# Probabilistic Road Maps with Obstacle Avoidance in Cluttered Dynamic Environment

Rami Al-Hmouz. Tauseef Gulrez. Adel Al-Jumaily
*Centre of Excellence for Autonomous Systems, Mechatronics and Intelligent Systems Group, faculty of Engineering, University of Technology Sydney, Australia.*
*{ralhmouz, tgulrez, adel}@eng.uts.edu.au*

**Abstract**

*This paper presents an experimental study of a Probabilistic Road Map (PRM) based obstacle avoiding algorithm, for motion planning of a non-holonomic mobile robot in cluttered dynamic environment. The PRM approach uses a fast and simple local planner to build a network representation of the configuration space. It is trading off the distance to both static objects and moving obstacles in compute the travelled path. Our work has been implemented and tested on Player / Stage, real time robotic software, in extensive simulation runs. The different experiments that runs had demonstrate that our approach is well suited to control the motions of a robot in a cluttered environment and demonstrates its advantages over other techniques.*

## 1. Introduction

Path planning is essential problem need to be solved in autonomous mobile robot. Chang [1] defines the capability of effectively planning as its motions are "Eminently necessary since, by definition, a robot accomplishes tasks by moving in the real world". Especially in the context of autonomous mobile robots, path planning technique has to simultaneously solve two complementary tasks. Firstly, the task is to minimize the length of the trajectory from the starting position to the target location, and secondly, they should maximize the distance from obstacles in order to minimize the risk of colliding with an object.

Automatic motion planning has application in many areas such as robotics, virtual reality systems, and computer-aided design. Although many different motion planning methods have been proposed, most are not used in practice since they are computationally infeasible except for some restricted cases like the case of mobile robots [2]. Indeed, there is strong evidence that any complete planner (one that is guaranteed to and a solution or determine that none exists) will require time that is exponential in the number of degree of freedom (dof) of the robot [3]. Recently, a new class of randomised path planning methods, known as "Probabilistic Roadmap Methods (PRMs)", has shown a great interest among the scientists dealing with autonomous mobile robot's path planning problem., The attention has focussed on randomized or probabilistic motion planning methods, notable among these are randomized potential field methods (RPP) [4], which work very well when the configuration space (C-space) is relatively uncluttered, but unfortunately they are also not successful in exist simple situations in which they need to work in [5][6]. Recently, a new class of randomized motion planning methods has gained much attention [7][8][9][6][10][11][12]. These methods, known as probabilistic roadmap methods (PRMs), use randomization (usually during pre-processing) to construct a graph in C-space as a roadmap [13]. Roadmap nodes correspond to collision-free configurations of the robot. Two nodes are connected by an edge if a path between the two corresponding configurations can be found by a local planning method. Queries are processed by connecting the initial and goal configurations to the roadmap, and then ending a path in the roadmap between these two connection points.

In this paper the PRM based obstacle avoidance algorithmic method for an autonomous mobile robot in dynamic environment is presented. The structure of this paper is as follows. In section 2 the mobile robot's configuration space is defined mathematically with two subsections shows the failure and success probability of finding a path. In section 3 the main algorithm is presented, followed by the, implemented GUI environment. Shortest path and obstacle avoidance techniques are presented in section 4 and 5. This paper is concluded in section 6 with future work in section 7.

## 2. Configuration Space

This section is written to provide some analysis of the running time configuration of PRM. The main objective is to show how probabilistic completeness can be proved for a given choice of planning problem, local planner and configuration generator. A PRM planner is probabilistically complete for any query, the probability of answering after building a roadmap goes to non-zero. In the following treatment, we analyse a simplified version of PRM which tries all pairs of connections in the roadmap.

