Motion Planning for Mobile Robot Navigation using Combine Quad-Tree Decomposition and Voronoi Diagrams

Shahed Shojaeipour[1]Sallehuddin Mohamed Haris[2]
Dept. of Mechanical & Materials Engineering
Universiti Kebangsaan Malaysia
43600, Bangi, Malaysia
shojaei@vlsi.eng.ukm.my[1] salleh@eng.ukm.my[2]

Khalil Khalili[3]
Dept. of Mechanical and Manufacturing Engineering
University of Birjand
Shohada Sq. Birjand, Iran
khkhalili@birjand.ac.ir[3]

Ali Shojaeipour[4]
Dept. of Computer Software Engineering
Islamic Azad University of Shirvan
University Ave. Shirvan, Iran
ali.shojaeipour@yahoo.com[4]

***Abstract*—this paper presents a novel method for mobile robot navigation by visual environments where there are obstacles in the workspace. The method uses a path selection mechanism that creates innovative paths through the workspace and learns to use trajectory that are more assured. This approach is implemented on motion robots which verified the shortest path via Quad-tree Decomposition (QD) and then used Voronoi Diagrams (VD(S)) algorithm we called (Q&V) algorithm. Based on the experimental data, we claim the robot's trajectory planned by Q&V algorithm is the better find and control the roadmap is completely modeled and hasn't the localization errors. We show that even small modeled obstacles can cause large used from the preplanned path. Our complementary approach of path selection decreases the risk of path following and increases the predictability of robot's behavior.**

Keywords-Motion planning; Mobile robot; Quad-tree; Q&V; Obstacle avoidance; Visual servo

I. INTRODUCTION

Today, vision based sensors such as webcams are falling in price more rapidly than any other sensor. This type of sensor is also a richer sensor than traditional ranging device, particularly since a camera captures much more data simultaneously [1]. Consequently, visual servo control of robotic manipulators has become an area of rapid research and development over the last two decades.

We consider the Cell Decomposition approach to path planning. This general method requires a complete specification of the environment and is often based on the construction of the mobile robot configuration space (*C*-Space) which reduces the mobile robot to a point. Although a localized method will basically be used, the proposed method can present an alternative approach for guidance of the point, or even joint trajectory planning if complete information about the robot is known [2].

The Cell Decomposition approach divides C_{free} into a set of non overlapping cells as shown in (Fig.1). The adjacency relationships between cells are represented by a Connectivity Graph. The graph is then searched for a collision-free path. A solution is obtained by finding the cells that contain the initial and final configurations and then connecting them via a sequence of adjacent cells [3].

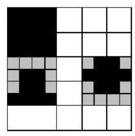


Figure 1. Cell Decomposition Model

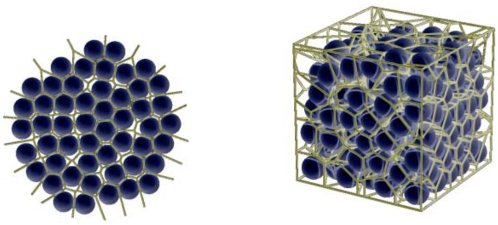
The Voronoi tessellation, originally proposed by Georgy Voronoi in 1907 [4], is a special decomposition of a metric space based on the distances of points to a specified discrete set of sites within the space. For a set *S* containing *i* number of sites in Euclidean space, the tessellation is defined by associating a cell to each site . Cell *i* contains all points that are closest to s_i and the cell boundaries are hyperplanes made up of points that are equidistant to two or more sites in *S* (Fig. 2). The Voronoi diagram (VD(S)) perfectly partitions the space, and it has found use in the sciences for solving problems that involve the assignment of space between groups of objects. In general, the interested reader should refer to the book written in [5].

Many methods have been developed and used by different researchers to compute Voronoi tessellations. For example, in [6], the Voronoi octree data structure was introduced to generate generalized 3D Voronoi diagrams. In [7], morphological operations were used to transform image data obtained from sensors into its corresponding VD(S) where the skeletal lines represent obstruction free paths.

