SAFE PATH PLANNING USING CELL DECOMPOSITION APPROXIMATION

Ahmad Abbadi, Vaclav Prenosil
Department of Information Technologies, Faculty of Informatics, Masaryk University
Botanicka 554/68a, 602 00 Brno. Czech Republic
Ahmad.Abbadi @ mail.com, prenosil@fi.muni.cz

Abstract: *Motion planning is an essential part in robotics domain; it is responsible for guiding the robot motion toward the goal. It generates a path from one location to another one, while avoiding the obstacles in the way. The planning modules could be configured to check the optimality, completeness, power saving, shortness of path, minimal number of turn, or the turn sharpness, etc., in addition to path safety. In this paper the cell decomposition approximation planar is used to find a safe path; the quad-tree approximation algorithm divides the workspace into manageable free areas, and builds a graph of adjacency between them. New methods are proposed to keep the robot far away from the obstacles boundaries by a minimum safe distance. These methods manipulate the weights of adjacency graph's edges. They utilize and reflect the size of free cells when planning a path. These approaches give a lower weight to the connection between big free cells, and a higher weight to the connections between the smaller cells. The planner after that searches for the lowest cost path based on these weights. The safe path in this work is the path which keeps the robot far away from obstacles by specified minimum safety distance and it bias the robot's motion to follow the bigger areas in the workspace. The shortest path is not considered. However a tradeoff between the real path cost and the safe path cost is considered when choosing the weight values.*

Keywords: Motion planning, Cell decomposition, Quad-tree, Safety path, Path planning

INTRODUCTION

Motion planning is one of the most challenging tasks in robotics fields. The main task for this problem is to find a suitable path between initial states and desire states. Many approaches were proposed to solve this problem and many efforts were done to overcome the complexity and difficulty of this problem. Some of these methods are exact and based on static environment, like Cell-decomposition algorithms [1]. Another type of these methods based on sampling-based algorithms, for example, potential field algorithm [2], or probabilistic algorithm, e.g. probabilistic road map [3], and rapidly exploring random tree [4].

From work principle point of view, some approaches are real time and require low computation cost. They based on local sensors to generate local path based on surrounding space, e.g. Bugs [5], VFH [6]. The main advantage is the tolerance to the environment changes, and main drawback of this algorithm is local minima. Other approaches are global planning which overcome local minima problem. The planners of these types produce full path form initial position to goal position. However, they have less tolerance to environment change.

In this paper we study the path safety problem in static workspace, for omnidirectional robot. The generated path in this work consider as safe if the robot translate far from obstacles by a specific distance and follow the large open areas. The cell decomposition approximation is used for path planning. The algorithm generates a graph of adjacency for free cells.

1. CELL DECOMPOSITION

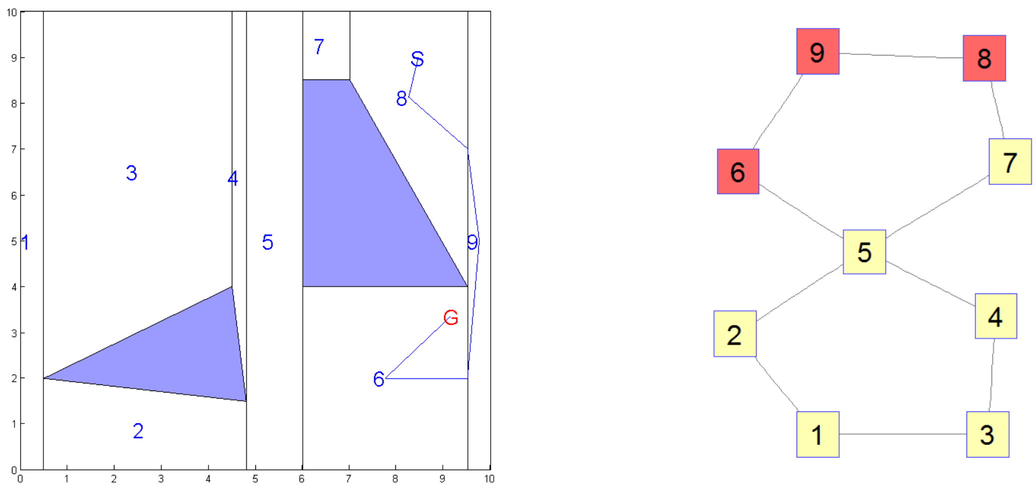
Cell decomposition algorithms are old applicable solutions for path planning. The idea of these methods is to find obstacles-free regions, and build a graph of adjacency for them [4, Ch. 6], [7]. The idea of dividing the space into manageable sections is presented in many researches. In general two categories of cell decomposition algorithms are existed; the exact cell decomposition methods and approximation methods [8].