We begin by providing an analysis of PRM operating in Euclidean $R^k$ space. Let $C_f$ (free-configuration) be an open subset of $[0,1]^k$ and let 'd' be the Euclidean metric on $R^k$. The local planner for the PRM connects points $x, y \in C_f$ when the straight line $\overline{xy}$ lies in $C_f$. The measure μ denotes the volume of a region of space, for example $\mu([0,1]^k)=1$. For any measurable subset $A \subset R^k$, $\mu(A)$ is its volume. The PRM we describe uses the uniform distribution on $C_f$ for sampling points. If $A \subset C_f$ is a measurable subset and $x$ is a random point chosen by the point sampling function then [14][12]

$$\Pr\left(x \in A\right) = \frac{\mu\left(A\right)}{\mu\left(C_f\right)} \qquad (1)$$

A path in free space consists of a continuous map $\gamma:[0,1]\rightarrow C_f$. The path is said to be from *(0)* to *(1)*. The clearance of a path is the furthest distance away from the path at which a given point can be guaranteed to be in free space. If is a path in free space then [14][12]

$$\text{Clearance}(\ ) = \sup_{\varepsilon \in R} \forall x \in R^n, r \in [0,1] : d(x,r) < \varepsilon \Rightarrow x \in C_f \qquad (2)$$

*A. Failure Probability*

The map has static obstacles, where $\varepsilon < 0$ is related to the *path clearance*, the minimum distance of a path to the obstacles. A result relating the probability of failure to the length of the isolated path and clearance was given. An extension of this result was made for when varied along the path. Finally, this result shows that the probability of failure of finding a particular path goes exponentially to zero as the number of points increase. The simplest of these results is related below.

*Theorem 1.1:* Let $a,b \in C_f$ such there exists a path between them lying in the $C_f$. Then the probability PRM will answer the query (a,b) correctly after generating $N$ points with probability greater than[12]:

$$\Pr\left[(a,b)\,Failure \leq \frac{2L}{\varepsilon} e^{-\alpha\ \varepsilon^k N}\right] \qquad (3)$$

where $L$ is the length of path , *= clearance( )* and $\alpha = \dfrac{\mu(B_1(0))}{2^k\,\mu(C_f)}$.

An extension to this technique can be made for small-time locally controllable robots [19], such as car-like robots and tractor-trailer robots. The property exploited is that for every point for $x > 0$ and $\varepsilon > 0$, there exists $\delta > 0$ such that any point within $\delta$ distance of $x$ can be reached by taking a path that stays within the $\varepsilon$ around $x$. A path with $\varepsilon$ clearance can thus be tiled with $\delta$. Again, the probability of failure was shown to decrease exponentially as increases.

*B. Reachability Probability*

In this section, we relate an argument to determine how PRM road maps capture the connectivity of space using the model from the previous section, we begin from the observation that $C_f$ can be broken into a union of connected components $C_{f1},\ldots\ldots,C_{fj},\ldots\ldots$. PRM samples a set of points N and computes the roadmap. We can think of N breaking into $N_1,\ldots\ldots,N_j,\ldots$ so that $N_j$ $N_j \subset C_{fj}$. The roadmap also breaks into components $R_1,\ldots\ldots,R_j,\ldots$ In this section, we bound the probability that every $R_j$ in the roadmap is connected by assuming some visibility properties. We begin by defining the reachable set.

This paper further formalized the notions of reachability and made use of measure. Concisely, for a connected set of points *S*, the *-Lookout (S)* is the subset of '*S*' whose points "see" using the local planner more than a fraction of the set of points which can be "seen" from . A space is *( , )*-Expansive, if the subset *-Lookout (S)* is always larger than an $\alpha$. Fraction of the measure of '*S*' for every connected subset '*S*' of the points reachable from any point in free space. Again, this paper provides a bound on the number of points in terms of , and required to generate a path, as shown below.

