Proceedings of the 17th International Conference on Automation & Computing, University of Huddersfield, Huddersfield, UK, 10 September 2011

Grid Roadmap based Real time Path Planning

M. R. B. Bahar, H. B. Bahar & F. Hashemzadeh
Department of Electrical Engineering & Computer Engineering
Tabriz University
Tabriz, IRAN
mrbbahar@live.com, hbbahar@tabrizu.ac.ir & hashemzadeh@tabrizu.ac.ir

***Abstract*— The probabilistic roadmap (PRM) is forceful for path planning in static environments. Also PRM based methods may be employed for real time path planning in dynamic environments. These methods for real time path planning desire vast amount of time for preprocessing. To mitigate desired time for initialization, we propose a new method based on Grid Roadmap (GRM) which is an edge less roadmap. By suggested roadmap and utilizing a training method for robot manipulator, we attain a shape deformed model for obstacles that cancels our ambition for configuration space. Accordingly, grid roadmap construction will be on workspace. Finally, the planner searches for a collision free path in workspace with dynamic and shape changing obstacles.**

Keywords- Path planning, Grid Roadmap, Dynamic environment, Shape deformed model.

I. INTRODUCTION

Nowadays robots are operating in areas where there may be different obstacles and humans. To satisfy a safe working area, path planning indeed will be an essential part of robots programming. Path planning requires the creation of optimized path which avoids static and dynamic obstacles in a given workspace [1].

The PRM path planning method employs probabilistic node generation in configuration space (C-Space) [2]. This stochastic method may cause to face a free narrow area of nodes or accumulation of nodes around a particular area. In addition, neighbor nodes searching and establishment of collision free edges may lead to even more iteration and time for processing. Besides, real time path planning methods based on PRM such as Dynamic Roadmap (DRM) [3], includes steps where the nodes and edges are generated in configuration space and then mapped to workspace cells. PRM generation and mapping require enormous processing time [4]. To alleviate PRM preprocessing time we should take considerations to mitigate waste time required steps. In this situation, edge segmentation and collision checking seems to require most operation time. Attention should be taken to have another basis path planning method instead of PRM that generates an alternate simple edge less roadmap. Consequently, we will require reduced amount of time for initialization. In addition, most of path planning methods desire to maneuver on configuration space and finally the resultant path is mapped to workspace. Mapping from workspace to configuration space and then vice versa takes considerable amount of time, which is not appropriate for path planning in dynamic environments. Presented problems motivate us to look for a path planning method which is free of edges to improve path planning execution rate and operates on workspace instead of configuration space.

In this article, we propose Grid Roadmap (GRM) method for path planning which is free of edges. Also the GRM creation is on workspace with no desire for configuration space to improve path planning required time. In this work, robot manipulator trains on GRM to extract morphed model for obstacles. Training will create mapping matrix that deforms the obstacles shape and invalidates the corresponding roadmap cells. For real time path planning, obstacle morph matrix brings obstacles deformed shape and invalidates collide cells of roadmap. Then local planner searches for a collision free path on the workspace from current position to destination.

The remainder of the article is organized as in the following. In section 2 we introduce the Grid Roadmap and obstacles shape deforming technique. Section 3 describes planner. In section 4 we discuss about manipulator locating and obstacles distinction on GRM. In section 5 we implement proposed method on robot manipulator in dynamic environment. Section 6 compares GRM and DRM path planning methods. Finally, conclusions are drawn in section 7.

II. GRID ROADMAP

As we know, PRM path planning has a robust performance in static environments and also is an excellent basis for more fast lazy PRM path planning [5] and real time path planning methods such as DRM [3]. First we take a brief review on DRM method to have comparisons by our proposed method in following steps. DRM uses a pre-computed workspace mapping for fast invalidation of blocked roadmap parts. Each workspace grid cell stores a list of roadmap nodes and edges that are in collision with the cell [4]. After initialization that takes lengthy time, each obstacle on workspace invalidates relevant edges and nodes and then local planner searches for a collision free path from current position to destination on free edges and nodes. Dynamic Roadmap (DRM) preprocessing steps [3] are:

1. Generation of PRM in a free of obstacle C-space.
2. Mapping PRM edges and nodes to workspace cells.

Subsequent to DRM initialization, in real time path planning, two steps below execute until achieving destination [3]:

3. Invalidating edges and nodes via workspace obstacles.
4. Searching for a path from current position to destination on valid edges and nodes.

