and J. Satake. Stereo-based Road Boundary Tracking for Mobile Robot Navigation. In *Proceedings of 2011 IEEE Int. Conf. on Robotics and Biomimetics*, pp. 331–336, 2011.

[5] R. Danescu and S. Nedevschi. Probabilistic Lane Tracking in Difficult Road Scenarios using Stereovision. *IEEE Trans. on Intelligent Transportation Systems*, Vol. 10, No. 2, pp. 272–282, 2009.

[6] 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.

[7] M. Montemoerlo et al. Junior: The Stanford Entry in the Urban Challenge. *J. of Field Robotics*, Vol. 25, No. 9, pp. 569–597, 2008.

[8] 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.

[9] K. Maček, B. Williams, S. Kolski, and R. Siegwart. A Lane Detection Vision Module for Driver Assistance. In *Proceedings of IEEE/APS Conf. on Mechatronics and Robotics*, 2004.

[10] 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.

[11] 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.

[12] P. Santana, N. Alves, L. Correia, and J. Barata. Swarm-Based Visual Saliency for Trail Detection. In *Proceedings of 2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 759–765, 2010.

[13] 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.

[14] 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.

[15] 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.