Local Minimum Free Motion Planning for Mobile Robots within Dynamic Environmetns

Jong Suk Choi*, Munsang Kim**, and Chong-won Lee***

All authors are at the Intelligent Robotics Research Center, KIST, Seoul 131, Korea
*(Tel: +82-2-958-5618; Fax: +82-2-958-5629; Email:cjs@kist.re.kr)
**(Tel: +82-2-958-5623; Fax: +82-2-958-5629; Email:munsang@kist.re.kr)
***(Tel: +82-2-958-5611; Fax: +82-2-958-5629; Email:cwlee@kist.re.kr)

Abstract: We build a local minimum free motion planning for mobile robots considering dynamic environments by simple sensor fusion assuming that there are unknown obstacles which can be detected only partially at a time by proximity sensors and can be cleaned up or moved slowly (dynamic environments). Potential field is used as a basic platform for the motion planning. To clear local minimum problem, the partial information on the obstacles should be memorized and integrated effectively. Sets of linked line segments (SLLS) are proposed as the integration method. Then robot's target point is replaced by *virtual target* considering the integrated sensing information. As for the main proximity sensors, we use laser slit emission and simple web camera since the system gives more continuous data information. Also, we use ultrasonic sensors as the auxiliary sensors for simple sensor fusion considering the advantages in that they give exact information about the presence of any obstacle within certain range. By using this sensor fusion, the dynamic environments can be dealt easily. The performance of our algorithm is validated via simulations and experiments.

Keywords: Mobile robot; motion planning; potential-field.

1. Introduction

Nowadays, map-building for robots' navigation is performed by hybrid methods of global topological map and local metric map [1], [2] since our environments are becoming too complex to be presented simply. Therefore corresponding navigation methods are divided into two phases: First is target generation that generate a target point (or node) for robots to go after investigating pre-known (topological) map of environments. The next is local motion planning that makes the robots go the target avoiding obstacles in the metric map. In most cases, however, unknown obstacles are existent and can be detected only partially at a time due to the limitation of sensing capability, which may causes failure in the second phases of the navigation.

We build a local minimum free motion planning for mobile robots considering dynamic environments by simple sensor fusion. There are three constraints assumed: i) there are unknown obstacles in our environment, ii) they can be detected only partially at a time by proximity sensors, iii) and they can be cleaned up or moved slowly (dynamic environments). Potential field is used as a basic platform for the motion planning since it has the advantages of simplicity, real-time computation. However, there is one shortcoming in the potential filed: it may cause local minimum whenever the curvature of the repulsive equipotential curve is less than the curvature of the attractive equipotential curve at the same configuration. Also, since the robot has only proximity sensors it may go back and forth repeatedly due to lack of information about the entire shape of unknown obstacles, then get into local minimum. Elliptical repulsive potentials have been developed [3], [4] to reduce the number and the size of the local minima of the potential field while having no exact rejection of the local minima. To cope with this problem, randomized path planner [5] has been proposed while the randomized planning has drawbacks in that the running time varies and the planner typically generates different paths if it is performed several times for the same problem. Grid potential with navigation function [6] is a good method even in arbitrary geometry of workspace, however, its computational complexity is highly dependent on the area and the grid's resolution.

As for the above previous researches, there remain limitations in removing the local minimum problem. Also, if the robot has only proximity sensors it may go back and forth repeatedly due to lack of information about the entire shape of unknown obstacles, then get into local minimum. To clear the problem, the partial information on the obstacles should be memorized and integrated effectively. In this paper, sets of linked line segments (SLLS) are proposed as the integration method. Then robot's target point is replaced by *virtual target* considering the integrated sensing information. Before getting to the virtual target point is created a reference point meaning the nearest one to the start point in the nodal points which are lying on the shortest path from start point to target point. At last, the virtual target is generated on tangential point between a circle centered at the reference point and a line beginning from the start point. Note that the virtual target is not fixed but changed continuously through the circular arc with regard to change of the start point. As a result, the robot can keep the continuous motion. As for the main proximity sensors, we use laser slit emission and simple web camera since the system gives more continuous data information, thus enables us to catch a line segment easily. However, the laser system is sensitive to light condition or obstacle's color, which may make non-existent obstacles or miss real existent obstacles as a result. We use ultrasonic sensors as the auxiliary sensors for simple sensor fusion considering the advantages in that they give exact information about the presence of any obstacle within certain range. By using this sensor fusion, the dynamic environments can be dealt easily. The performance of our algorithm is validated via simulations and experiments.