Steps 1 and 2 (initialization steps) require establishment of edges and nodes and then mapping them to workspace cells. Edges creation and mapping them in workspace take vast preprocessing time [4]. To reduce path planning initialization time, we propose a new method that plans an edge less roadmap with uniform distribution of cells and operates on workspace with no desire for C-Space. We call our method Grid Roadmap (GRM). GRM consists of two main sections; Initialization and planning. After preprocessing step, path planning is executed on real time. GRM initialization for a robot manipulator is described in the following subsection:

A. GRM INITIALIZATION

GRM preprocessing is described in two steps below:

1. We employ an arrange cells in workspace (GRM). These cells information is stored in a same dimension matrix called mapping matrix. Mapping matrix is 4D matrixes for 2D workspace and 6D matrixes for 3D workspace with all zero indices. Mapping matrix at last creates obstacles de-shaped form.
2. We situate manipulator on each cell of GRM. Next we distinct robot arm and assign nearest cells to arm. We utilize ∞-norm in workspace as distance function declared by (1).

$$D_{\infty}^{W}(q,q') = max_n \|q' - q\| \quad (1)$$

The nearest cells to arm for a particular location of manipulator are defined by (2).

$$\text{Near Cells} = \{q \mid \forall q',\ D_{\infty}^{W}(q,q') \le \text{ucl}\} \quad (2)$$

Where q is (i,j) index of GRM, and q' is all arm segments in workspace. ucl Or unit cell length is defined by (3) for 2D workspace.

$$ucl = \max\left(\frac{Workspace\ lenght}{Mapping\ matrix\ columns}, \frac{Workspace\ height}{Mapping\ matrix\ rows}\right) \quad (3)$$

Also ucl is defined by (4) for 3D workspace.

$$ucl = \max\left(\frac{Workspace\ lenght}{Mapping\ matrix\ columns}, \frac{Workspace\ height}{Mapping\ matrix\ rows}, \frac{Workspace\ width}{Mapping\ matrix\ depth}\right) \quad (4)$$

An example of 40*20 GRM (Right half plane) on 2D workspace `and 2-DOF robot arm are shown in Figure 1. The end effecter of arm is located on (1.7, 1.8) and near cells to arm's elbow are designated by blue circles.

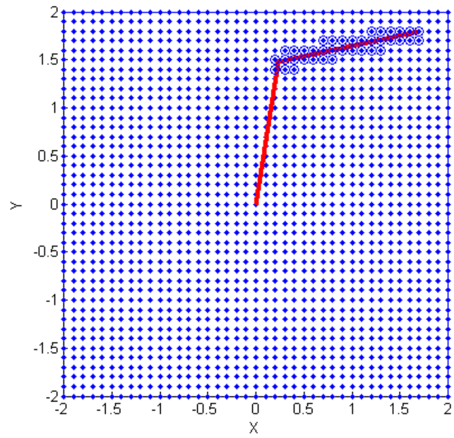


Figure 1. 2-DOF Robot manipulator on 40*20 GRM.

For Figure 1 example ucl is 0.1, links length are 1.5 and we consider 150 segments on elbow. As seen in Figure 1, obstacle zone must be out of Near Cells and colliding Near Cells will invalidate end effecter position. We define Near Cells index (i,j) and end effecter index (k,l). Then mapping matrix (i,j,k,l) index is set to one. And this step is repeated for all GRM cells to assign all Mapping matrix indices.

