The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Visual Route Navigation using an Adaptive Extension of Rapidly-Exploring Random Trees

Heon-Cheol Lee, Seung-Hwan Lee, Doo-Jin Kim, and Beom-Hee Lee, *Fellow*, *IEEE*

Abstract— **This paper proposes an adaptive and probabilistic extension of Rapidly-exploring Random Tree (RRT) for visual route navigation of a mobile robot. Using measurements from cameras and infrared range sensors, a temporary local map is built probabilistically with Gaussian processes and adaptively to the change of the route curvature. Based on the probabilistic map, RRT searches the most robust and efficient local path with the probability of collision, and the robot is controlled along the selected path. The performance of the proposed method was verified by reducing not only centering error and standard deviation in simulations but also travel time in real experiments.**

I. INTRODUCTION

Visual route navigation is that a robot perceives routes by vision sensors and navigates while avoiding obstacles autonomously, which is performed in different ways according to the following considerations: environments, robot dynamics, hardware system configuration, and motion planning. The environments where a robot navigates highly affect the navigation strategy of the robot, which include routes, illumination, ground status, and obstacles. In unknown environments, at least, the robot has to be given the definition of the navigation route because it affects the types of sensors. If the route is defined by a specific color or the color difference from other environments, vision sensors should be used for the route perception.

After the hardware system configuration is completed according to the environments and robot dynamics, the method to control the robot motion has to be determined. If the navigation task requires very rapid actions for real-time implementation, the reactive control methods are reasonable and powerful because of its low computational cost, which affects the performance of the overall navigation. Especially, in unknown environments, the robot motion for visual route navigation with the reactive control method can be simply compressed into the change of the heading angle of the robot.

This work was supported in part by a Korea Science and Engineering Foundation(KOSEF) NRL Program grant funded by the Korean government (No.R0A-2008-000-20004-0), and in part by the ASRI, and in part by the Brain Korea 21 (BK21) Project, and in part by the Industrial Foundation Technology Development Program of MKE/KEIT [Development of CIRT(Collective Intelligence Robot Technologies)].

Heon-Cheol Lee is an IEEE student member and with the Department of Electrical Engineering, Seoul National University, Seoul, Korea (corresponding author to provide phone: +82-2-880-6487; fax: +82-2-888-4182; restore98@snu.ac.kr).

Seung-Hwan Lee is with the Department of Electrical Engineering, Seoul National University, Seoul, Korea (leeyiri1@snu.ac.kr).

Doo-Jin Kim is with the Department of Electrical Engineering, Seoul National University, Seoul, Korea (djkim00@snu.ac.kr).

Beom-Hee Lee is an IEEE fellow and with Faculty of the Department of Electrical Engineering, Seoul National University, Seoul, Korea (bhlee@snu.ac.kr).

In [1], visual route navigation was performed using a series of reference images and corresponding robot actions such as go straight, turn left and turn right, which is similar strategy used in this work. Thorpe et al. [2] accomplished the roadway following for a car by tracing a white line which is painted along the roadway or roadsides. Their method to line detection is used in this work since it can quickly give a robot the information on correspondence between robot and path. Ohno et al. [3] performed the campus walkway following using the vanishing point of the walkway. However, the strategy to find and move toward vanishing point is not reliable when obstacles on the walkway disturb the vision system. Lee et al. [4] proposed the vision system which consists of two monocular cameras and one stereo camera for two-wheel drive robot, which is different hardware system from this work. Zhang et al. [12] performed the appearance based visual route following for navigation in outdoor environments using an omnidirectional camera and sonar range sensors. They showed robust performance in the condition that there was no unknown obstacle on the route.

Fulgenzi et al. [5] proposed a navigation algorithm which integrates perception uncertainty and incompleteness in the path planning strategy using a probabilistic framework such as Rapidly-exploring Random Tree (RRT) and Gaussian Process (GP). On the basis of the local occupancy grid map, the fusion of the two methods improved the performance of motion planning in the environments with moving obstacles. However, off-line process was needed to learn the pattern of the moving obstacles, and the constraints of the environments are different from our work.

In this paper, an adaptive and probabilistic extension of RRT (APRRT) is proposed to improve the performance of visual route navigation with the reactive control method. Differently from [5], the proposed method models probabilistically the local space considering constraint lines changed adaptively to the curvature as well as obstacles. Besides, the modeled local map is built by not only range-based measurements but also visual-based measurements. The system for the proposed method consists of three cameras for the perception of the route and infrared range sensors for obstacle avoidance.

