SICE Annual Conference 2008
August 20-22, 2008, The University Electro-Communications, Japan

Real-time Motion Planning in Unknown Environment: a Voronoi-based StRRT (Spatiotemporal RRT)

Hiroto Sakahara[1], Yasuhiro Masutani[2] and Fumio Miyazaki[3]

[1]Department of Mechanical Science and Bioengineering, Osaka University, Osaka, Japan
(Tel : +81-6-6850-6208; E-mail: sakahara@robotics.me.es.osaka-u.ac.jp)
[2]Department of Computer Science, Osaka Electro-Communication University, Osaka, Japan
(Tel : +81-72-876-5107; E-mail: masutani@isc.osakac.ac.jp)
[3]Department of Mechanical Science and Bioengineering, Osaka University, Osaka, Japan
(Tel : +81-6-6850-6205; E-mail: miyazaki@me.es.osaka-u.ac.jp)

Abstract: For a robot to move autonomously in an unknown dynamic environment, a real-time motion planning is necessary. When the motions of obstacles are unknown, motion planning is usually performed by virtually expanding the area obstacles occupy. However, there are many cases in which the adequate solution trajectory cannot be obtained due to inadequate expansion. Another approach would be to determine candidates of the safe trajectory from the layout of obstacles as a generalized Voronoi diagram and then randomly search for an appropriate trajectory. However, this requires many calculations. It may be sufficient if a local generalized Voronoi diagram is available for limiting the sampling area. Based on this idea, this paper proposes a Voronoi-based StRRT composed of an StRRT subjected to biasing extraction of sample points toward the border of a generalized Voronoi diagram and show the possibilities of this method to plan safe trajectories.

Keywords: Mobile robot motion-planning, Navigation, Real time systems, Safety

1. INTRODUCTION

For a robot to move autonomously in an environment with human beings and multiple robots and to cope with dynamic changes and the unpredictable properties of such an environment, a real-time motion planning and control function that obtains environmental information within an allowable time limit and renews and executes the motion plan is necessary. Many motion planning algorithms have been proposed in recent years for actualizing this function. One of these, the random-sampling technique, enables searching the workspace quickly and efficiently, so it is a practicable motion planning algorithm for movement tasks under various environments [1]-[6]. Random-sampling techniques that have been proposed so far can be broadly divided into two categories, multiple-query motion planning[1]-[5] and single-query motion planning[6]. A movement task in which individual moving robots are given a single goal will require the latter category, i.e., a single-query motion planning algorithm. RRT is a technique for quickly searching for a route to the target point that avoids obstacles by extending a random tree consisting of edges and nodes in the workspace. When utilizing this technique in real-time motion planning, the efficiency of searching for the path can be further improved by partially reusing or refining an existing tree.

A common approach to handle moving obstacles utilizes a configuration-time space which consists in adding the time dimension to the robot's configuration space[7]. Although this method has usually been applied to motion planning in a known operating environment, it has come to be applied to real-time motion planning in an unknown operating environment in recent years. A method in which RRT is expanded to the configuration -time space has also been proposed. Spatiotemporal RRT (StRRT) [8], which we proposed, is one such method and enables partially reusing an existing tree and quickly searching the spatiotemporal path, taking into consideration the restrictions on the kinematics and dynamics of the robot just like the methods based on an RRT search for the path in the workspace. This method is also able to cope with the problem of not being able to determine the required time to arrive at the target point, which is unavoidable in an unknown dynamic environment.