The first category uses geometric based algorithms to explicitly determine the obstacles and build the cells [9], [10]. The union of all generated cells is equal to the free space exactly. However, finding exact free cells is not an easy task especially in high dimensions, that lead to the second category which uses the approximation techniques to divide the spaces, e.g. quad-tree, octree division, and voxel grid, etc. [1], [11].

In motion planning applications, this algorithm is utilized by dividing the free robot's workspace into smaller regions called cells. Then it builds a connectivity graph according to the adjacency relationships between the free cells. The graph's nodes represent the cells, while graph's edges represent the adjacency relations between the cells. From this connectivity graph, a continuous path can be found by following the adjacent free cells

1.1 Exact cell decomposition

The trapezoidal decomposition method or vertical cell decomposition decomposes the free space into trapezoidal and triangular cells. It draws parallel segments from each polygon's vertex in the workspace to the exterior boundary. The generated cells form the nodes of connectivity graph. The adjacent nodes in the workspace are linked to form the edges in the connectivity graph [12], [13]. The path in this graph corresponds to sequence of striped free cells.



(a) (b)

Fig. 1. Trapezoidal cell decomposition. a: the generated vertical free cells. b: the graph of adjacency which corresponding to paths between cells.
Source: own

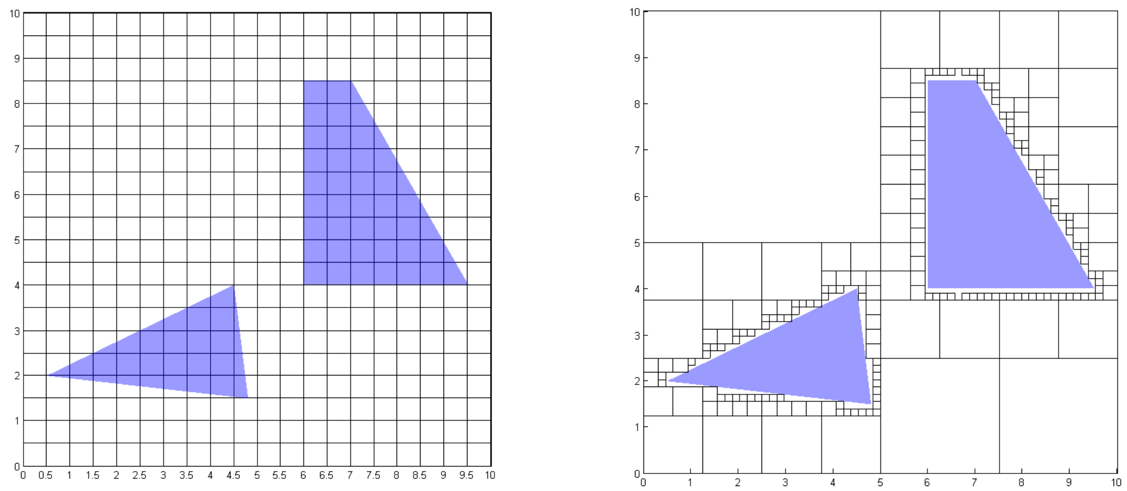
When planning query is establish, the planner finds the start and goal cells, then it searches for a path between these two cells, if a path is found the planner connect the start and goal locations through the free cells on that path [14]. Fig.1-a. shows the principle of cell decomposition planner. Fig.1-b. shows the generated graph and the corresponding path from start's cell to goal's cell.