This paper is organized as follows: Section II describes visual route navigation system, and the problem of the inaccurate heading angle of a robot is stated. Section III describes the proposed method to solve the problem. The improved results are presented by computer simulations in Section IV and real experiments in Section V. Finally, Section VI gives conclusion.

II. Visual Route Navigation

In this section, visual route navigation system with the reactive control method is briefly described with the problem of the inaccurate heading angle of a robot.

A. Hardware System Configuration

A robot used in this work is IRON-C which was produced by Redone Technology. The robot has four-wheel drive and skid differential steering system, and its maximum translation velocity is 0.36m/s. Nine infrared range sensors whose maximum detecting range of 160mm are equipped at the front of the robot. Three web cameras are equipped to measure the distance and angle between the robot and routes because the color of the route is known. As shown in Fig. 1, the two cameras are equipped at the robots both sides, respectively, and the other camera is equipped at the front of the robot. Finally, for image processing and robot control, a lab-top computer is equipped at the robots rear side.

B. Perception and Measurements for Routes

Routes were perceived by extracting the lines as forward edges and both side edges of routes whose color was known, which is a similar strategy with [2]. To overcome the illumination problem and obtain reliable line features, a lot of image processing techniques are performed as shown in Fig. 2, and the results of the line feature extraction at the route curvature of 30° are shown in Fig. 3.

The extracted line features provide the robot with the measurements of distance and angle between robot and routes. The angle measurement, θ_{route}, is easily obtained by computing the angle between the extracted line feature and horizontal line on image since the both-sides cameras are equipped perpendicular to the heading of the robot as shown in Fig. 3(b). A single image, however, cannot give the distance measurement, d_{route}, from the camera to the line. Also, the general method to obtain the distance such as feature matching requires much computation time, which is not appropriate for rapid robot control. Thus, virtual distance function which approximately represents the relation between the real distance from the robot to the line and the corresponding height of the image is used as shown in Fig. 3(c). Because the orientation of each camera is fixed, the real distance can be matched to the corresponding heights of the image. Then, the four-order polynomial function is obtained

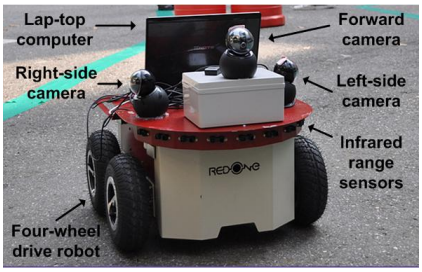


Fig. 1. Hardware system configuration. A camera for front-view, two cameras for both side-view, and infrared range sensors for detecting obstacles are equipped on the four-wheel drive mobile robot.

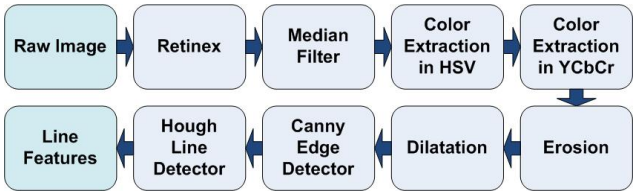
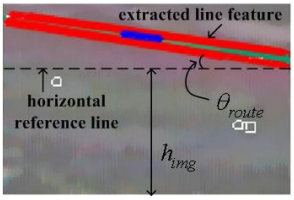
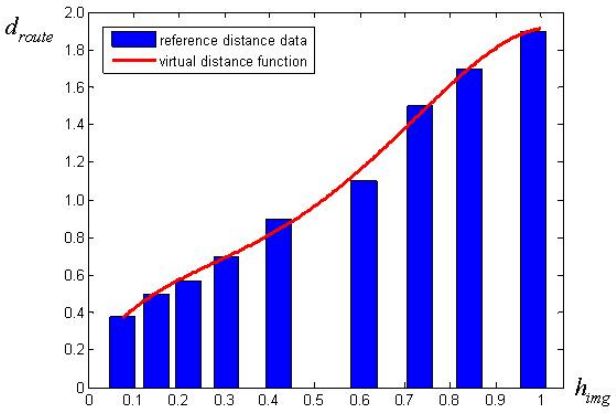


Fig. 2. The process of the color-based line feature extraction. Two types of color spaces are used, and Retinex algorithm is used to reduce the illumination effects. Median filter, erosion, and dilatation are used to reduce noises. Canny edge and Hough line detector are used to extract line features.