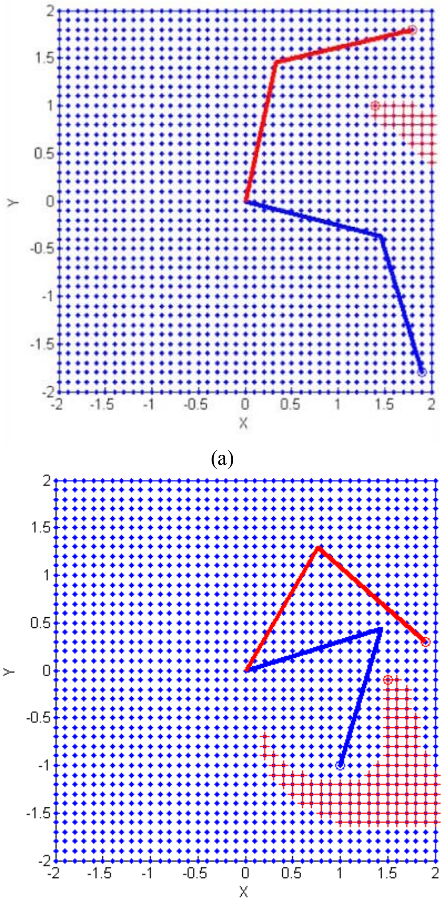
In the following subsection we clarify application of mapping matrix for invalidating workspace cells via obstacles shape deforming. Note that shape deforming is a potent tool for visual based robot control [6] and character animating [7, 8].

B. OBSTACLES SHAPE DEFORMOING

Subsequent to Mapping matrix construction, we have the invalidate cells for a particular obstacle in position (i,j) on GRM. For (i,j) obstacle index, any (k,l) that sets (i,j,k,l) index of mapping matrix to one illustrates invalid cell index on GRM. Consequently, for (i,j) obstacle index on GRM, invalid cells indices on GRM are defined by (5).

$$\text{Invalid Cells} = \{(k,l) \mid \exists (i,j), Mapping\ matrix(i,j,k,l) = 1\} \quad (5)$$

The designed examples illustrated in Figure 2 indicates an obstacle location on (1.4, 1) and (1.5, -0.1) and invalidate cells are designated by red pluses for a 2-DOF arm.



(a)

(b)

Figure 2. A spot obstacle and counterpart invalid cells. (a) Obstacle on location (1.4, 1). (b) Obstacle on location (1.5, -0.1).

A simple meaning of obstacle deformed shape is that if a spot obstacle locates on (1.4, 1) or (1.5, -0.1) as shown in Figure 2, the manipulator end effecter is unable to establish on invalid cells (red pluses). Finally a path is searched from start to end on GRM cells by avoiding invalid cells.

III. ATTRACTING AND REPELLING PLANNER

We use an attracting and repelling planner similar to potential fields [9, 10] on GRM. The near cells to destination are more weighted to attract manipulator end effecter and distant cells are less weighted. Invalid cells are assigned an immense negative weight, so no path will generate on invalid cells. Near cells to invalid cells are assigned negative weights and outlying cells are allot less negative weights and then the effect of destination and invalid cells are adjoin together to generate Cost matrix. A path is searched from start to end on more positive weighted indices on Cost matrix. Cost matrix dimension for Figure 2 example are 40*20 as the same GRM dimensions. For weight determination of Cost matrix we utilize 2-norm in workspace as distance function stated by Equation 6.

$$D_2^W(q,q') = \|q' - q\| = \left(\sum_{i=1}^{n}(q'_i - q_i)^2 \right)^{\frac{1}{2}} \quad (6)$$

Each cell of GRM is able to bond to eight adjacent cells although neglecting marginal cells. The planner starts planning from current position and explores for more positive weighted near cells on Cost matrix .The planner connects current position to more weighted near cell and makes it present cell. And again searches for more positive weighted cells around it and repeats searching until achieving destination. A generated path for Figure 2 (a) example which starts from (1.9, -1.8) to (1.8, 1.8) and avoiding invalid cells is illustrated in Figure 3.