*Theorem 1.2:*

Let $\gamma \in [0,1]$ be a constant, and '*M*' be a set of '*2n+2*' points chosen independently and uniformly at random in free space '*F*' , which breaks into connected components $F_j$ [12]. Let $R_j$ be the roadmap defined by *M* on $F_j$. If

$$n = 8\ln\left(\frac{8}{\varepsilon\alpha\gamma}\right) / \varepsilon\alpha + \frac{3}{\beta} \qquad (4)$$

then with probability $1-\gamma$, $R_j$ is connected.

The configuration space is a powerful conceptual tool because it seems to be the natural space where the path planning problem lies.

## 3. The Algorithm

PRMs use randomization (usually during pre-processing) to construct a graph of representative paths in C-space (a roadmap) whose vertices correspond to collision-free configurations of the robot and in which two vertices are connected by an edge if a path between the two corresponding configurations can be found by a local planning method. The proposed path planning algorithm is as follows is:

***1****: Pre-processing of the environment.*
***2****: Generate N points at random (node generation) in the free configuration area.*
***3****: Locate start and the goal points in the graph.*
***4****: Connect the points that have certain distance to each other (connections).*
***5****:* ***for*** *each query of the form, "is there path from start to end.*
***6****:* ***if*** *paths from start to goal lie in.*
***then***
***7****: Compute the shortest path (Djikstra's shortest path).*
***8****: else*
***9****: return* *go to 2.*
***10****: end if*
***11****: end for*
***12****: Reactive obstacle avoidance*
*Stop* ***OR*** *Deviate*
***13****: Follow the planned path.*

PRMs have been shown to perform well in practice. In particular, after the roadmap is constructed during pre-processing such as the one shown in figure.1, many difficult planning queries can be answered in fractions of seconds. Although PRMs are particularly suitable when multiple queries will be answered in the same static environment, the general PRM strategy can be used to solve single queries by only constructing useful portions of the roadmap.

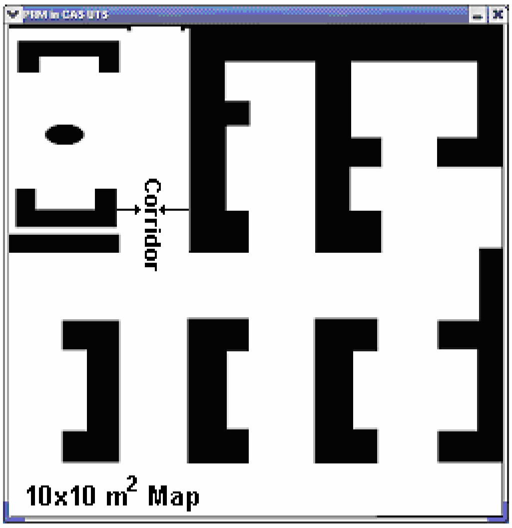


Fig.1: Pre-processing of environment

*A. Node Generation*

The first PRMs use uniform sampling in C-space to generate roadmap candidate nodes (collision-free configurations are retained); roadmaps are enhanced by further sampling in `difficult' regions. These methods are not well suited for dynamic environment, and their effectiveness decreases as the environments become more cluttered since uniform sampling of C-space is unlikely to yield configurations in narrow regions of C-space. To obtain improved roadmaps in crowded situations, we have used PRMs with random nodes generator which can explore maps that cannot be expected by the uniform samplers. Random node generation strategies are the methods used to select collision-free robot configurations to be used as nodes in the roadmap. A good node generation strategy will produce nodes that can be connected to form a roadmap that is representative of the connectivity and complexity of C-free. Ideally, the roadmap should contain nodes in every C-space crevice and corridor that robot can move freely in them (such as the environment shown in figure1). However, guaranteeing this requires the costly computation of the constraint surfaces which is what randomized methods seek to avoid. Every single node should come in the free area, and should have a safe distance form the obstacle, in another word the obstacles are scaled to make the robot moves freely if the path comes close to the obstacles.

*B. Connection*

After the collision-free roadmap candidate nodes are generated as shown in figure.2, they must be connected to form the roadmap as depicted in figure.3.