(a) Raw image at 30°



(b) Extracted line features



(c) Virtual distance function obtained from reference distance data

Fig. 3. Route perception and depth measurements at the curvature of 30°. To measure the distance (d_{route}) from the extracted line features, the virtual distance function is obtained by matching and interpolating the reference distance data and the height of line features on the image (h_{img}).

by mapping and interpolating the matched reference distance data as follows:

$$d_{route}(h_{img}) = c_4 h_{img}^4 + c_3 h_{img}^3 + c_2 h_{img}^2 + c_1 h_{img} + c_0 \quad (1)$$

where h_{img} is the height of the line feature, $c_4 = -686.6$, $c_3 = 1381.0$, $c_2 = -839.8$, $c_1 = 319.3$, and $c_0 = 17.2$.

C. Problem of the Inaccurate Heading Angle of a Robot

In visual route navigation with the reactive control method, the robot performs obstacle avoidance reactively as used in [4], [12]. Therefore, the robot motion control can be compressed into the change of the heading angle of a robot at every control time step. However, it is difficult to calculate the accurate heading angle due to the followings: limitation on the range of infrared sensors, limitation on the field of view of cameras, and inevitable errors in the measurements. In this paper, APRRT is proposed to make visual route navigation with the reactive control method more robust, and the overall structure of visual route navigation

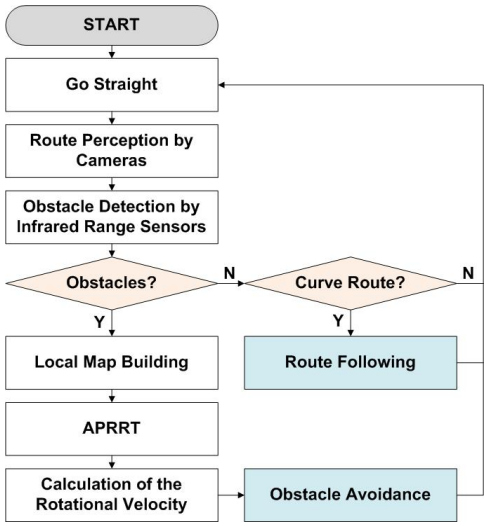


Fig. 4. Overall navigation structure. Route following and obstacle avoidance are performed by the reactive control. The rotational velocity of the robot is calculated by APRRT.

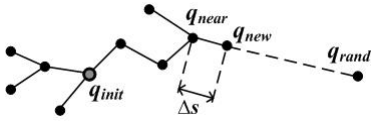


Fig. 5. Rapidly-Exploring Random Tree (RRT). A new configuration q_{new} is obtained applying an admissible control from q_{near} toward q_{rand}.

using the proposed method is shown in Fig. 4. The robot goes straight initially and changes its rotational velocity reflectively according to the variation of distance and angles to the route. If obstacles are detected, a temporary local map is built using measurements from the cameras and range sensors. Then, APRRT is performed to calculate the rotational velocity of the robot for obstacle avoidance.

III. Proposed Method

In this section, a single-query planning algorithm, Rapidly-exploring Random Tree (RRT) is introduced, and its probabilistic extension, probabilistic RRT (PRRT), is reviewed. Then, adaptive PRRT (APRRT) which is another extension of RRT using adaptive and probabilistic strategies for various types of curves is proposed.

A. Rapidly-Exploring Random Tree (RRT)

RRT [9] searches both convex and nonconvex high-dimensional spaces, which is an incrementally constructed tree which can reduce the expected distance of a randomly-chosen point to the tree compared with other expansive-spaces trees. The algorithm randomly chooses a configuration q_{rand} in the configuration space and tries to extend the current search tree toward the chosen configuration. Then, the nearest configuration q_{near} in the original tree T is chosen for extension as shown in Fig. 5. A new configuration q_{new} is obtained applying an admissible control from q_{near} toward q_{rand} by a predetermined step size Δs. If q_{new} is collision-free, it is added to T. The algorithm can be stopped once the goal is found, or it can keep on running to find a better path. Once the goal state is reached, the path from the initial state to the goal is retrieved.