When the motions of obstacles are unknown, motion planning is usually performed by virtually expanding the area obstacles occupy. However, there are many cases in which the solution trajectory cannot be obtained due to excessive expansion even if the target point can be reached in reality or no safe trajectory can be obtained due to insufficient expansion. Another approach would be to determine candidates of the safe trajectory from the layout of obstacles as a generalized Voronoi diagram and then randomly search for an appropriate trajectory [8]-[10]. In this case, search efficiency improves and a safer trajectory can be obtained because the points for random sampling are limited. However, generation of a generalized Voronoi diagram for the whole workspace requires many calculations, so it is necessary to incorporate this portion into hardware. For real-time motion planning that repeats trajectory planning and execution through local space searching, however, it is not always necessary to obtain a generalized Voronoi diagram for the whole workspace. It may be sufficient if a local generalized Voronoi diagram is available for limiting the sam-

PR0001/08/0000-2326 ¥400

pling area when letting the tree grow. Based on this idea, this paper proposes a Voronoi-based StRRT composed of an StRRT subjected to biasing extraction of sample points toward the border of a generalized Voronoi diagram. Because this technique does not require a generalized Voronoi diagram, a safer trajectory can be searched within a short time without incorporating the processing into hardware.

This paper is organized as follows. Section 2 describes the problems in real-time motion planning in dynamic workspace, i.e. motion planning for a robot moving in a workspace with unknown moving obstacles. Section 3 introduces StRRT, which is a method in which RRT is expanded to the space including the time axis, together with its algorithm. Section 4 presents a method for extracting sample points by applying bias toward the border of a generalized Voronoi diagram. Section 5 verifies the effectiveness of the Voronoi-based StRRT through several numerical experiments.

2. MOTION PLANNING IN UNKNOWN DYNAMIC ENVIRONMENTS

Consider a mobile robot tasked with moving from the start point to its goal while avoiding obstacles in some environment. The environment is assumed to be a two dimensional closed area in which a group of obstacles are moving while irregularly changing the speed and the direction. It is also assumed that the mobile robot acquires the environmental information, plans the motion trajectory, and generates the action every predefined constant time. In other words, the positions and velocities of the robot itself ($R(t_i)$) and obstacles ($O_j(t_i)$) ($j = 1, 2, ...$) are given at the i-th sampling time $t_i(i = 1, 2, ...)$ and the velocity command ($C(t_i)$) at t_i, which is calculated using some motion planning algorithm, is output to the robot's controller. Repeating this process every sampling period, the robot moves toward its goal point(Fig.1).

As for motion planning algorithms, most of them use the only positional information, which is insufficient in planning the robot's motion in a dynamically changing environment. In such a case, it is required to accompany the environmental information with the velocities of the robot and obstacles. This can be realized by utilizing a configuration-time space, in other words, the space with time. Spatiotemporal RRT (StRRT) [8], an extended version of Rapidly-exploring Random Trees (RRT) [1], can generate the trajectory taking into consideration a temporal change in the environment, by applying a random sampling to the working space to which a time axis is added.

When performing the motion planning in a two dimensional plane, StRRT considers the three-dimensional space (x, y, t) where a time-axis t is added to the xy plane as shown in Fig.2. If the goal point remains stationary, it is expressed as a straight line parallel to a time-axis in the three dimensional space. This straight line is called Destination Line. A circular obstacle which moves at

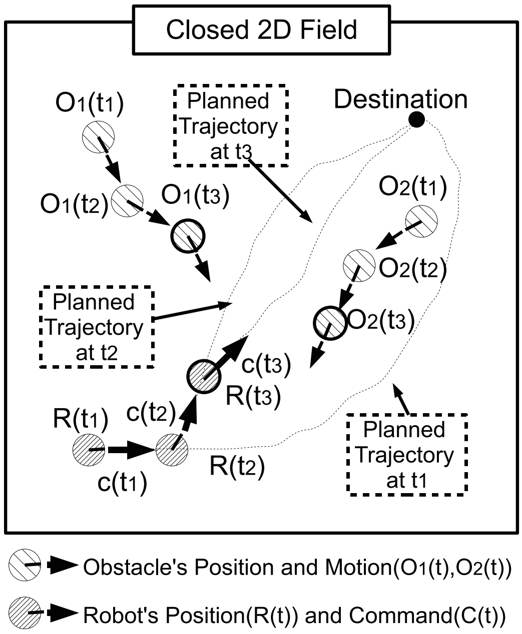

Fig. 1 Trajectory Generation: Making Successive Velocity Command with Real Time Sensing