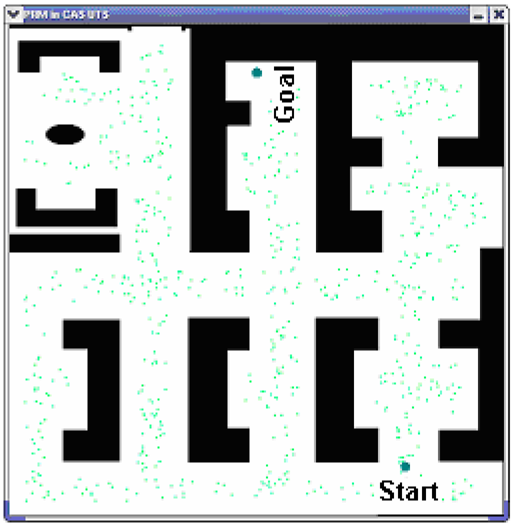


Fig.2: Random nodes generation

Fig.3: Nodes connection

The basic idea is to attempt to connect selected pairs of roadmap nodes using some local planning method(s) as shown in figure.4; each successful connection identities can edge in the roadmap. To save space, the paths found in this stage are not recorded since they can be regenerated quickly when processing queries. The methods by which a PRM determines which (and how many) nodes to attempt to connect, and the local planner(s) selected to make those connections can crucially impact both the quality of the resulting roadmap and the running time of the PRM. Indeed, even though most PRMs greatly limit the number of connections attempted (say, to ten for each node), they still typically spend more than 95% of their pre-processing time in the connection phase. The general strategy of PRMs is to first make as many of the `easy' and `cheap' connections as possible, and then to use more sophisticated techniques to improve the roadmap's quality. For example, the PRM of [6][10] first tries to connect each node to the k (a parameter) closest nodes (as determined by some distance metric) using the common straight-line in C-space local planner, and then attempts to enhance the roadmap by sampling more nodes in identified difficult regions and/or by using more sophisticated local

## 4. Shortest Path from the PRMs

In figure.3, number of cases has been selected to show all aspects of the operation of PRM algorithm. Our algorithm provides a select a shortest path to execute. Note that it starts by assigning a weight of infinity to all nodes, and then selecting a source and assigning a weight of zero to it. When a node is (start point) selected, the weights of its neighbours is calculated to obtain a trajectory which connects the starting point to the Goal which has the least weight in our case the weight is the distance between nods as seen in figure 4. Once the selection of shortest path has made, a smoother has been added so that the robot will move the less unwanted turning, and maintain it workable for a non-holonomic robot.

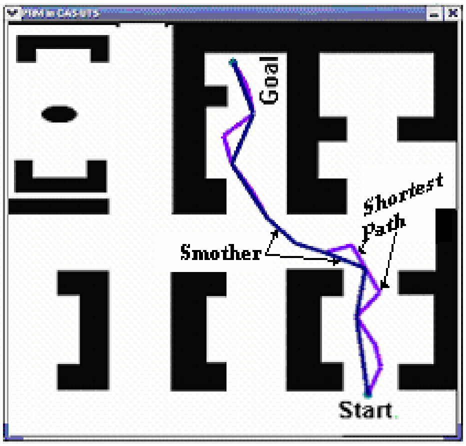


Fig.4: Shortest Path selection with smoother

Figure.5 shows the simulation of the Pioneer-2DX robot (illustrated in figure 6) in player/stage [15] with some other static and moving obstacles. GUI is made in g2 to show the robot trajectory as shown in figure 7. The thinner line shows the actual path of the robot obtained from PRM and the thicker line is its actual trajectory obtained from real time simulation. As can be seen that robot always maintains its actual trajectory with PRM path (thick line).