B. Probabilistic RRT (PRRT)

RRT needs a local map which requires very accurate measurements from sensors because it lies on a deterministic representation of the environments. However, in real environments, it is difficult to build a deterministic local map because the measurements contain inevitable errors. Fulgenzi et al. [5] proposed PRRT which integrates perception uncertainty and incompleteness in the path planning strategy using probabilistic frameworks. In their work, GPs which could represent paths as continuous functions in a probabilistic manner were used to overcome the disadvantage of discretization of state spaces. But, this improvement required off-line processes to learn the pattern of moving obstacles because they assumed the moving obstacles had typical patterns. Since K components correspond to each two dimensional state, the likelihood of the state based on the N training data is as follows:

$$P(x|z,\mu_x,\theta) = \prod_{n=1}^{N}\prod_{k=1}^{K} G(x_n|\mu_{x,k}, C(\theta_k))^{Z_{n,k}} \quad (2)$$

$$P(y|z,\mu_y,\theta) = \prod_{n=1}^{N}\prod_{k=1}^{K} G(y_n|\mu_{y,k}, C(\theta_k))^{Z_{n,k}} \quad (3)$$

where z is the vector of component weights, and θ is the GP hyper-parameter, and μ is the mean function of GP, and $C(\theta_k)$ is the covariance matrix of GP parameterized by θ.

Considering a candidate path with J states, M moving obstacles, and K the number of GPs associated to the typical patterns, for an obstacle O_m at the state s_j, the associated probability of collision $P_{cd}(s_j, m)$ is calculated by overlapping of the robot and the combination of the two likelihoods. If each component k is considered seperately, the probability of collision can be rewritten by $P_{cd}(s_j, m, k)$. Then, for the path $\pi(s_J) = [s_0, s_1, \ldots, s_J]$, a single obstacle m, and the k-th GP, the probability of collision is calculated by

$$P_\pi(s_J, m, k) = 1 - \prod_{j=0}^{J}(1 - P_{cd}(s_j, m, k)) \quad (4)$$

Finally, the probability of success that the robot traverses the path without colliding obstacles, which means the probability of *not* having collision, is calculated by

$$L_\pi(s_J) = \prod_{m=1}^{M}(1 - (\sum_{k=1}^{K} l_{k,m} \cdot P_\pi(s_J, m, k))) \quad (5)$$

where $l_{k,m}$ is a predetermined constant of the k-th GP for the obstacle m. The collision-free path is selected based on the probability of success, and the robot is controlled according to the selected path. More detailed description and formulas are in [5].

C. Adaptive and Probabilistic RRT (APRRT)

The environments in this work are different from PRRT [5] in the context of the followings: static obstacles, lack of sensibility, and routes restricted by two lines. The first difference makes the problem easier. But, the other differences make the problem more difficult because the lack causes errors in calculation of the heading angle and the restriction requires the faster computation process. In this paper, APRRT which is an adaptive and probabilistic extension of RRT is proposed to calculate the heading angle more rapidly and accurately. Differently from PRRT [5], APRRT models the local space considering constraint lines as well as obstacles, probabilistically. Besides, the constraint lines are changed adaptively to the curvature, and the modeled local map is built by not only range-based measurements but visual-based measurements.

For the constraint line i at the state s_j with the k-th GP, the probability of collision for the constraint lines at time t is calculated by

$$P_{ci}(s_j, i, k, t) = G(s_j | \mu_{i,\mathbf{x},k}(t), C_I(\theta_k, t)) \tag{6}$$

where $\mathbf{x} = [x, y]^T$, and $\mu_{i,\mathbf{x},k}(t)$ and $C_I(\theta_k, t)$ are the mean and covariance matrix. The mean of the constraint line i is adaptively changed according to the curvature of the perceived route as follows:

$$\mu_{i,\mathbf{x},k}(t) = \lambda_s(i,t)(d_s(i,t) - \gamma_s(i,\kappa,t)) \tag{7}$$

where if it is the left constraint line, $i = 1$, otherwise $i = 2$. $\lambda_s(i,t) = -1$ if $i = 1$, otherwise $\lambda_s(i,t) = 1$. $d_s(i,t)$ is the measurement of distance between the robot and the left line if $i = 1$, otherwise $d_s(i,t)$ is the measurement of distance between the robot and the right line. Finally, $\gamma_s(i,\kappa,t)$ is the adpative coefficient which is a function of i and the curvature κ, which is predetermined according to all the possible curvature κ for each line i. This adaptive change of the mean yields the more robust and efficient paths for route navigation than the paths produced by RRT or PRRT.