Another example of exact cell decomposition is decomposition based on obstacles edges. This method considers each edge like a line. It finds the intersections with other edges or cells, and then it builds the free cells in the free space based on these intersections [1].

1.2 Cell decomposition Approximation

The approximation methods were proposed due to the high computation and geometric calculation which are required by exact cell decomposition. The most forward approximate cell decomposition method is voxel grid. It uses regular voxel grid or pixel grid, Fig.2-a. It excludes the cells on obstacle areas and builds a graph of adjacency for cells on free area. This method is efficient for low dimensions space. However, it generates large number of cells. This method is resolution complete; which means the algorithm's completeness depends on how fine the grid is [1], [11].

Another improvement for approximate cell decomposition was by using quad-tree decomposition. This approach uses a recursive method. It recursively subdividing the cells until one of the following scenarios occurs
1- Each cell lies completely either in a free space or in the C-obstacle region.
2- Or, an arbitrary limit resolution is reached.



(a) (b)

Fig. 2. Cell decomposition approximation. a: voxel approximation methods. b: quad-tree approximation methods
Source: own

Once a cell fulfils one of these criteria, it stops decomposing. After decomposition steps, the free path is found by following the adjacent free cells [14], [15]. This method is used in 2D [11, Ch. 14]. Fig.2-b. shows the generated cells of this method. In similar way the Octree approximate the decomposition in 3D spaces. It decomposes the cell to 8 parts [16].

The quad-tree and octree methods are resolution complete. They can work efficiently for low dimensions workspaces; three or less [17].

2. PROPOSED METHODS

In this work the path safety problem in static workspace is studied. The path is considered as safe if 1- It passes through obstacles without colliding with them. 2- If it navigates while keeping the safety distance R far from obstacles boundaries. 3- If it follows the large open areas on workspace when it possible. We utilize the cell decomposition approximation algorithm to find an approximation of free areas, and exploit the resolution feature to satisfy the minimum distance condition. The resolution R corresponds to the smallest cell's edge (box's edge). We have proposed that the robot pass through the center of the cell when execute the path; based on that assumption the R is chosen to be equal to 2*(safety distance).

In this work new methods have been proposed to plan a safe path. These methods manipulate the graph edges' weights in order to make the planner chooses the largest cells when translating toward goal position. The first approach uses equal weights for translating from one cell to another. The idea behind this proposal is to minimize the total number of cells in the path, which in consequence force the planner to use bigger cells, when searching for lower path cost.

The Second method introduces a penalty for translational between different cells size. This penalty is added to edge's weight, and it is disproportional to cells size, which means the weight of translating between the larger cells is smaller than the weight of translating between the small cells, while the weight of translating between same cells size is kept fixed. This proposal forces the planner to make the translating in large cells when it possible and at the same time keep some trade-off between making the translation in large cells, and planning a path much longer then the shortest path.

The last proposed method is very similar to the second approach in spite of it introduces disproportional penalty not only with different cells size, but also with cells have the same size. The benefit of these methods is to push the path toward large cells when it possible by adding more penalties when translating between small cells, in addition to the benefits of second approach.