2. Problem statement

Let the configuration of a point be

$$\mathbf{q} = [x \ y]^T \tag{1}$$

We will represent the start and target points of robot as $\mathbf{q}_s = [x_s \ y_s]^T$ and $\mathbf{q}_t = [x_t \ y_t]^T$, and a point of obstacles as $\mathbf{q}_o = [x_o \ y_o]^T$.

2.1. Problem Statement

Problem 1: Given start position $\mathbf{q}_s$ and target position $\mathbf{q}_t$, find the desired linear and angular velocities, v^d and w^d, for the robot to reach the target position with obstacle avoidance where the set of obstacles can be detected only partially at a time by proximity sensor.

2.2. Potential Field

Potential function used here is constructed as the sum of attractive potential associated with the target configuration and repulsive potential associated with the obstacle region:

$$U(\mathbf{q}) = U_{att}(\mathbf{q}) + U_{rep}(\mathbf{q}) \tag{2}$$

Each potential is simply defined as

$$\begin{array}{rcl} U_{att}(\mathbf{q}) & = & \frac{1}{2}\xi\rho_t^2(\mathbf{q}) \\ U_{rep}(\mathbf{q}) & = & \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0}\right)^2 \rho_t^2(\mathbf{q}), & \text{if } \rho(\mathbf{q}) \leq \rho_0 \\ 0, & \text{if } \rho(\mathbf{q}) > \rho_0 \end{cases} \end{array} \tag{3}$$

where

$$\rho_t(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_t\|, \quad \rho(\mathbf{q}) = \|\mathbf{q} - \mathbf{q}_o\|$$

Note the last part $\rho_t^2(\mathbf{q})$ of the U_{rep} is add for the case where the obstacles are near to the target point [7]. The field of artificial forces is produced from a gradient of the potential function as

$$\begin{array}{rcl} \overrightarrow{F}(\mathbf{q}) & = & -\overrightarrow{\nabla}U(\mathbf{q}) \\ & = & \overrightarrow{F}_{att}(\mathbf{q}) + \overrightarrow{F}_{rep}(\mathbf{q}) \end{array} \tag{4}$$

where the attractive force is

$$\begin{array}{rcl} \overrightarrow{F}_{att}(\mathbf{q}) & = & -\overrightarrow{\nabla}U_{att}(\mathbf{q}) \\ & = & -\overrightarrow{\nabla}\frac{1}{2}\xi\rho_t^2(\mathbf{q}) \\ & = & -\xi\rho_t(\mathbf{q})\overrightarrow{\nabla}\rho_t(\mathbf{q}) \\ & = & -\xi\rho_t(\mathbf{q})\overrightarrow{\nabla}\left[(x-x_t)^2 + (y-y_t)^2\right]^{\frac{1}{2}} \\ & = & -\xi\rho_t(\mathbf{q})\left[\frac{(x-x_t)}{\rho_t(\mathbf{q})}\hat{x} + \frac{(y-y_t)}{\rho_t(\mathbf{q})}\hat{y}\right] \end{array} \tag{5}$$

and where the repulsive force is

$$\begin{array}{rcl} \overrightarrow{F}_{req}(\mathbf{q}) & = & -\overrightarrow{\nabla}U_{rep}(\mathbf{q}) \\ & = & -\overrightarrow{\nabla}\frac{1}{2}\eta\left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0}\right)^2 \rho_t^2(\mathbf{q}) \\ & = & \eta\left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0}\right)\frac{\rho_t^2(\mathbf{q})}{\rho^2(\mathbf{q})}\overrightarrow{\nabla}\rho(\mathbf{q}) \\ & & -\eta\left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0}\right)^2 \rho_t(\mathbf{q})\overrightarrow{\nabla}\rho_t(\mathbf{q}) \\ & = & \eta\left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0}\right)\frac{\rho_t^2(\mathbf{q})}{\rho^2(\mathbf{q})}\left[\frac{(x-x_o)}{\rho(\mathbf{q})}\hat{x} + \frac{(y-y_o)}{\rho(\mathbf{q})}\hat{y}\right] \\ & & -\eta\left(\frac{1}{\rho(\mathbf{q})} - \frac{1}{\rho_0}\right)^2 \rho_t(\mathbf{q})\left[\frac{(x-x_t)}{\rho_t(\mathbf{q})}\hat{x} + \frac{(y-y_t)}{\rho_t(\mathbf{q})}\hat{y}\right] \end{array} \tag{6}$$