There also exist a number of software packages readily available for computing the Voronoi tessellation. These include, for example, the software package [8] which can

978-1-4244-5586-7/10/$26.00 © 2010 IEEE

compute VD(S) in arbitrary dimensions and the Voronoi function in MATLAB. The program in [9] which is well-known for mesh generation via Delaunay Triangulation also computes Voronoi tessellations.



A B

Figure 2. (A) the 2D Voronoi tessellation (B) the 3D Voronoi tessellation

Mesh computation codes can compute VD(S) of single objects which, given a set of points, will return the complete mesh dividing the space containing the set of sites into a mesh of cells as seen in figure 3. However, in practical applications, it is more natural to associate a single VD(S) with each particle, and compute them separately. This makes it easier to compute just a subset of Voronoi meshes, or to tailor the computations to handle special cases and complex boundary conditions. This also makes it straightforward to compute mesh-based statistics, such as mesh volumes, or the number of faces per mesh [10], [11].

II. PATH PLANNING USING VORONOI DIAGRAMS

A. *Generation of Voronoi Diagrams*

VD(S) are constructed by first performing the Delaunay Tessellation which is regarded as the dual to Voronoi Tessellations. Firstly, any two sites *p,q* for which there exists a circle *C* that passes through *p* and *q* and doesn't contain any other site of *S* in its interior or boundary, are connected by a line segment. The set of such line segments form the edges of the Delaunay Tessellation DT(s), called Delaunay edges [10][12][13]. Now, bisection of the Delaunay edges by another set of line segments results in the formation of Voronoi cells where each cell contains one site enclosed by the line segments forming Voronoi edges.

An example is depicted in figure 3, where VD(S) is depicted by solid lines and DT(S) by dashed lines. Note that a Voronoi vertex need not be contained in its associated face of DT(S). The sites *p, q, r, s* are co-circular, giving rise to a Voronoi vertex *v*. Consequently, its corresponding Delaunay face is boarded by four edges [12].

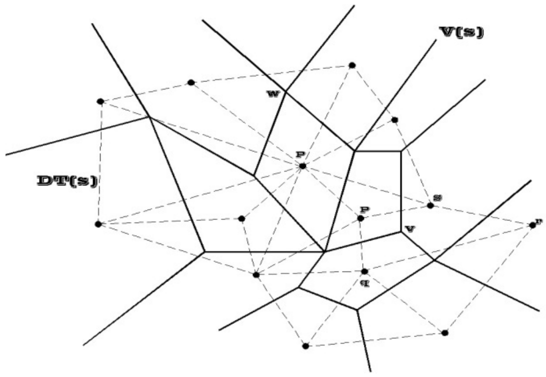


Figure 3. Voronoi Diagram and Delaunay Tesselation

B. *Motion Planning using VD(S)*

Application of Voronoi diagrams to mobile robot motion planning has been discussed in [12]. The concept is reproduced in the following excerpt.

Suppose that for a disc-shaped robot centered at some start point, *s,* a motion to some target point, *t*, must be planned in the presence of *n* line segments as obstacles.

We assume that the line segments are pair wise disjoint, and that there are four line segments enclosing the scene, as shown in figure 4. While the robot is navigating through a gap between two line segments, l_1 and l_2, at each position *x* its "clearance ", i.e. its distance

$$d(x,l_i)=\min\{d(x,y);y\in l_i\}$$

to the obstacles, should be a maximum. This goal is achieved if the robot maintains the same distance to either segment. In other words, the robot should follow the bisector $B(l_1,l_2)$ of the line segments l_1 and l_2 until its distance to another obstacle gets smaller than $d(x,l_i)$.

Roughly, this observation implies that the robot should walk along the edges of the Voronoi diagram VD(S) of the line segments in $S = \{l_1,\ldots\ldots,l_n\}$. This diagram is connected, due to the four surrounding line segments.

If start and target points are both lying on VD(S), the motion planning task immediately reduces to a discrete graph problem: After labeling each edge of VD(S) with its minimum distance to its two sites, and adding *s* and *t* as new vertices to VD(S), a breadth first search from *s* will find, within $O(n)$ time, a path to *t* in VD(S) whose minimum label is a maximum. If this value exceeds the robot's radius, a collision-free motion has been found.