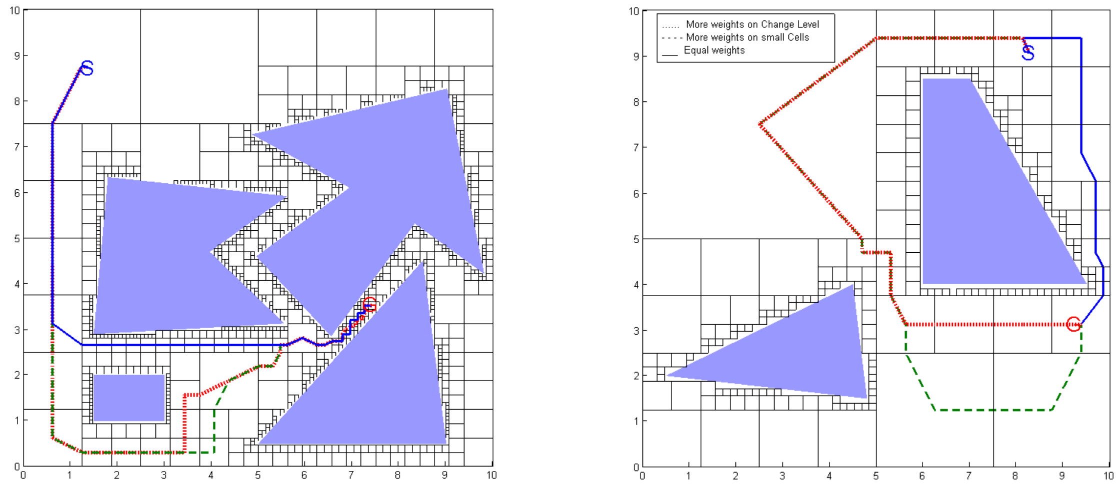
The proposed methods direct the planner to use the large cells more than small cells for planning a path, at the same time they bias the motion to translate in large cells. In practice when a robot executes the path, it follows a safe path, because this path keeps the robot far from obstacles' boundaries at least by safety distance.

3. RESULTS AND DISCUSSION

The path safety problem in static workspace is discussed in this paper. We utilize the cell decomposition approxmation algorithm to find an approximation of free areas. The generated path is considered as safe path, if the robot passes through obstacles and keeps a safe distance far from them. The value of this safety distance is given as resolution limit to the algorithm.

In the first proposed method, the weights of edges are uniformed to cost of 1, which corresponding to the cost of translating from one cell to another one, regardless to cells' size.

In the second proposed method, we associate to each cell of the free cells a level; this level disproportional to cell size. The level is used when calculating the new graph' weights. The edge' weight between two cells is set to be equal to the biggest level between these cells. I.e. if cell1 has level of 2, and cell2 is smaller and has the level of 4, The edge's weight between them has the value of max(2,4) which is 4. The translation between cells from same level is fixed to the weight of 1.



(a) (b)

Fig. 3. Results of safe path generation; a and b are the testing workspace. The solid line represents the first methods (equal weights of translation), ... line represents the second methods (disproportional penalty to change cells size), --- line represents the third methods (disproportional penalty to size of cells). The safe distance is set to be 0.1

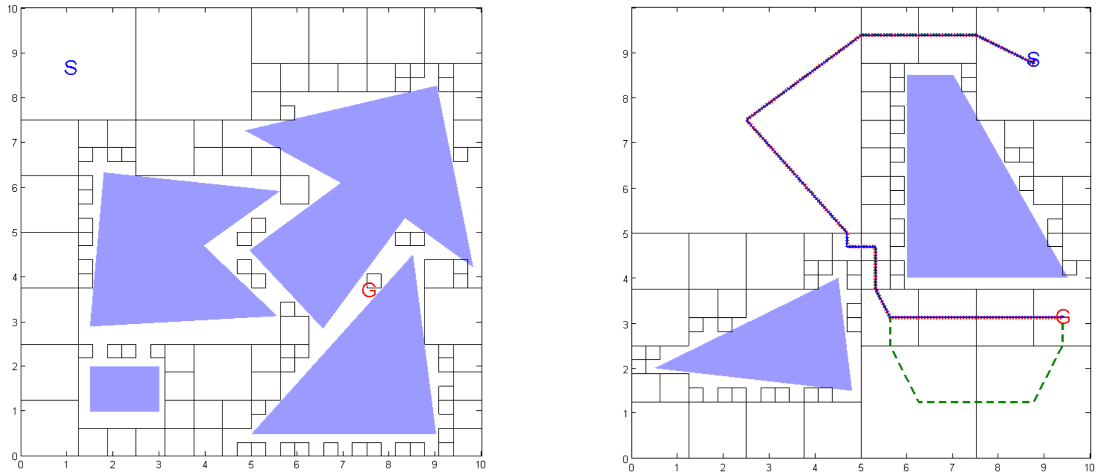
Source: own

The weights in the last proposed method are calculated in the same way as in the 2nd method, but here the transition between same cells size is vary also based on cell's level. For example the translation's weight between the cells which have level of 3 will take the value of 3.