Then, we produce the desired linear and angular velocities, v^d and w^d, from a projection of $\overrightarrow{F}(\mathbf{q})$ into the direction of robot's heading angle θ:

$$\begin{array}{rcl} v^d & = & k_v\|\overrightarrow{F}(\mathbf{q})\|\cos\phi \\ w^d & = & k_w\|\overrightarrow{F}(\mathbf{q})\|\sin\phi \end{array} \tag{7}$$

where

$$\phi = \angle\overrightarrow{F}(\mathbf{q}) - \theta$$

3. Local motion planning

For the expression of obstacles, we will build set of line segments $L = \{L_i\}$ by memorizing the partial (point-based) information from proximity sensors where each line segment L_i is composed of two extreme points, $\mathbf{q}_1^{L_i}$ and $\mathbf{q}_2^{L_i}$ (Fig. 1). Since, real information about the obstacles, which is given

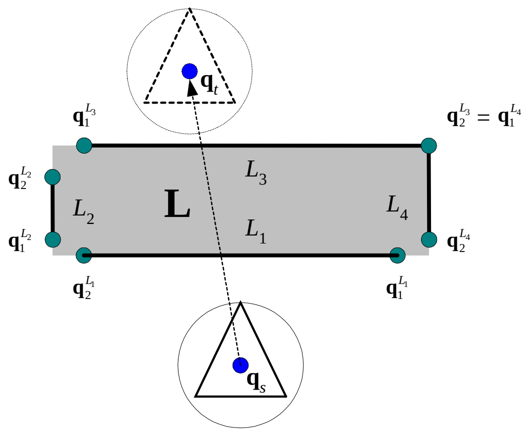

Fig. 1. Start and target position where line segments exist between them

by proximity sensors is limited to instantaneous and local points, Algorithm 1 of [8] for building the line segments is necessary to make the obstacles' shapes for motion planning. However, if we use continuous data from laser which will be explained in the next section, we can easily gather the direct information of the line segments.

3.1. Sets of Linked Line Segments (SLLS)

At first, let us use two notations and two definitions:

$d(\mathbf{q}_i, \mathbf{q}_j)$ is the distance between two points $\mathbf{q}_i$ and $\mathbf{q}_j$,

$d(\mathbf{q}, L)$ is the distance between a point $\mathbf{q}$ and line segment L.

Definition 1: L_i and L_j are said to be the **linked** with each other at $\mathbf{q}_k^{L_i}$ (or $\mathbf{q}_l^{L_j}$), if any pair of $\mathbf{q}_k^{L_i}$ and $\mathbf{q}_l^{L_j}$ $(k = 1, 2, \ l = 1, 2)$ exists such that $d(\mathbf{q}_k^{L_i}, \mathbf{q}_l^{L_j}) \leq 2\rho_r$ where ρ_r is the radius of a circle outside which obstacles are guaranteed not to collide with the robot.

In this definition, both $\mathbf{q}_k^{L_i}$ and $\mathbf{q}_l^{L_j}$ are merged into a new common edge (or corner) point as shown in Fig. 2. Hence, a set of linked line segments is composed of two extreme points and edge (or corner) points (See Fig. 3). Note that the corner point is to be rejected for the reference to the motion planning. Considering both extreme point and edge point as nodes and free path between them as arcs, the shortest path from start point to target point via the nodal points can