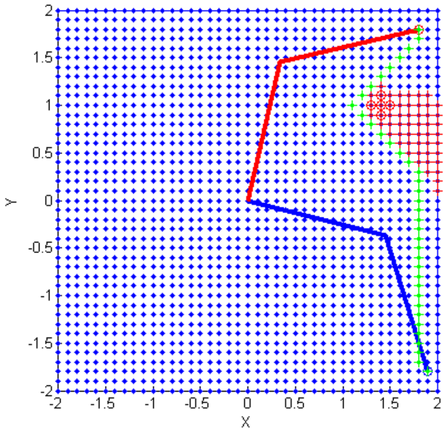


Figure 3. Generated collision free path for static spot obstacle

To make a safe area for path planning we enhance obstacle and add four pseudo spot obstacles on restrictions of original obstacle, And give more attraction gain to destination. These tasks are not crucial for static obstacles but may be applicant for dynamic environments.

A vital problem of this method is eschewing local minima's and escaping them. Hence, probabilistic methods or any other methods may be employed [11].

IV. MANIPULATOR LOCATING AND OBSTACLE DISTINCTION ON GRM

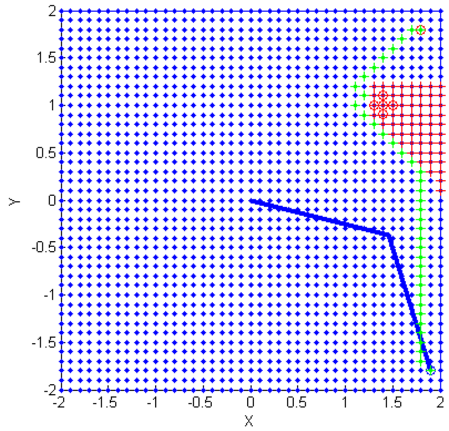
As we divide workspace to many discrete cells, manipulator start and end position and obstacle location may not be exactly on cells. For this reason we connect manipulator end effecter start and end position to nearest cells by 2-norm distance function declared by Equation 6 and then planner searches for a collision free path from nearest cell to start trough nearest cell to destination. In addition, we distinct obstacle to smaller segments and enhance each obstacle segment to 4 near cells. The enhanced shape of obstacle is defined by Equation 7.

$$\text{Enhanced Obstacle} = \{ q \mid \forall q',\ D_\infty^W(q,q') \leq \text{ucl} \} \quad (7)$$

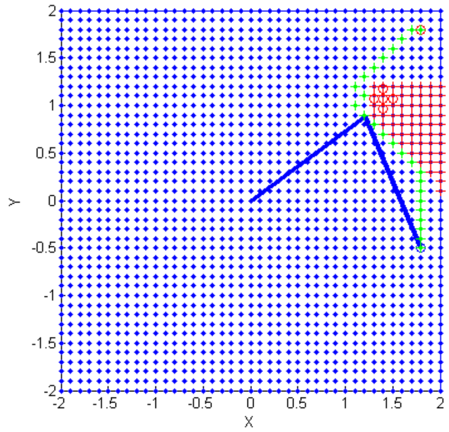
Where enhanced obstacles cells are q and GRM cells are q'. We utilize Enhanced Obstacle shape instead of original obstacle in all progresses of path planning. Addition to this distinction, for dynamic path planning we add pseudo boundary obstacles to Enhanced Obstacle for more safe area in real time path planning.

V. 2-DOF ROBOT ARM PATH PLANNING IN DYNAMIC ENVIRONEMENT

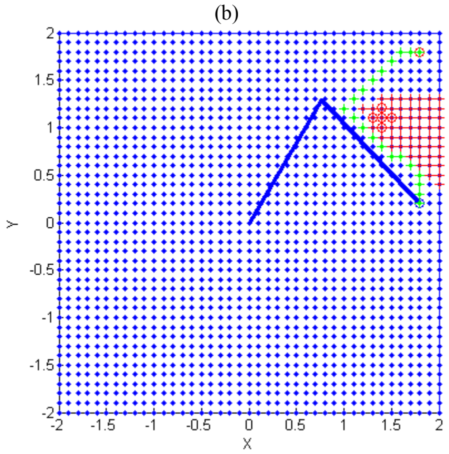
We design an example for dynamic path planning for a 2-DOF robot manipulator which starts from (1.9, -1.8) to (1.8, 1.8) location. A spot obstacle is moving from (1.4, 1) to (1.4, 1.2) during path planning. Note that we add 4 pseudo spot obstacles on boundaries of original obstacle. Planner creates collision free path for each manipulator moving steps. Results are illustrated in Figure 4.