If the target point, *t*, does not lie on VD(S), we first determine the line segment $l(t)$ whose Voronoi region contains *t*. Next, we find the point $z(t)$ on $l(t)$ that is closest to *t*; see figure 4. If its distance to *t* is less than the robot's radius then the robot cannot be placed at *t* and no motion from *s* to *t* exists. Otherwise, we consider the ray from $z(t)$ through *t*. It hits a point *t'* on VD(S) which serves as an intermediate target point.

Similarly a point *s'* can be defined if the original start point, *s*, does not lie on VD(S) [12].

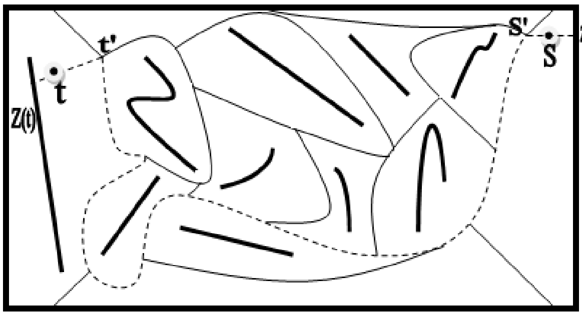


Figure 4. The Motion Planning in Voronoi Diagrams Algorithm

III. QUAD-TREE DECOMPOSITION

Quad-tree Decomposition (QD) is a technique that divides an image into 2D homogeneous regions. Typically, the image is first divided to form quadrants. Each quadrant may be full, partially full or empty. A partially full quadrant is then recursively subdivided into smaller quadrants until all quadrants are homogeneous, or when some pre-determined cutoff level is reached [14][15][16].

The use of quad-tree decomposition for path planning roadmaps has been proposed by several researchers. Among these, [17] used the Markov Decision Process to model the robot's interactions with its environment. The optimal path were then determined using Value Iteration [18] and Policy Iteration [19] algorithms. In [20], the path obtained from Cell Decomposition using quad-trees was further improved using Local Node Refinement, and then smoothed using cubic splines.

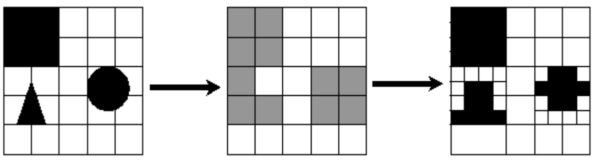


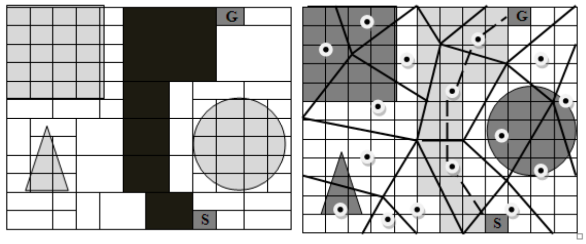
Figure 5. Quad-tree Decompositiofn o 2D environment

IV. THE PROPOSED Q&V ALGORITHM

We now present our proposed Q&V algorithm. The strategy is to firstly divide the workspace into square-shaped cells using the QD algorithm. In principle, the robot needs to keep away from obstacles, hence none empty cells are eliminated and what remains are possible paths from $Start_{point}$ to $Goal_{point}$ (seen in figure 6A).

In next step, the VD(S) algorithm is performed only on the section of workspace available from the previous process. Then, the *p* points of the Voronoi Diagram that are contiguous to each other are connected. However, only points from the region obtained through the QD process are selected. Therefore we can find the shortest path.

This novel algorithm is developed since, in the QD algorithm, there exists some possibility that the robot may get trapped in a snare; while the VD(S) algorithm may perform too slowly when finding the shortest path in a complex workspace, or the algorithm might also end up in a local minimum. This novel algorithm is capable of performing faster in finding the optimal path (seen figure 6B).