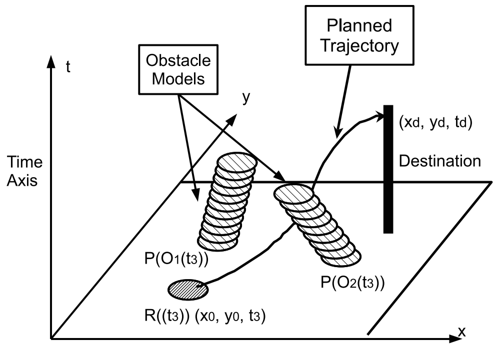

Fig. 2 Spatiotemporal Path Generation at t_3

a fixed velocity $v_{ab}$is expressed as an inclined cylinder with an inclination of $\tan^{-1} \frac{1}{v_{ab}}$ to the xy plane as shown in Fig.2. If the obstacle does not move at a constant velocity, we consider that the velocity of the obstacle can be approximated constant within a short time interval. In this paper, we assume that an environmental change is observed every definite period of time interval and the motion planning is executed within this time interval. The velocity command to the robot is also generated in this period and the robot is actually moved. We will explain StRRT in more detail in the next section.

3. STRRT(SPATIOTEMPORAL-RRT)

3.1 Basic Procedures of RRT algorithm

We write up the basic procedures of RRT algorithm which searches for the route from the initial point to the goal point while extending the search tree in the work space before explaining StRRT that is the technique for enhancing RRT to the spatiotemporal workspace.

1) Extracting sample points

Basically, a candidate of passed point of the route (a sam-

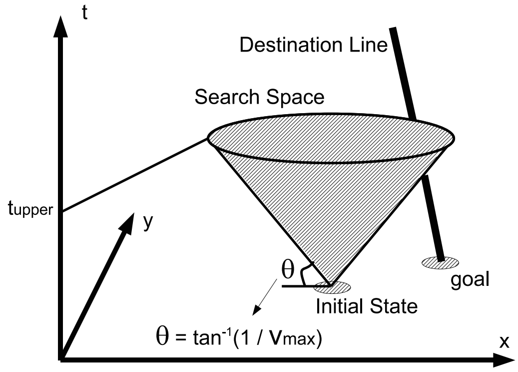

Fig. 3 Spatiotemporal Search Space

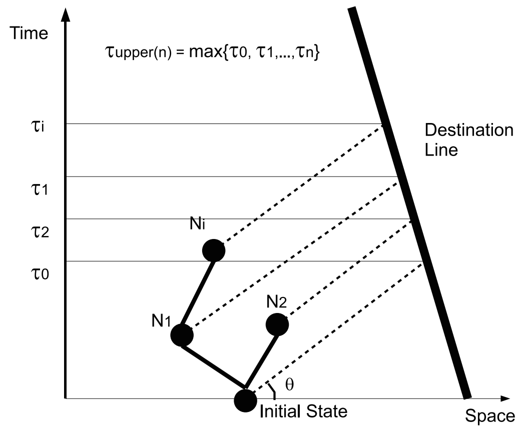

Fig. 4 The Search Area of time-axis

ple point) is extracted at random in the workspace like the other random sampling method.

2) Choice of the nearest neighbor node of the tree
The node in the existing tree nearest to the sample point which is extracted in the procedure 1) is chosen.

3) Decision of the new node and updating the tree
A temporary node is placed on the segment connecting the sample node and its nearest neighbor node in the existing tree. If there is no obstacle on this segment, the temporary node becomes the new node. If there are obstacles on this segment, the tree is not updated at that time.

3.2 StRRT algorithm

Most important problem about extension of RRT to the spatiotemporal space is to decide the arrival time at the goal point. The position of the goal point is spatially definite, but the time will vary with the planned motion of the robot. Therefore, the time value of the goal point can't be predefined and the following procedure is required for handling this problem.

3.2.1 Obstacle models in spatiotemporal space

When the motion is planned under the environment with obstacles varing the speed and direction randomly, it is desirable to use the acquirable dynamic information as effective as possible. In the StRRT, the obstacle model is given as an oblique circular cylinder in the spatiotemporal space assuming that the robot moves with a constant sensed velocity. In addition, if the constant margin or growing margin along the time axis is considered, StRRT can handle uncertain factors.