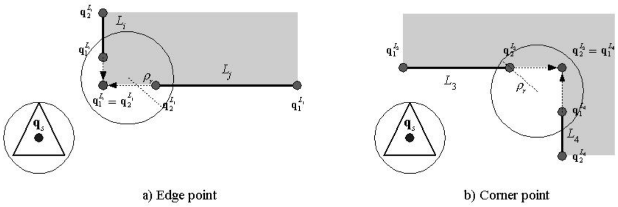

Fig. 2. Two types of merged points when linked

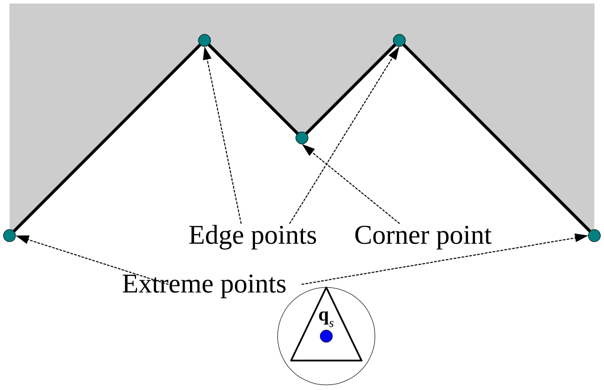

Fig. 3. A set of linked line segments

be calculated by graph search [9]. Figs. 4 and 5 shows an example. At first, Fig. 4 represents a SLLS established from the past sensing data. Target point $\mathbf{q}_t$ is considered as the first node $N_{0,0}$ where the first subscript i of $N_{i,j}$ means i^{th} level and the second subscript j means j^{th} sibling in the level. And then, $\mathbf{q}_5$ and $\mathbf{q}_6$ are the next level's nodes respectively $N_{1,0}$ and $N_{1,1}$ since they have free path from the previous level's node $N_{0,0}$. Repetitively graphical search is performed until the free path finds the start point $\mathbf{q}_s$, $N_{3,0}$ here. Then the shortest path and the resulted reference point is shown in Fig. 5.

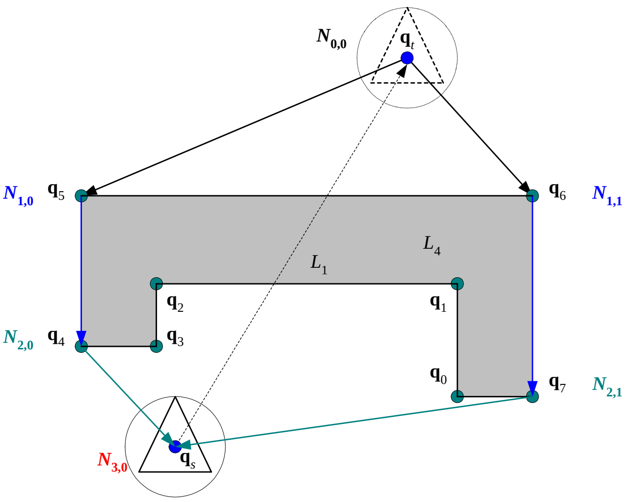

Fig. 4. A SLLS and its nodal points

3.2. Virtual Target Point in the Presence of Obstacles.

We establish a virtual target point (VTP), $\mathbf{q}'_t$ when there is no free path to real target point in the presence of obstacles