## 5. Obstacle Avoidance on PRM

A simple SICK LMS200 laser range finder has used for the obstacle avoidance purposes. Once the command has sent to the motor controller of the robot it starts moving along the path as illustrate in figure.7. If an obstacle obstructs the path of the robot, the Laser information made it possible for the

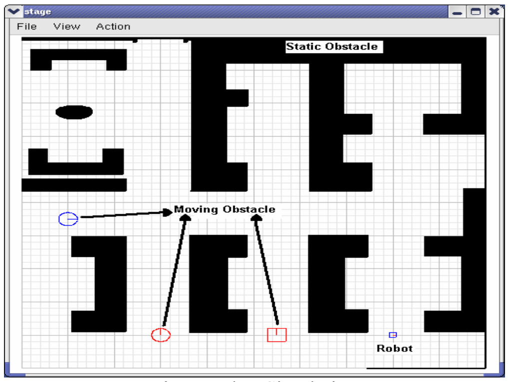


Fig.5: Robot Simulation


Fig.6: Pioneer 2DX

robot to decrease its speed and waits until the obstacle clears the path. In another case, we have changed the start and the goal position and obtained the same results of the PRM for any other start and goal position all over the free configuration area, and if the obstacle comes towards the robot, it deviates from its path and gives way to the obstacle to pass and returns back to its original path and continue the trip as can be seen in figure 8. The diverted trajectory (thicker line) shows the temporary deviation of robot from its PRM path and after obstacle avoidance it returns to its path again.

## 6. Conclusion

In this paper, we reformulated the robot path-planning problem in terms of probability spaces, measures, and computation of the transitive closure of a given local relation. We have shown that if it was possible to guess a path between two given points at random, then 'n' sets of strictly positive measure existed, so that guessing at least one point in each set would produce some path between these points. This allowed us to bind the probability of failure and reachability in terms of 'n' and successful generation of an optimal path with respect to nodes generated on the map. This algorithmic approach in the moving obstacles environment is unique in a sense that it endorse the shortcomings of the other path planning methods like potential field, Vector Field, and Histogram VFH, like local minima and oscillation of trajectory in the corridors.

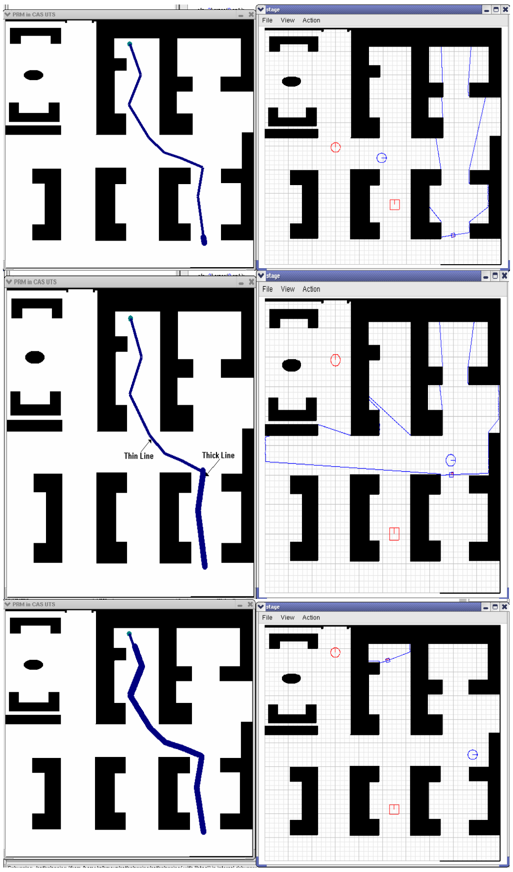


Fig. 7: Robot trajectory and movement along the path.