The Dijkstra algorithm as graph search algorithm is used to find the path over the graph. The Dijkstra Algorithm finds the minimal cost of the path efficiently.

The tests are done in two workspaces with three value of safety distance {0.1,0.3,0.75}. The results are shown in Fig.3, Fig. 4., Fig. 5. respectively.

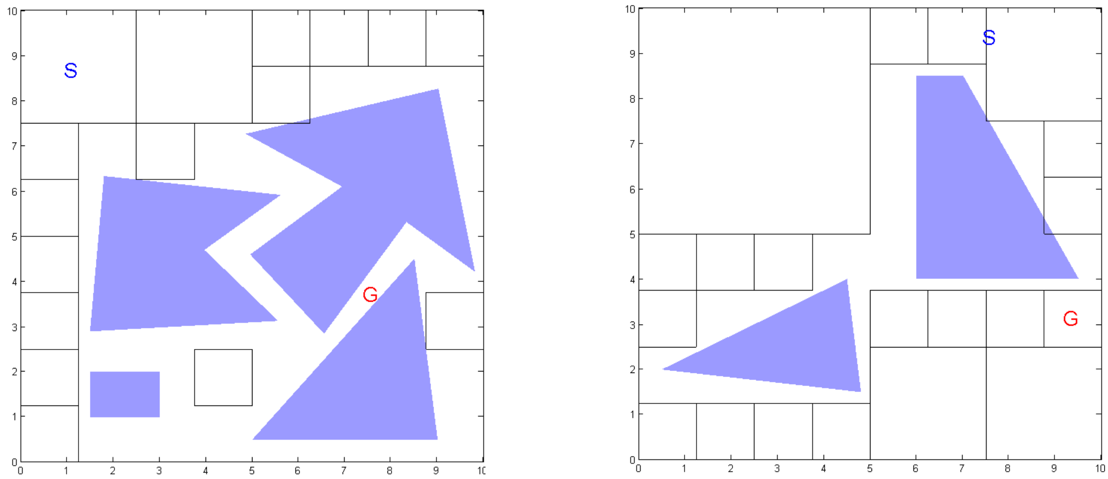
We can infer from the results that the proposed methods generate a path respect the safety distance condition. The first method try to minimize the number of cells as shown on Fig.3-(a,b), where the solid line represent the first method. The path keeps the safe distance but not follow the large areas. The second method (the dotted line) is better in this criterion. It forces the planner to go to large cells in order to minimize the cost. However, it follows the large cells but not if smaller cells are adjacent to each other; in that case the algorithm plan through these adjacent cells. The last approaches solve this drawback (dashed line), and it plans in large open regions when it possible.



(a) (b)

Fig. 4. Results of safe path generation; a and b are the testing workspace. The safe distances 0.3, the goal in a is unreachable. All methods in b have the same results.
Source: own

The Fig. 4-a shows unreachable path based on the safety distance. The same in Fig.5-(a,b). That because the algorithm excludes the collided cells with obstacles, which break the continuity of graph’s edges. Fig4-b. shows the same results for all methods.



(a) (b)

Fig. 5. Results of safe path generation. a and b are the testing workspaces, The safe distance is 0.75, the goal is unreachable in both workspaces.
Source: own

CONCLUSION

In this paper the cell decomposition approximation planar is used to find the robot’s path; the quad-tree approximation algorithm divides the workspace into manageable free areas, and builds a graph of adjacency between them. Three approaches have been proposed to plan a safe path. These methods manipulate the edges' weights in order to make the planner chooses the largest cells when translates toward goal position. That keeps the robot far from obstacles by safe distance. The proposed methods show the ability to plan the desire path. And force the planner to plan the path in the large open areas over the workspace.

LITERATURE

[1] SLEUMER, Nora H, TSCHICHOLD-GÜRMAN, Nadine. Exact Cell Decomposition of Arrangements used for Path Planning in Robotics, *Technical Report* 1999.