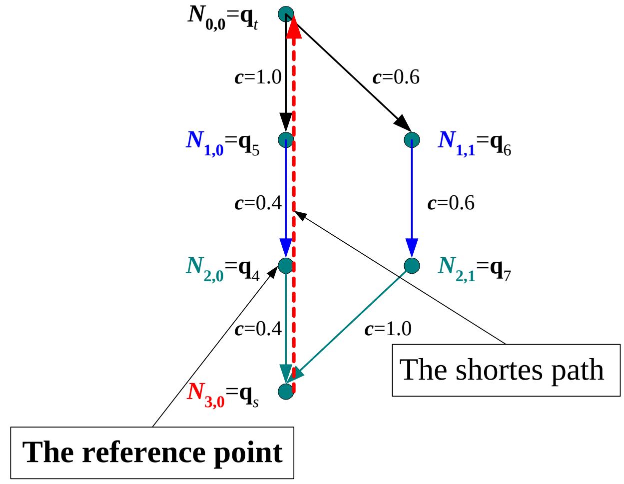

Fig. 5. The shortest path and the resulted reference point

which are represented as SLLS. VTP is generated from the reference point which is the closest point of SLLS in the shortest path from the start to the target. As shown in Fig. 6, the VTP $\mathbf{q}'_t$ is located on the arc from the target point $\mathbf{q}_t$ and simultaneously on the line which is started from the $\mathbf{q}_s$ and tangential to the circle centered at the reference point $\mathbf{q}_2^{L_i}$ ($\mathbf{q}_1^{L_i}$). The ρ_r explained previously is also used for the radius of the circle. Note that the virtual target point $\mathbf{q}'_t$ is continuously moved toward the real target point $\mathbf{q}$ although the robot's motion is perturbed from the planned direction, which is very important in real world for smooth motion.

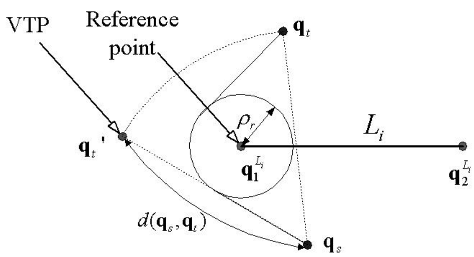

Fig. 6. Generation of the virtual target point from the reference point

4. Simulations and experimental results

In this section, simulations and experimental results are explained. Our sensing systems are divided into two kinds of devices as shown in Fig. 7. The major one is small laser whose emission is slit-typed plus camera (WebCAM) to detect the line reflection of the laser, and the other is sonar system which is composed of twelve ultrasonic sensors and a control board. (See Fig 8). The laser system has sensing capability of 1m in distance with 60 degree in front angle, and sampling speed of 100ms. With each frame image of 640x480 pixels captured from the camera, we extract red-colored pixels and consider them as point-wise data of obstacles. On the other hand, ultrasonic sensors have the maximum detectable distance of 2m with beam angle of 35 degree, and sampling speed of the sonar system is 70ms. The laser system gives more continuous data information, thus enables us to catch a line segment easily. However, the laser system is sensitive to light condition or obstacle's color, which may make non-existent obstacles or miss real existent obstacles as a result. We use ultrasonic sensors as the auxiliary sensors for simple sensor fusion considering the advantages in that they give exact information about the presence of any obstacle within certain range. By using this sensor fusion, the dynamic environments can be dealt easily: if any obstacle which has been registered in robot's memory is removed, the ultrasonic sensors give clear information about the resultant empty zones within their detection ranges and unregister the obstacle if its memorized location is included in the empty zones.

Figs. 9 to 12 show several captured screen of major steps in our simulation. Our mobile robot is located at origin $\mathbf{q}_s = [0\text{cm}\ 0\text{cm}]^T$ and tries to go to target $\mathbf{q}_t = [120\text{cm}\ \ -30\text{cm}]^T$ directly since it has no idea about its environments initially. Then, any obstacle is thought as an added dynamic environment. In the first figure (Fig. 9), the virtual target point $\mathbf{q}_t'$ is generated in the right side to the real target $\mathbf{q}_t$ since just a little part (one line segment) of obstacle was detected. After several steps of motion, with more sensing information, the line segment has grown to be linked line segments as shown in Fig. 10. That results in changing the virtual target from the right side to the left side. In other words, with no local minimum occurred, the robot performs intelligent motion as it detects more part (right side) of the obstacle. Fig. 11 shows more linked line segments causing slight changes in virtual target and in the resultant robot's motion as well. With the final result as shown in Fig. 12, the robot has reached the real target having neither local minimum or unnecessary detour motion around the lower side of the obstacle. Figs. 13 to 16 shows experimental results from the initial start to the final arrival, which explains successful performances of the proposed algorithm. Alos, final results in GUI of the experiment is shown in Figs. 17.