For the path $\pi(s_J) = [s_0, s_1, \ldots, s_J]$ at time t, a single obstacle m, a constraint line i, and the k-th GP, the probability of collision with obstacles, $P_{co}(\cdot)$, and the probability of collision with the constraint lines, $P_{ci}(\cdot)$, are respectively calculated similarly with [5] as follows:

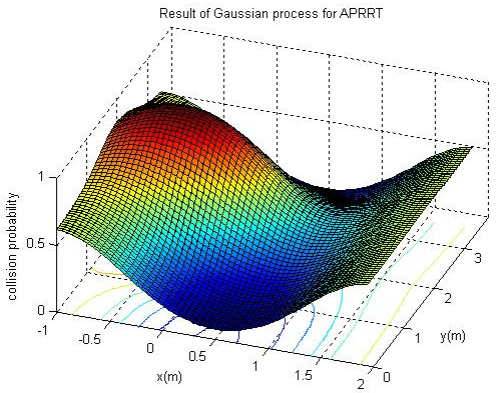
$$P_{\pi o}(s_J, m, k, t) = 1 - \prod_{j=0}^{J}(1 - P_{co}(s_j, m, k, t)) \tag{8}$$

$$P_{\pi i}(s_J, i, k, t) = 1 - \prod_{j=0}^{J}(1 - P_{ci}(s_j, i, k, t)) \tag{9}$$

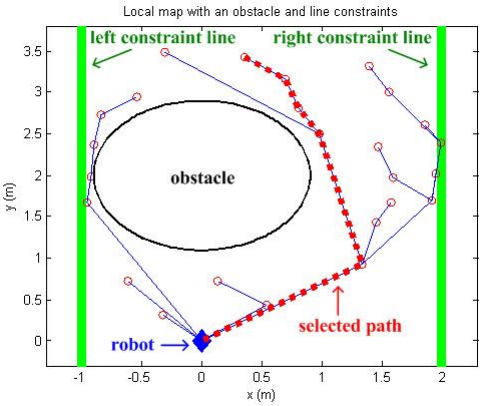
The local space is represented by the probability of collision with obstacles and constraint lines as shown in Fig. 6(a).

Then, the probability of success at time t that the robot traverses the path without colliding obstacles, $L_{\pi o}(s_J, t)$, and the probability of success that the robot traverses the path without crossing the constraint lines, $L_{\pi i}(s_J, t)$, are respectively calculated by

$$L_{\pi o}(s_J, t) = \prod_{m=1}^{M}(1 - (\sum_{k=1}^{K} l_{k,m} \cdot P_{\pi o}(s_J, m, k, t))) \tag{10}$$



(a) The result of Gaussian processes



(b) Temporary local map for APRRT

Fig. 6. Adaptive and Probabilistic RRT (APRRT). The temporary local map is built by Gaussian processes in configuration space. APRRT is performed in the map and selects the most reliable path.

$$L_{\pi i}(s_J, t) = \prod_{i=1}^{I}(1 - (\sum_{k=1}^{K} l_{k,i} \cdot P_{\pi i}(s_J, i, k, t))) \tag{11}$$

where $l_{k,m}$ and $l_{k,i}$ are predetermined constants of the k-th GP for the obstacle m and the constraint line i, respectively. Using the two probabilities, the total probability of success is at time t calculated by

$$L_{\pi}(s_J, t) = L_{\pi o}(s_J, t) \cdot L_{\pi i}(s_J, t) \tag{12}$$

Finally, the most reliable path is selected by the probability of success as shown in Fig. 6(b). The robot is controlled along the selected path using the more robust and efficient rotational velocity at time t, $w(t)$, obtained by the following equation:

$$w(t) = \alpha(t)(\frac{\beta(t) \cdot \theta_{IR}(t)(w_{\max} - w_{\min})}{\pi} - w_{\max}) \tag{13}$$

where $\alpha(t)$ and $\beta(t)$ are the direction and amount between the robot orientation and the edge from the state t-1 to the state t, respectively. If $\alpha(t) = -1$, the robot turns left, and if $\alpha(t) = 1$, the robot turns right, if $\alpha(t) = 0$, the robot goes straight. $\theta_{IR}(t)$ is the angle measurement from the robot to the obstacle by infrared range sensors. $w_{\max}$ and $w_{\min}$ are the maximum and minimum rotational velocities, respectively.