A B

Figure 6. Strategy Q&V Algorithm (A) Quad-tree Decomposition then (B)Voronoi Diagrams only for

A. *Advantages and limitations of this strategy:*

This new strategy in robot motion planning has several advantages as follows:

a) Avoids the local minimum: With two existing algorithms to remove problem, the robot would be able to avoid falling into the local minimum.

b) High-speed search of path to target: High accuracy in computation to find the shortest path since the two algorithms complement each other. Each algorithm strengthens the other's weaknesses, hence execution speed is increased.

c) Optimization of path: Capable of finding the nearest possible to the optimized path.

d) Use in 2D and 3D workspace: 3D algorithms exist for both algorithms. Hence, the method could be extended to 3D workspace.

Currently, limitations of the Q&V algorithm are as follows:

a) Cannot be used online: Currently, this algorithm is not for online implimentation. In future, the algorithms will be further developed for online use.

b) Noise from complex workspace: For some paths that have complex shapes; or in cases where the robot is unable to distinguish different colors within the workspace, difficulties in processing data may arise. This problem will be solved with further improvements to the existing algorithms.

V. MOTION PLANNING USING Q&V ALGORITHM

A. *Define $Start_{Point}$ and $Goal_{Point}$*

In general, if we assume point obstacles and a starting point; we can use Q&V for navigation. One can think of Q&V as a graph, made up of edges and linear vertices. To go from $Start_{point}$ to $Goal_{point}$, we first find the nearest points on the Q&V map to ($Start_{point}$ to $Goal_{point}$) and mark this as ($Start^*_{point}$ to $Goal^*_{point}$). The application uses a standard graph type algorithm (QD) to traverse the linear vertices and edges of the graph from ($Start^*_{point}$ to $Goal^*_{point}$).

Then, suppose the obstacles aren't a point and the robot isn't a point, so the two-dimensional region in the robot moves is having makes and other style of quad-tree, each of which can be represented via the obstacle [21].

B. Methods:

- Mark the boundaries of the shape of all obstacles with the highest number of points that result from subdividing each side of the original quad-tree into small segments (seen in Fig. 7).
- Compute the Q&V for this collection of points.
- Once this complicated Q&V is made, erase the quad-tree edges which have one or two $goal_{point}$ paths crossing any of the obstacles.
- The remaining quad-tree is the best path obtained from Q&V, avoiding the original obstacles on the roadmap.

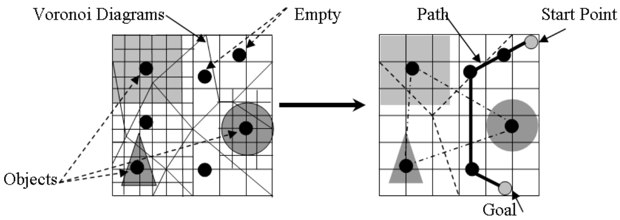


Figure 7. Q&V Algorithm

To take into account that the robot is not point sized, we can find critical lines and critical points on the Q&V map. These are locations where the Q&V path has a local minimum. At these locations, we need to see if the robot diameter fits so as to pass through the path. If the diameter is greater than the critical line length, then the robot will not be able to pass. For this purpose, we use the maximum diameter of the robot over all rotations.

VI. CONCLUSIONS AND FUTURE WORK

A novel concept for mobile robot motion planning has been presented, where the mobile robot navigates from $Start_{point}$ to $Goal_{point}$. This navigation concept is made up of two parts obtained from existing methods: the Quad-tree Division algorithm (QD) and the Voronoi Diagram VD(S) algorithm. Each of these methods may produce some errors. Since the two complement each other, this novel method is capable of overcoming these weaknesses. The optimal path (shortest path) can be found using our proposed Q&V algorithm. Generally, in the first step, the QD algorithm is run to find all possible paths to the target. Next, the VD(S) algorithm is run only for the paths defined from the first step. Hence, the Q&V algorithm obtains the best path, improving from the shortest path initially found through QD. In future, this algorithm will be implemented on an actual robot workspace. The results will then be compared to those obtained via other methods.