3.2.2 Search area of the tree

The search area of the tree is restricted in a subspace of the spatiotemporal space. That is, the search area is within the inverted cone whose vertex is the current position of the robot as shown in Fig.3. The side surface of this cone is inclined at an angle of $\theta(=\tan^{-1}1/v_{max})$ (v_{max} is the maximum speed of the robot). The upper limit of time in searching the tree, that is, the base of the cone, is given as explained below.

3.2.3 The search area of time-axis

Though the arrival time at the goal point can not be predefined, its upper limit can be defined. When generating a new node, the minimal arrival timeτ_i at the goal point from each node N_i is calculated assuming that the robot moves at the maximum speed (with the angleθ) from N_i directly to the goal point as shown in Fig.4.

Then the time τ_i is compared with the minimal times from the other nodes. Finally the maximum minimal time will be the upper limit of time in searching the tree. In this way, the search area of the tree is adaptively adjusted.

3.2.4 Real time operation of StRRT

There is a case that StRRT can't momentarily plan the trajectory to the goal point in the environment with many dynamic obstacles moving randomly. In such a case, the minimal arrival time τ_i of each node is calculated and the node from which the calculated arrival time is the minimum is selected. Then, the trajectory is generated based on the tree connecting the selected node and the current node.

3.2.5 Refinement of the trajectory

If the whole trajectory to the goal point is found in the time interval allotted for the motion planning, better trajectories can be found within the remaining time. Once StRRT finds more than one trajectory to the goal point, StRRT evaluates them from the point of the arrival time. In this way, StRRT can refine the trajectory to the goal point.

3.2.6 Reuse the existing trajectory

The trajectory generated in the last motion planning may be reused partially even if the environmental information is updated. This can be realized by extracting the usable part of the trajectory with the updated information and extending it further using StRRT.

4. VORONOI BASED STRRT

In this section, we present the Voronoi Based StRRT, that is to say the safety-conscious StRRT. This method conceptually incorporates a generalized Voronoi diagram

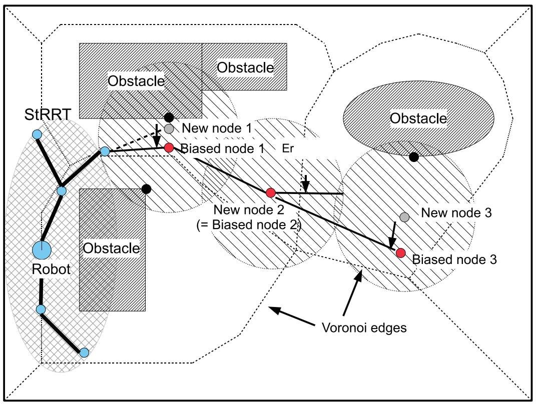

Fig. 5 Basic Idea of Voronoi Based StRRT

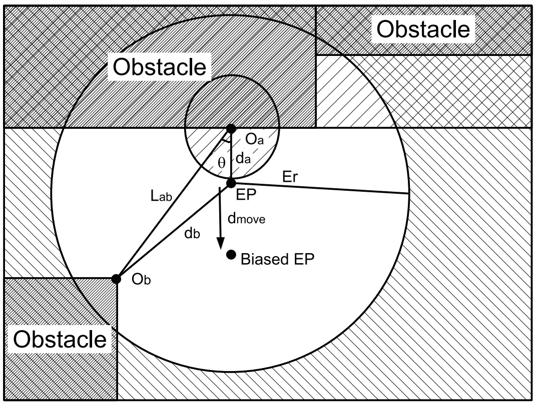

Fig. 6 Biasing Extraction of Sample Points

into StRRT and enables the motion planner to generate a safer trajectory by biasing extraction of sample points toward the border of a generalized Voronoi diagram as shown in Fig.5.