5. Conclusion

We proposed a local minimum free motion planning for mobile robots considering dynamic environments by simple sensor fusion. Potential field is used as basic platform for the motion planning since it has the advantages of simplicity, real-time computation. To get rid of the local minimum in the presence of obstacles, we have built the VTP considering the obstacles which are sensed by camera with laser and represented as SLLS. The VTP has been created from a reference point of the SLLS which is on the shortest path and closest to the start point. Also, in order to keep continuous motion, the virtual target is not fixed but changed continuously through a circular arc with regard to the updated information about obstacles and current robot's position. We used laser system as the main proximity sensors since it gives more continuous data information, and added ultrasonic sensors as the auxiliary sensors for simple sensor fusion considering the advantages in that they give exact information about the presence of any obstacle within certain range. By using this sensor fusion, the dynamic environments have been dealt easily. The performance of our algorithm has been validated via simulations and experiments.

As for the further works, we are going to consider noise of sensors to make robust motion planning even when surrounded by more complex obstacles. Also, moving obstacles are going to be dealt by dynamic removal of the sensed data if required.

References

[1] S. Simhon and G. Dudek, "A Global Topological Map formed by Local Metric Maps," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, Victoria, B.C., Canada, pp. 1708–1714, Oct. 1998.

[2] N. Tomatis, I. Nourbakhsh, and R. Siegwart, "Simultaneous Localization and Map Building: A Global Topological Model with Local Metric Maps," in *Proc. IEEE Int. Conf. Intelligent Robots and Systems*, Maui, HI, pp. 421–426, Oct. 2001.

[3] P. Khosal and R. Volpe, "Superquadric Artificial Potentials for Obstacle Avoidance and Approach," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Philadephia, PA, pp. 1778–1784, 1988.

[4] R. Volpe and P. Khosla, "Artificial Potentials with Elliptical Isopotential Contours for Obstacle Avoidance," in *Proc. IEEE Int. Conf. on Decision and Control*, Los Angeles, pp. 180–185, 1987.

[5] J. Barraquand and J. C. Latombe, "A Monte-Carlo Algorithm for Path Planning with Many Degrees of Freedom," in *Proc. IEEE Int. Conf. on Robotics and Automation*, Cincinnati, OH, pp. 1712–1717, 1990.

[6] J. Barraquand and J. C. Latombe, "Robot Motion Planning: A distributed Representation Approach," *Int. J. Robotics Research*, vol. 10, pp. 628–649, 1991.

[7] S. S. Ge and Y. J. Cui, "New Potential Functions for Mobile Robot Path Planning," *IEEE Trans. Robotics and Automation*, vol. 16, no. 5, pp. 615–620, 2000.

[8] J. S. Choi and C. won Lee, "Sensor-Based Motion Planning for Mobile Robots," in *Int. Conf. on Control, Automation, and Systems*, Muju, Korea, pp. 170–174, Oct. 2002.

[9] G. F. Luger and W. A. Stubblefield, eds., *Artificial Intelligence*. The Benjamin/Cummings Publishing Company, Inc., 2nd ed., 1993.

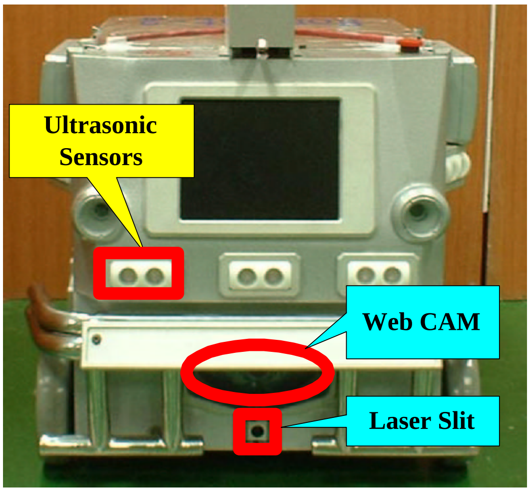

Fig. 7. WebCAM with Laser)

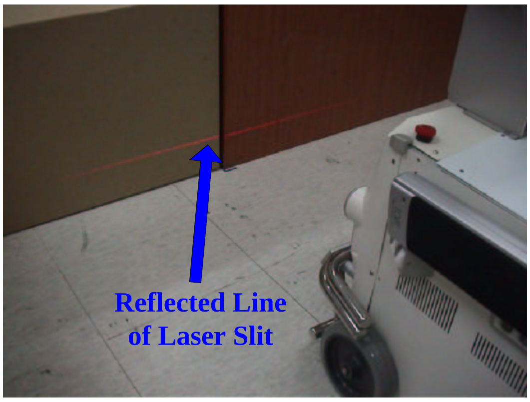

Fig. 8. Reflected line of laser)