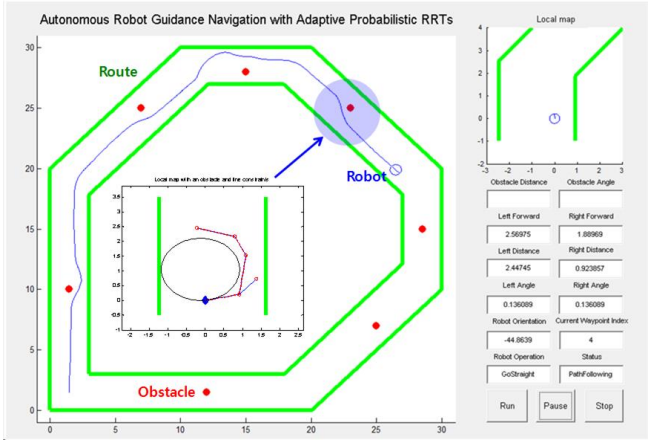


Fig. 7. The simulator for autonomous robot route navigation with APRRT. The size of the environments was 30m by 30m. The route of 3m width was defined by two lines, and there were six curves and seven obstacles represented by red circles.

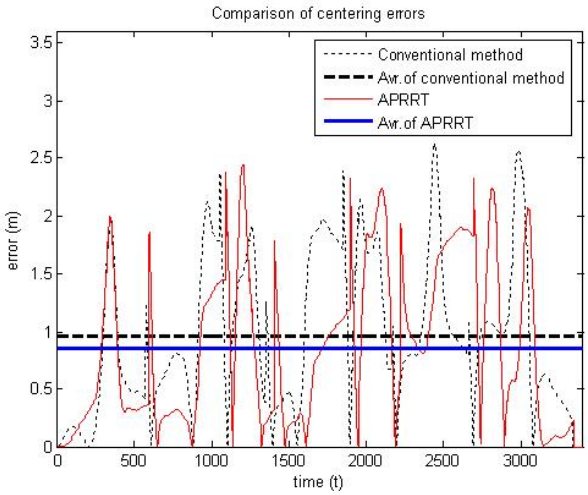


Fig. 8. Comparison of the centering errors. The centering errors of APRRT were compared with those of the conventional method. The dotted black line and the thin red line represent the centering error of conventional method and APRRT, respectively. The dashed thick black line and the thick blue line represent the averages of conventional method and APRRT, respectively.

TABLE I

TOTAL COMPARISON OF THE PERFORMANCE IN SIMULATIONS

	Conventional mtd.	APRRT (proposed.)
Total centering error (m)	0.9639	0.8561
Standard deviation (m)	1.1351	1.0542

IV. SIMULATIONS

A. Simulator Description

The simulator for autonomous robot route navigation with APRRT was made by MATLAB7.0 as shown in Fig. 7. The robot which had the width of 0.5m and the length of 0.71m was represented by a blue circle, and its local map was shown at the top-right side in the figure. To realistically perform simulations, the errors in robot motion and measurements were carefully modeled because they are subject to noise in reality, are modeled by a zero-centered random variable with finite variance [8] as follows:

$$\begin{pmatrix}\hat{v}\\ \hat{w}\end{pmatrix} = \begin{pmatrix}v\\ w\end{pmatrix} + \begin{pmatrix}\sigma_{\epsilon_1|v|+\epsilon_2|w|}\\ \sigma_{\epsilon_3|v|+\epsilon_4|w|}\end{pmatrix} \tag{14}$$

$$\begin{pmatrix}\hat{d}\\ \hat{\theta}\end{pmatrix} = \begin{pmatrix}d\\ \theta\end{pmatrix} + \begin{pmatrix}\sigma_{\epsilon_5|v|+\epsilon_6|w|}\\ \sigma_{\epsilon_7|v|+\epsilon_8|w|}\end{pmatrix} \tag{15}$$

where v is the ideal translational velocity of the robot whose maximum translational velocity was 0.6m/s, and w are the ideal rotational velocity of the robot whose maximum rotational velocity was 0.2rad/s. $\hat{v}$ and $\hat{w}$ are the erratic translational and rotational velocities of the robot, respectively. d and θ are the ideal distance and angle measurements, respectively. $\hat{d}$ and $\hat{\theta}$ are their erratic measurements, respectively. σ_b is a zero-centered random variable with standard deviation b. The constants from ϵ_1 to ϵ_4 are robot-specific parameters that represent the accuracy of the robot motion. The less accurately the robot moves, the larger the parameters are. The constants from ϵ_5 to ϵ_8 are sensor-specific parameters. The maximum detectable distance of infrared range sensors was set by 1.4m, and their detectable range of angles was set from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$.