4.1 Biased extraction of sample points

A method using generalized Voronoi diagrams to generate a mobile robot's path greatly reduces the possibility that the robot will actually come in contact with an obstacle. The Voronoi diagram for a collection of given points is the graph formed by the boundaries of specially-constructed cells (Voronoi edges). Each of these cells surrounds one of the given points and has the property that all points within the cell are closer to the enclosed point than to any other point. Voronoi diagrams can be generalized to situations in which the given points are two-dimensional obstacles rather than mere points. Although useful approximation methods for finding the generalized Voronoi diagram for a collection of two-dimensional obstacles have been proposed, generating the generalized Voronoi diagram for the whole workspace requires many calculations. For real-time motion planning that repeats trajectory planning and execution through local space searching, it is not necessarily to obtain a generalized Voronoi diagram for the whole workspace. It may be sufficient if a local generalized Voronoi diagram is available for limiting the sampling area when letting the tree grow. Based on this concept, we consider biasing extraction of sample points toward near the Voronoi diagram's edges. The procedures to move a sample point to near the Voronoi edges are as follows.

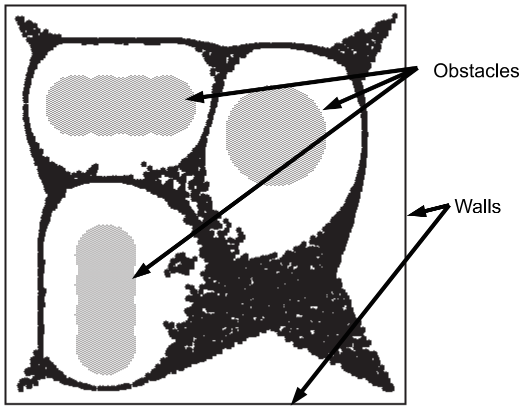

Fig. 7 Moved Sample Points

- Move a sample point (EP in Fig.6) in the direction opposite to the nearest point on the surface of the nearest obstacle.
- Adjust the amount of movement d_{move} depending on the distribution of neighboring obstacles and the distance between EP and the obstacle's nearest point.

We explain how to adjust d_{move} in more detail. Let O_a be a point on the surface of obstacles nearest to EP, d_a be the distance between EP and O_a, and O_b be a point on the surface of obstacles second nearest to EP which is located within the area satisfying $|\theta| < \pi/2$ and $L_{ab} > d_a$ where θ is the angle between the two segments $EP - O_a$ and $O_a - O_b$, and L_{ab} is the distance between O_a and O_b as shown in Fig.7. Then, introducing a parameter Er, d_{move} is determined by

$$d_{move} = 0 \quad \text{if } E_r < d_a \qquad (1)$$

$$d_{move} = (E_r - d_a)/2 \quad \text{if } d_a \le E_r < d_b \qquad (2)$$

$$d_{move} = (d_b - d_a)/2 \quad \text{if } d_b \le E_r \qquad (3)$$

where d_b is the distance between EP and O_b. The parameter E_r plays an important role in moving a sample point to near the Voronoi edges.

4.2 An example of moved sample points

Fig.8 shows the result of moving sample points randomly selected in a workspace including static obstacles by using the above-mentioned procedures. We can see that sample points cluster near the Voronoi diagram's edges.

5. SIMULATION RESULTS

In this section, We evaluate Voronoi Based StRRT comparing it with the original StRRT through numerical simulations. Because of the random nature of trajectory generation and environmental changes, we have run 100 trials in the same simulation environment.

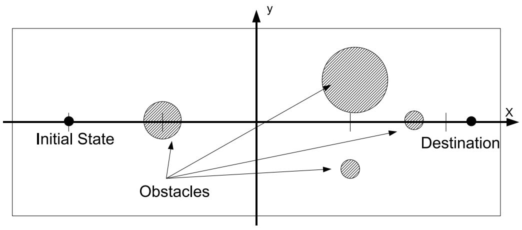

Fig. 8 Static Environment

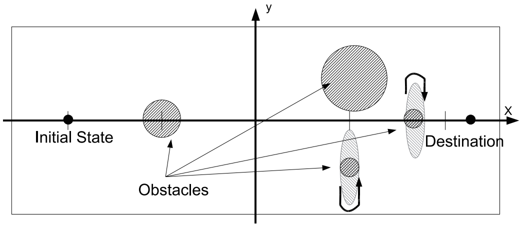

Fig. 9 Dynamic Environment

Table 1 Experimental Results in a Static Environment with StRRT corresponding to Fig.10 (Success : success rate, Time : average arrival time, Distance : nearest distance from obstacles)