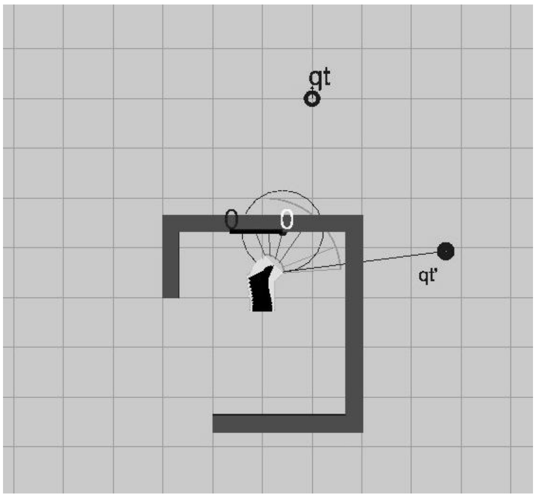

Fig. 9. Simulation a)

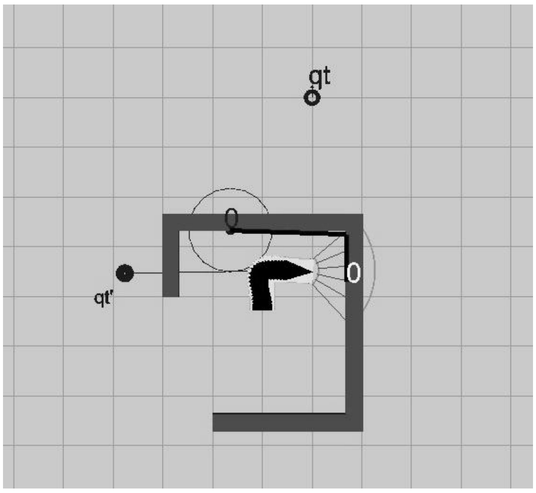

Fig. 10. Simulation b)

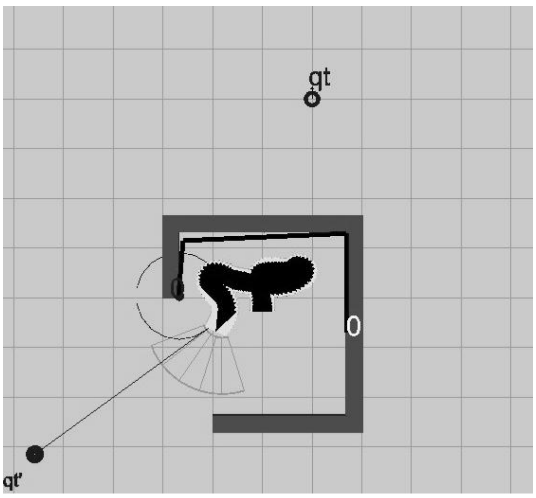

Fig. 11. Simulation c)

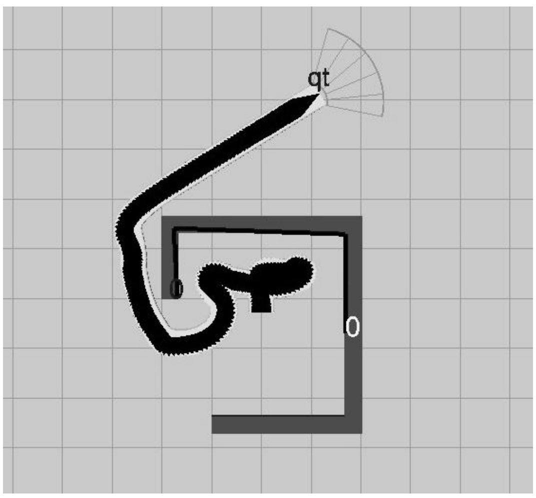

Fig. 12. Simulation d)