B. Simulation Results

The robot completed successfully the navigation task various with APRRTs. Among randomly sampled fifty configurations, the number of configurations which were components of the final trees was below ten in almost all time steps. It means the configuration space was too constrained by obstacles and line features. The excessive constrained local map was caused by the errors in line measurements since the errors affected the mean of the constraint line which was an component of the local map by (7).

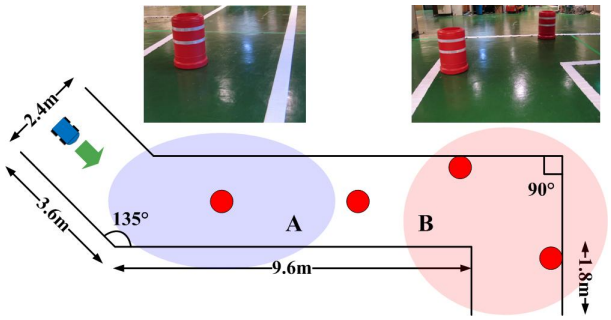
The robot motion affects the errors in line measurements by (15). If the robot changes its own orientation frequently or excessively, the variance of the error model becomes larger. This means the probability that the robot acquires bad measurements becomes larger. If the robot navigates at the center of the route, the probability of bad measurements acquisition becomes smaller. Thus, for the performance evaluation, the centering error at time t was defined by

$$e_c(t) = |d_s(1,t) - d_s(2,t)| \tag{16}$$

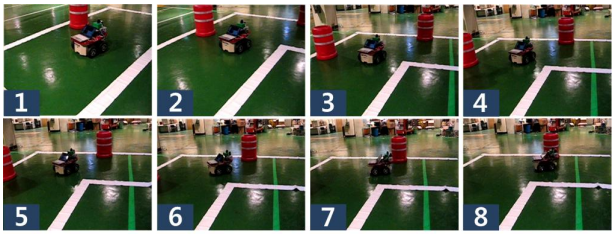
In Fig. 8, the centering error at every time step t was compared with the conventional method which is the previous naive method of fully reflective control based navigation. All the results were averaged from ten simulations including the worst and best cases. At some time steps, the difference between the two methods was not clear, but the total centering error and the standard deviation of APRRT were smaller than those of the conventional method as shown in Table I.

V. EXPERIMENTS

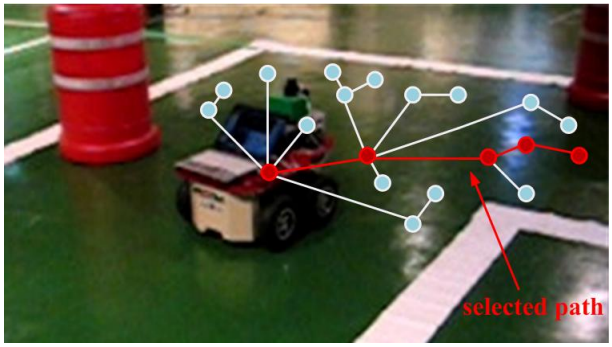
For the more realistic verification of the robustness and efficiency of APRRT, real experiments were conducted in the 15m route as shown in Fig. 9(a). Based on the knowledge of



(a) Dimensions of the route for experiments



(b) Snapshots of the navigation in the region B



(c) The result of APRRT at the snapshot 2 in the region B

Fig. 9. Experiments in the 15m route whose width was 2.4m. (a) Red small circles represent obstacles, and the only information given to the robot was the color of the route. (b) The robot navigated the unknown route successfully even in the abrupt curve with APRRT. (c) The path selected by APRRT gave the more robust and efficient rotational velocity to the robot.

the color of the route, the robot autonomously navigated the route whose width was 2.4m. The perception of the routes has suffered from the bad illumination status even though the Retinex algorithm as shown in Fig. 2 was applied for solving the problem, which affected the overall performance of the navigation. The route color perceived best was white in the environments. The abrupt curve with the curve angle of 90° was a challenging situation for the route navigation of the reactively controlled robot. APRRT enabled the robot to overcome the difficult situation at the speed of 0.3m/s as shown in Fig. 9(b), and the result of APRRT at the time is shown in Fig. 9(c). Among randomly sampled 50 nodes, 19 nodes were used to build the tree, and the only 5 nodes were selected for the calculation of the heading angle of the robot.