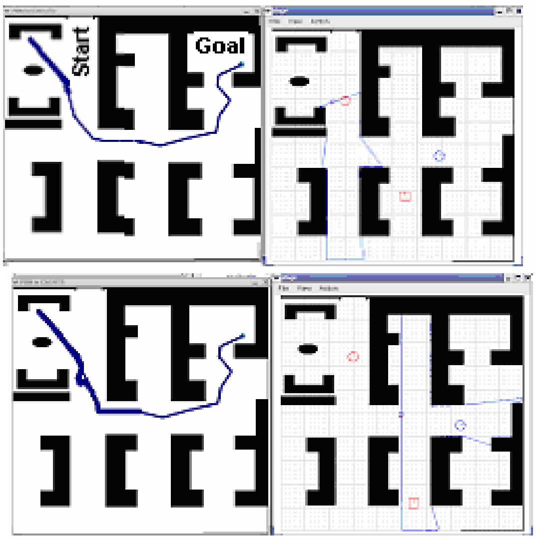


Fig. 8: Obstacle avoidance and continuing on the original path.

## 7. Future Work

We would like to extend the research of path planning using PRM in dynamic environment with the estimation theory used for the detection of Obstacle coming across the robot's way. Also efficient localisation methods, like laser beacon based can make it possible, to give us an exact robot's position when it deviates from the path or a re-planner can be introduced at that time to replan its path from the deviation, only hitch involved in this criteria is that whole system becomes so computational expensive that robot might stop in front of the obstacle or after crossing it.


## Acknowledgment

This work is supported by the ARC Centre of Excellence program, funded by the Australian Research Council (ARC) and the New South Wales State Government.


## REFERENCES


[1] H. Chang and T. Y. Li, "Assembly maintainability study with motionplanning," in Proc. IEEE Int. Conf. Robotics and Automation, 1995, pp.1012–1019.

[2] K. Gupta and A. P. del Pobil, Practical Motion Planning in Robotics. West Sussex, U. K.: Wiley, 1998.

[3] L. Kavraki, J. Latombe, R. Motwani, and P. Raghavan, "Randomized query processing in robot motion planning," in Proc. ACMSymp. Theory of Computing, 1995, pp. 353–362.

[4] P. Billingsley, Probability and Measure. New York: Wiley-Interscience,1995.

[5] R. Bohlin, "Robot path planning," Ph.D. dissertation, Chalmers Univ. ofTechnology, Goteborg, Sweden, 2002.

[6] D. Hsu, R. Kindel, J. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," Int. J. Robot. Res., vol. 21, no. 3, pp. 233–255, 2001.

[7] P. K. Agarwal, L. E. Kavraki, and M. Mason, Robotics: The Algorithmic Perspective. Natick, MA: A. K. Peters, 1998.

[8] E. Anshelevich, S. Owens, F. Lamiraux, and L. Kavraki, "Deformablevolumes in path planning applications," in Proc. IEEE Int. Conf. Robotics and Automation, 2000, pp. 2290–2295.

[9] P. W. Finn and L. E. Kavraki, "Computational approaches to drug design," Algorithmica, vol. 25, pp. 347–371, 1999.

[10] J. Xiao, and J. Xuerong, "On automatic generation of high-level contact state space," Int. J. Robot. Res., vol. 20, no. 7, pp. 584–606, 2001.

[11] L. Kavraki, "Random networks in configuration space for fast path planning," Ph.D. dissertation, Stanford Univ., Stanford, CA, 1995.

[12] L. Kavraki, M. N. Kolountzakis, "Measure Theoretic Analysis of Probabilistic Path Planning" IEEE Transaction on Robotics and Automation, vol. 20, no. 2,pp-229-242 April 2004.

[13] J. Xiao, and J. Xuerong, "Planning motion compliant to complex contact states," Int. J. Robot. Res., vol. 20, no. 6, pp. 446–465, 2001.

[14] D. Hsu, "Randomized single-query motion planning in expansive spaces," Ph.D. dissertation, Dept. Comput. Sci., Stanford Univ., Stanford, CA, 2000.

[15] R. Vaughan and B. Gerkey, "Player Version 1.4rc2 Manual",http://playerstage.sourceforge.net/doc/Player-manual-1.4-html/node1.html.