Margin	Success	Time	Distance
50mm	100 / 100	4.36s	74.4mm
100mm	100 / 100	4.61s	122mm
200mm	100 / 100	5.39s	209mm
500mm	0 / 100	—	—
1000mm	0 / 100	—	—

Table 2 Experimental Results in a Dynamic Environment with Voronoi Based StRRT corresponding to Fig.11 (Success : success rate, Time : average arrival time, Distance : nearest distance from obstacles)

Er	Success	Time	Distance
50mm	100 / 100	4.11s	30.6mm
100mm	100 / 100	4.48s	70.9mm
200mm	100 / 100	4.89s	135mm
500mm	100 / 100	5.47s	170mm
1000mm	100 / 100	4.93s	134mm

5.1 Simulation environment

Fig.8 shows a static environment with four static obstacles and Fig.9 shows a dynamically changing environment with two static obstacles and two dynamic obstacles moving back and forth along a longitudinal line in a two-second cycle with a piecewise constant acceleration (the maximum speed is 600 mm/s). The task for the robot is to move from the start point to its goal while avoiding obstacles in both environments.

5.2 Results

Fig.10 and Fig.11 indicate how the minimum distance between the robot and obstacles in the process of moving toward the goal varies depending on the parameter set for generating the robot's safe trajectory. The parameter called "margin" corresponds to the size of expansion of the area obstacles occupy in case of the original StRRT.

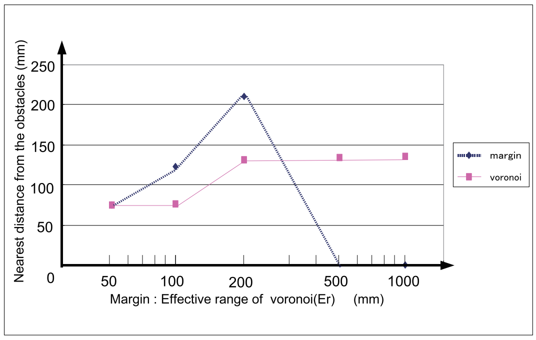

Fig. 10 Nearest Distances from obstacles in a Static Environment

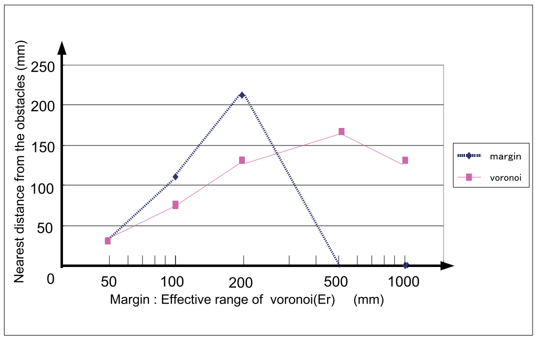

Fig. 11 Nearest Distances from obstacles in a Dynamic Environment

On the other hand, in case of Voronoi Based StRRT, this parameter corresponds to the value of the parameter E_r. We can see from these figures that the adequate path to the goal cannot be obtained if we set more than 500mm as the size of expansion in case of the original StRRT, which corresponds to zero distance from the nearest obstacle in both figures. In case of Voronoi Based StRRT, however, the adequate path to the goal can be obtained no matter how we set the value of E_r. Moreover, the distance from the nearest obstacle increases and approaches some constant value as the value of E_r increases. Table.1 shows the results of 100 trials obtained by changing the margin, which corresponds to blue data points in Fig.10. Table.2 shows the results of 100 trials obtained by changing E_r, which corresponds to red data points in Fig.11. Success denotes the rate of success in arriving at the goal and Time means the average time it takes to arrive at the goal. We see that all trials ended in success except for the cases where we set more than 500 [mm] as the size of expansion and had almost equal average time. Typical path of the robot generated using each planning method in a dynamically changing environment is shown in Fig.12 and Fig.13 respectively. The red marks in these figures mean the closest points of moving obstacles to the robot in the process of moving toward the goal. Comparing these figures, we can see that Voronoi Based StRRT exhibits the feature that the robot's path is generated near the Voronoi diagram's edges.