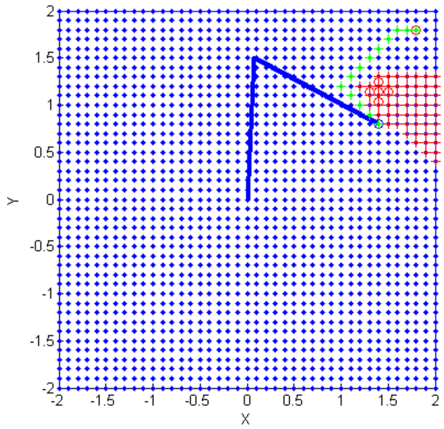
(a)



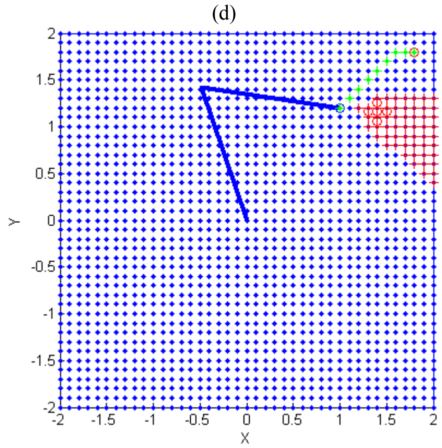
(b)



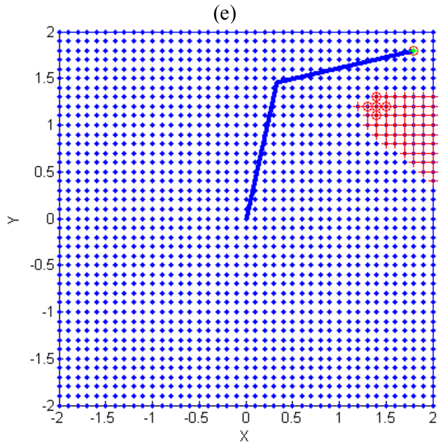
(c)



(d)



(e)



(f)

Figure 4. 2-DOF robot manipulator path planning in dynamic environment. The motion sequences are from (a) to (f).

Figure 4 example, demonstrates capability of Grid Roadmap method for real time path planning. Additionally, path smoothing can be applied on resultant path.

VI. ADVANTAGES OF GRM PATH PLANNING METHOD

Grid Roadmap (GRM) designs an edge less roadmap, accordingly the processing on edge collision checking will confiscate. By applying this method instead of DRM, initialization time for edge establishment will eliminate and result a great time save in initialization steps. A brief investigation shows that most path planning methods desire C-Space. One of the key advantages of GRM real time path planning in dynamic environments is initialization and path search of GRM, which is on workspace with no need for C-Space. Path planning only on workspace eliminates mapping to C-space and vice versa and we save a vast amount of preprocessing time. Proposed method facilitates choosing planner for path planning. Most methods like Focused D* [12], RRT [13] and RRT based methods [14, 15, 16] are able to achieve desired path on GRM.

We acquire initialization time of GRM and DRM methods on 2-DOF robot arm using MATLAB. Resultant time for DRM is given below:

- PRM generation with 800 random points in C-Space and each point connected to 4 nearest neighbors: 7.081 s.
- Mapping edges and nodes to workspace cells, with 20 points on manipulator elbow and 10*10 workspace cells: 132.032 s.

Total time for DRM initialization is: 139.113 s.

And resultant time for GRM is:

- "Mapping matrix" creation on a 40*20 GRM with 150 points on manipulator elbow: 12.61 s

Total time for GRM initialization is: 12.61 s.

And the system properties are:
Pentium IV, 3 GHz CPU and 512 MB of RAM.