ACKNOWLEDGMENT

This research was supported by the Ministry of Science Technology and Innovation, Malaysia, Grant No. 03-01-02-SF0459.

REFERENCES

[1] J. Campbell, R. Sukthankar, I . Nourbakhsh , and A. Pahwa, "A robust visual odometry and precipice detection system using consumer-grade monocular vision. Proc," Int. Conf. on Robotics and automation, Barcelona, Spain, 2005.

[2] J.C. Latombe, "Robot Motion planning. Kluwer Academic publishers. London. 1991.

[3] Y. K. Hwang, N. Ahuja,"Gross motion planning - a survey. ACMComp," Surveys, 24(3), pp. 219-29, 1992.

[4] G. Voronoi. Nouvelles applications des param`etres continus `a la theorie des forms quadratiques, Journal f¨ur die Reine und Angewandte Mathematik 133, pp. 97–178, 1907.

[5] A. Okabe, Barry Boots, K. Sugihara, and S. N. Chiu," Spatial tessellations: concepts and applications of Voronoi diagrams," John Wiley & Sons, Inc., New York, NY, 2000.

[6] I. Boada, N. Coll, N. Madern and J.A. Sellares," Approximations of 3D Generalized Voronoi Diagrams," EWCG 2005, Eindhoven, March 9-11, 2005.

[7] S.Garrido, L.Moreno, M. Abderrahim and F. Martin," Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching," Int. Conference on Intelligent Robots and Systems Oct, 2006.

[8] Qhull code for convex hull, Delaunay triangulation, Voronoi diagram, and halfspace intersection about a point, http://www.qhull.org/.

[9] Triangle: A two-dimensional quality mesh generator and Delaunay triangulator, http://www.cs.cmu.edu/_quake/triangle.html

[10] E. Masehian and G. Habibi," Robot Path Planning in 3D Space using Binary Ineger Programming," International Journal of Mechanical System Science and Engineering ISSN 1307-7473, 2008.

[11] G. Habibi, E. Masehian, M.T.H. Beheshti," Binary Integer Programming Model of Point Robot Path Planning," The 33rd Annual Conference of the IEEE Industrial Electronics Socitey IECON ,Nov 5-8 , 2007.

[12] F. Aurenhammer, R. klein," Voronoi Diagrams," Resarch Concerning Voronoi diargrams, www.pi6.fernuni-hagen.de/publ/tr198.pdf

[13] S.W. Bae and K.Y. Chwa," Shortest Paths and Voronoi Diagrams with Transportation Networks Under General Distances," Springer – Verlag Brilin Heideberg, 2005.

[14] E. Shusterman, F. Meir,"Image Compression via Improved Quad-tree Decomposition Algorithms," IEEE Transaction on Image Processing, VOL.3, NO.2, Mar 1994.

[15] R. Mathew and D.S. Taubman," Motion Modeling with Sparate Quad-Tree Structures for Geometry and Motion,"IEEE MMSP, 2008.

[16] J.D. Foley, A. van Dam, S.K. Feiner and J.F. Hughes, Computer Graphics Principles and Practice, 2nd edition in C. Addison-Wesley 1996.

[17] B. Julien, A. Olivier, f. Thievrry,"Robust motion planning using markov Decision process and quad-tree decomposition," 2004.

[18] R. Bellman, J. Holland and R. Kalaba, On an Application of Dynamic Programming to the Synthesis of Logical Systems. ACM Press, 1959, vol. 6, no. 4.

[19] R.A. Howard, Dynamic Processing and Markov Processes. Cambridge, Massachussets, USA: MIT Press, 1960.

[20] N. Katevas, S. Tzafestas and Ch. Pnevmatikatos,"The Approximate Cell Decomposition with local node refinement global path planning method: path nodes refinement and curve parametric interpolation," Journal of Intelligent and Robotic Systems 289-314, 1998

[21] J. Baltes, and J. Anderson,"Flexible Binary Space Pratitioning Rescue," IEEE/RSJ International Conference on Intelligent Robots and Systems,(IROS 2003),2003.