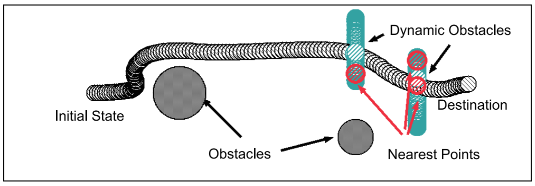

Fig. 12 Trajectory in Dynamic Environment with StRRT

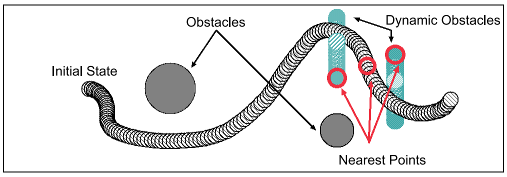

Fig. 13 Trajectory in Dynamic Environment with Voronoi Based StRRT

6. CONCLUSION

In this paper, we have proposed a method of planning the safe trajectory of the robot in an unknown dynamic environment in real-time. This method called Voronoi-based StRRT is composed of an StRRT subjected to biasing extraction of sample points toward the Voronoi diagram's edges. Though the robot's path to the goal generated using Voronoi-based StRRT does not follow the Voronoi diagram's edges, it keeps in the neighborhood of Voronoi edges. We have verified through numerical simulations that a safer trajectory can be searched within a short time.

REFERENCES

[1] S. M. LaValle,"Rapidly-exploring random trees: A new tool for path planning", TR98-11, *Computer Science Dept. Iowa State University, Oct.*, 1998.

[2] J. Bruce, and M. Veloso, "Real-time randomized path planning for robot navigation", *Proceedings of IROS-2002*, pp.2383-2388, 2002.

[3] S. M. LaValle, and J. J. Kuffner,"Randomized Kinodynamic planning", *Proceedings of ICRA-1999*, pp.473-479, 1999.

[4] D. Hsu, R. Kindel, J.C. Latombe, and S. Rock," Randomized kinodynamic motion planning with moving obstacles", *Int. J. Robotics Research*, Vol.21, No.3, pp.233-255, 2002.

[5] M. Zucker, J. Kuffner, and M. Branicky, "Multipartite RRTs for Rapid Replanning in Dynamic Environments", *Proceedings of ICRA-2007*, April, 2007.

[6] J. J. Kuffner Jr, and S. M. LaValle,"RRT-Connect: An Efficient Approach to Single-Query Path Planning",*Proceedings of ICRA-2000*, pp.995-1001, San Francisco, CA, 2000.

[7] T. Tsubouchi, S. Kuramochi, and S. Arimoto, "Iterated forecast and planning algorithm to steer and drive amobile robot in the presence of multiple moving objects", *Proceedings. of IROS-1995*, Vol.2, pp33 - 38, 1995.

[8] H. Sakahara, Y. Masutani, and F. Miyazaki, "Real Time Motion Planning in Dynamic Environment Containing Moving Obstacles Using Spatiotemporal RRT", *Transactions of the Society of Instrument and Control Engineers*, Vol.43, No.4, April , pp.277 - 284, 2007.

[9] S.Garido, L.Moreno, M. Abderrahim, and F. Martin, "Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching", *Proceedings of IROS-2006*, pp.2376 - 2381, 2006.

[10] K. Hoff, T. Culver, J. Keyser, M. Lin, and D. Manocha, "Interactive motion planning using hardware-accelerated computation of generalized Voronoi diagrams" *Proceedings of ICRA-2000*, Vol. 3, pp.2931 - 2937, 1999.