[2] KHATIB, Oussama. Real-time obstacle avoidance for manipulators and mobile robots. In: *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, 1985, vol. 2, pp. 500–505.

[3] KAVRAKI, Lydia E., ŠVESTKA, Petr, LATOMBE, Jean Claude, OVERMARS, Mark H. Probabilistic roadmaps for path planning in high-dimensional configuration spaces, *IEEE Trans. Robot. Autom*. vol. 12, no. 4, pp. 566–580, 1996.

[4] LAVALLE, Steven M. *Planning Algorithms*. Cambridge ; New York: Cambridge University Press, 2006. ISBN 9780511546877.

[5] CHOSET, Howie, LYNCH, Kevin M., HUTCHINSON, Seth, KANTOR, George A., BURGARD, Wolfram, KAVRAKI, Lydia E. *Principles of Robot Motion, Theory, Algorithm and Implementation*. MIT Press. 2005. ISBN 9780262033275.

[6] BORENSTEIN, Johann, KOREN, Yoram. The vector field histogram-Fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom*. vol. 7, no. 3, pp. 278–288, 1991.

[7] SEDA, Milos. Roadmap methods vs. cell decomposition in robot motion planning. In: *Proceedings of the 6th WSEAS International Conference on Signal Processing, Robotics and Automation*, 2007, pp. 127–132.

[8] LATOMBE, Jean-Claude. *Robot Motion Planning*, vol. 54. Boston: Springer US, 1991. ISBN 978-1-4615-4022-9.

[9] BROOKS, Rodney a., LOZANO-PEREZ, Tomas. A subdivision algorithm in configuration space for findpath with rotation. *IEEE Trans. Syst. Man. Cybern.*, vol. SMC-15, no. 2, pp. 224–233, 1985.

[10] SCHWARTZ, Jacob T., SHARIR, Micha. On the 'piano movers' problem. II. General techniques for computing topological properties of real algebraic manifolds. *Adv. Appl. Math.*, vol. 4, no. 3, pp. 298–351, Dec. 1983.

[11] DE BERG, Mark, CHEONG, Otfried, VAN KREVELD, Marc, OVERMARS, Marc. *Computational Geometry, Third Edit*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. ISBN: 978-3-540-77973-5.

[12] ABBADI, Ahmad, MATOUSEK, Radomil. Path Planning Implementation Using MATLAB. In: *Technical Computing Bratislava*, 2014, pp. 1–5.

[13] ABBADI, Ahmad, RADOMIL, Matousek, OSMERA, Pavel, KNISPEL, Lukas. Spatial Guidance to RRT Planner Using Cell-Decomposition Algorithm. In: *20th International Conference on Soft Computing, MENDEL*, 2014.

[14] FRIED, Joshua, DAVYDOV, Eugene, PA, Weilyn. Robotics and Motion Planning, The Intellectual Excitement of Computer Science. 28-Dec-1998. [Online]. Available: <http://cs.stanford.edu/people/eroberts/courses/soco/projects/1998-99/robotics/>.

[15] KATEVAS, Nikos I, TZAFESTAS, Spyros G, PNEVMATIKATOS, Christos G. The Approximate Cell Decomposition with Local Node Refinement Global Path Planning Method : Path Nodes Refinement and Curve Parametric Interpolation. *J. Intell. Robot. Syst.*, vol. 22, no. 3–4, pp. 289–314, Feb. 1998.

[16] CHOI, Jinwoo, CHOI, Minyong, NAM, Sang YEP, Chung, Wan Kyun. Autonomous topological modeling of a home environment and topological localization using a sonar grid map. *Auton. Robots*, vol. 30, no. 4, pp. 351–368, Dec. 2011.

[17] VAN DEN BERG, Jur P., OVERMARS, Mark H. Using Workspace Information as a Guide to Non-uniform Sampling in Probabilistic Roadmap Planners, *Int. J. Rob. Res.*, vol. 24, no. 12, pp. 1055–1071, Jan. 2005.