The accurate ground truth which is a general index for error analysis is difficult to be obtained in real experiments. Real-time kinematic GPS used by Royer et al. [7] was considered but inapplicable due to buildings occluding GPS signals. Therefore, the chosen method to evaluate the performance was the travel time from losing the center of the route to recovering it. The comparison results between the conventional method and APRRT in the region A and B are summarized in Table II. The travel time was reduced by APRRT in both regions, and the difference was noticeable in the region B which is the abrupt curve with the curvature of 90°. Consequently, the performance of visual route navigation was improved by the proposed method, APRRT.

TABLE II

COMPARISON OF THE TRAVEL TIME IN EXPERIMENTS

	Conventional method	APRRT (proposed method)
Region A	23.55 (s)	21.44 (s)
Region B	31.23 (s)	22.98 (s)

VI. CONCLUSION

Visual route navigation with the reactive control method has the problem of the inaccurate heading angle of the robot due to the limitation on sensibility and the inevitable errors in measurements. In this work, APRRT which uses adaptive and probabilistic methods was proposed to solve the problem, and it enables the robot to perform the route navigation task more efficiently and robustly. Its improved performance was verified by reducing centering error, standard deviation, and travel time in computer simulations and real experiments.

REFERENCES

[1] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual Navigation Using View Sequenced Route Representation," *Proc.* IEEE *Int. Conf.* Robotics and Automation, 1996.

[2] K. Kluge and C. Thorpe, "Representation and Recovery of Road Geometry in YARF," *Proc.* Intelligent Vehicles, 1992.

[3] K. Ohno and T. Tsubouchi, "A Mobile Robot Campus Walkway Following with Daylight-Change-Proof Walkway Color Image Segmentation," *Proc.* IEEE /RSJ *Int. Conf.* Intelligent Robots and Systems, 2001.

[4] Heon C. Lee, Tae S. Lee, Seung H. Lee, Gyu H. Eoh, and Beom H. Lee, "Visual Path Following and Obstacle Avoidance Using Multiple Cameras for Outdoor Environments," *Proc. Int. Conf.* Ubiquitous Robots and Ambient Intelligence, 2009.

[5] C. Fulgenzi, C. Tay, A. Spalanzani, and C. Laugier, "Probabilistic Navigation in Dynamic Environment using Rapidly-Exploring Random Trees and Gaussian Processes," *Proc.* IEEE/RSJ *Int. Conf.* Intelligent Robots and Systems, 2008.

[6] K. Macek, R. Philippsen, and R. Siegwart, "Path Following for Autonomous Vehicle Navigation Based on Kinodynamics Control," *Journal of Computing and Information Technology*, vol. 17, no. 1, pp. 17-26, 2009.

[7] E. Royer, J. Bom, M. Dhome, B. Thuilot, M. Lhuillier, and F. Marmoiton, "Outdoor autonomous navigation using monocular vision," *Proc.* IEEE /RSJ *Int. Conf.* Intelligent Robots and Systems, 2005.

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

[9] S. LaValle and J. Kuffner, "Randomized Kinodynamic Planning," *Proc.* IEEE *Int. Conf.* Robotics and Automation, 1999.

[10] P. Missiuro and N. Roy, "Adapting Probabilistic Roadmaps to Handle Uncertain Maps," *Proc.* IEEE *Int. Conf.* Robotics and Automation, 2006.

[11] V. Subramanian, T. F. Burks, and W. E. Dixon, "Sensor Fusion Using Fuzzy Logic Enhanced Kalman Filter for Autonomous Vehicle in Citrus Groves," *IEEE Trans. ASABE*, vol. 52, no. 5, pp. 1141-1422, 2009.

[12] A. M. Zhang and L. Kleeman, "Robust Appearance Based Visual Route Following for Navigation in Large-scale Outdoor," *The International Journal of Robotics Research*, vol. 28, no. 3, pp. 331-356, 2009.