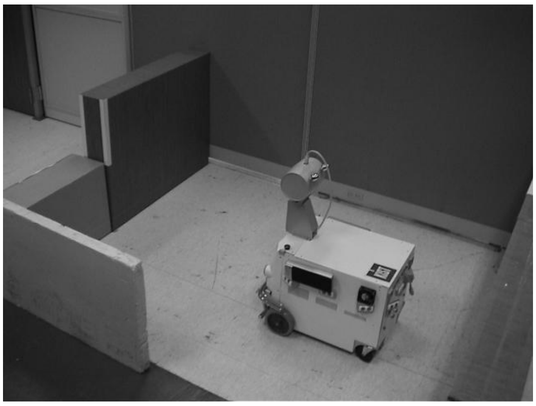

Fig. 13. Experiment a)

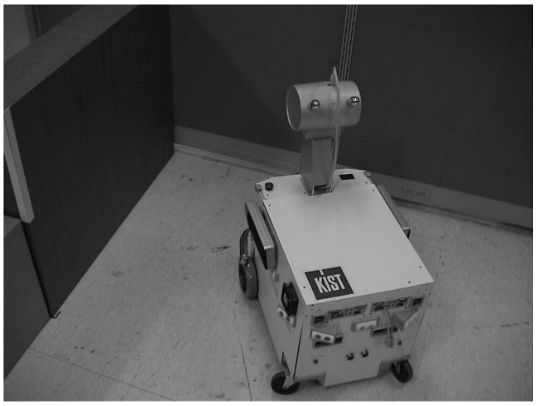

Fig. 14. Experiment b)

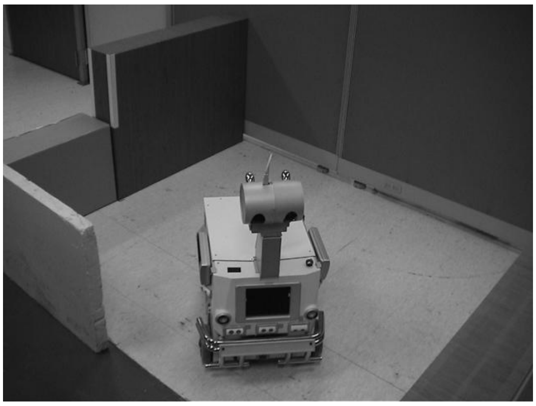

Fig. 15. Experiment c)

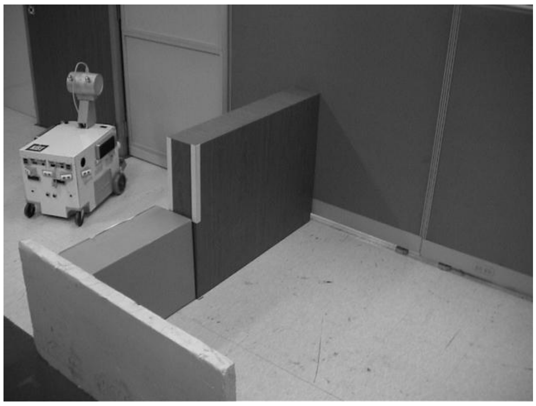

Fig. 16. Experiment d)

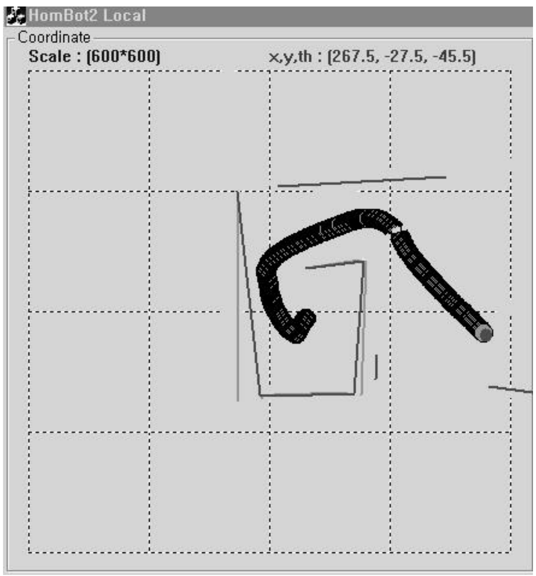

Fig. 17. Final results in GUI of the experiment)