We consider nodes and edges with cells are relatively proportion together to have a clear comparison. The achieved resultant times indicates that, the most desired time for DRM method caused by mapping edges and nodes to workspace which is satisfied by GRM method. Accordingly, GRM will be an advance alternate for PRM, DRM and many other PRM based path planning methods. Additionally, Path search for GRM method takes 54 ms and Random Path search for DRM method takes 535 ms. Twice methods path search execution time is suitable for real time path planning in changing environments.

VII. CONCLUSION

In this paper, we have presented a new Grid Roadmap method for path planning by comparing with DRM and PRM based path planning methods. Proposed roadmap is free of edges and consequently no edge collision checking desired. Hence, improvement in preprocessing time is successfully achieved. We have designed Grid Roadmap on workspace with no desire for configuration space. Consequently, mapping to configuration space and returning to workspace is omitted. Accordingly, we have a great alleviation on initialization time. Additionally, most planners are able to create collision free path on Grid Roadmap.

REFERENCES

[1] P. J. McKerrow, Robotics, Addison Wesley, pp. 507-515, 1992.

[2] M. W. Spong, S. Hutchinson and M. Vidyasagar, Robot Modeling and Control, Wiley, 2005.

[3] P. Leven and S. Hutchinson, "A Framework for Real-time Path Planning in Changing Environments," The International Journal of Robotics Research, Vol. 21, No. 12, pp. 999-1030, December 2002.

[4] T. Kunz, U. Reiser, M. Stilman and A. Verl, "Real-Time Path Planning for a Robot Arm in Changing Environments," International Conference on Intelligent Robots and Systems, October 2010.

[5] R. Bohlin and L. E. Kavraki, "Path planning using lazy PRM," IEEE International Conference on Robotics and Automation. Vol. 1, pp. 521-528, 2000.

[6] R. Singh, R. M. Voyles, D. Littau and N. P. Papanikolopoulos, "Shape Morphing-Based Control of Robotic Visual Servoing," Autonomous Robots– AROBOTS, Vol. 10, No. 3, pp. 317-338, 2001.

[7] Sh. Takahashi, Y. Kokjima and R. Ohbuchi, "Explicit Control of Topological Transitions in Morphing Shapes of 3D Meshes," Pacific Conference on Computer Graphics and Applications - PG, pp. 70-81, 2001.

[8] Y. Weng, W. Xu, Y. Wu, K. Zhou and B. Guo, "2D shape deformation using nonlinear least squares optimization, The Visual Computer," Vol. 22, No. 9-11, pp. 653-660, 2006.

[9] Y. K. Hwang and N. Ahuja, "A potential field approach to path planning," IEEE Transaction on Robotics and Automation, Vol. 8, pp. 23-32, February 1992.

[10] H. Xiaoxi and Ch. Leiting, "Path Planning Based on Grid-Potential Fields," International Conference on Computer Science and Software Engineering, Vol. 2, pp. 1114-1116, December 2008.

[11] M. H. Mabrouk, C. R. McInnes, "Wall Following to Escape Local Minima for Swarms of Agents Using Internal States and Emergent Behaviour," Proceedings of the World Congress on Engineering, Vol. I, July 2008.

[12] A. Stentz, "The Focused D* Algorithm for Real-Time Replanning," In Proceedings of the International Joint Conference on Artificial Intelligence, August 1995.

[13] J. J. Kuffner and S. M. LaValle, "RRT-Connect: An Efficient Approach to Single-Query Path Planning," International Conference on Robotics & Automation, April 2000.

[14] R. Pepy and A. Lambert, "Safe Path Planning in an Uncertain-Configuration Space using RRT," International Conference on Intelligent Robts and Systems, October 2006.

[15] B. Burns and O. Brock, "Single-Query Motion Planning with Utility-Guided Random Tree," IEEE International Conference on Robotics and Automation, April 2007.

[16] C. Fragkopoulos and A. Graser, "Extended RRT algorithm with dynamic N-dimensional cubic diamonds." International Conference on Optimization of Electrical and Electronic Equipment, OPTIM, 2010.