Obstacle-Avoidance Path Planning Based on Delaunay Triangulation

Gene Eu Jan, *IEEE Member*, Ming Che Lee, *IEEE Member*, Chaomin Luo, *Senior Member* and Wei Fan Chiang

***Abstract*— In the Euclidean plane with obstacles, the main purpose of the shortest path algorithm is to find the best path from the start point to the end point. For the shortest path problem with obstacles, this research proposes a Delaunay Triangulation-based path planning algorithm. Based on the Delaunay Triangulation, this research constructs triangulations of the free space (obstacle avoidance space), and applies the triangle centroid to plan the shortest free path from the origin to the destination. Compared with other traditional algorithms, the algorithm proposed in this study has O(nlog n) time complexity and can plan obstacle-avoidance free paths in a very short time, where n is the number of obstacles.**

I. Introduction

Determining shortest path of points on the Euclidean plane with obstacles is a very important issue. Among them, the most interesting is to connect two vertices to construct a path and find the best path. This problem involves whether the path between two vertices passes through obstacles, and through calculation to determine which path is the smallest sum of continuous line segments. The shortest path algorithm was first applied to automatically find the shortest path between two locations in real life. This type of method has been applied to path planning of robots and navigation vehicles. In addition, it is also applied to autonomous underwater vehicles (AUV) [1], mountaineering, operation research [2,3], piano mover's problem [4], plant equipment layout, vessel navigation [5,6,7] and other fields [8,9,10]. Dijkstra's algorithm [11,12], A* [13,14] are two well-known algorithms to solve the shortest path problem in the literature. Furthermore, Potential Field, Roadmaps, and Cell Decomposition are the three major approaches of the shortest path problem in the Euclidean plane with obstacles [15, 16].

Dijkstra's algorithm is the simplest and easiest approach to implement the shortest path problem. It can be used for weighted directed or undirected connected graphs. Subsequent versions are also widely used in transportation, route planning and other fields [17-20]. The A* algorithm uses a heuristic function to estimate the distance between the start point and the end point in a straight line during the calculation process, and select the road section with a better estimate to enter the calculation. A* performs better while the nodes are evenly distributed. The main approach of cell decomposition is Trapezoidal Decomposition [21], which vertically cuts the vertices of obstacles into mutually exclusive spaces. The method is very dependent on the configuration of the plane space, and the process of constructing paths is not efficient. In this paper, we decompose the free space by Delaunay Triangulation, then apply Dijkstra's algorithm for the global routing and finally obtained the free shortest path in $O(n\log n)$ time.

Three well-known approaches of roadmaps are: Visibility Graph, Voronoi diagram and Delaunay Triangulation (DT). Visibility Graph was proposed by Lozano-Perez and Wesley [22]. The path is composed of line segments connected by nodes on a full plane. It is the best solution to find the shortest path. However, the time complexity is $O(n^3)$. With the number of nodes increase, its execution time will increase significantly. In order to reduce the time and space complexity of the full-plane Visibility Graph, Rohnert [23], Ghosh and Mount [24], and Pocchiola and Vegter [25] proposed methods to reduce the line segments connected by nodes. The time complexity can be reduced to $O(n^2)$. The method to further improve the Visibility Graph is to adopt the Local Visibility Graph, respectively proposed by Zhang *et al*. [26], Gao *et al*. [27], and Li *et al*. [28]. The method adopted is to select the range with a circle, and then use Visibility Graph to find the shortest path.

The sections of this article are described as follows: section II describes DT related research, section III is the proposed algorithm, section IV is the experimental result, and the final gives the conclusion.

II. CELL DECOMPOSITION

Five major approaches of the cell decomposition are addressed as follows.

A. *Trapezoid Decomposition*

Trapezoid Decomposition, also known as vertical cell decomposition [29,30], is widely used in robot motion. The method is to separate the free space with trapezoidal units, where each trapezoid has vertical parallel sides. The idea of constructing this subdivision is to draw a vertical scan line from each vertex. If the extension line hits an obstacle, the extension will stop. In order to reduce the time complexity, first sort them according to the *x* coordinates of all vertices, and then use the scan line moves in the sorted vertex array. After the algorithm processes each vertex, the sequence that intersects with the obstacle is processed (as shown in Figure 1-a).

Gene Eu Jan is with the Tainan National University of the Arts, Principal Office, 66, Daqi, Guantian Dist., Tainan City 72045, Taiwan (R.O.C.) (email: gejan@mail.ntpu.edu.tw)

Ming Che Lee is with the Department of Computer and Communication Engineering, Ming Chuan University, No.5 De Ming Rd., Gui Shan District, Taoyuan County 333, Taiwan (R.O.C) (email: leemc@mail.mcu.edu.tw)

Chao min Luo is with the Department of Electrical and Computer Engineering, Missisipi State University, P. O. Box 9730, Mississippi State, MS 39762, U.S.A (e-mail: chaomin.luo@ece.msstate.edu).

Wei Fan Chiang is with the Department of Computer Science and Information Engineering, National Taipei University, New Taipei City, Taiwan (R.O.C) (email: s710983117@gm.ntpu.edu.tw)

B. Delaunay Triangulation

This section introduces the widely used Delaunay Triangulation (DT) method [31-33]. The main concept of DT is to input a group of points and connect these points to form a triangle with special attributes. The inner angle of the triangle divided by DT has the maximum angle, and it is guaranteed that the circumcircle of each triangle does not contain any other input points. However, in some cases, Delaunay triangulation cannot be used, because obstacles will be divided during the division, and the formed triangle will be partly in the free space and partly on the obstacles. To overcome this problem, [34,35] developed a special Delaunay triangulation, retaining some predefined line segments (as shown in Figure 1-b).

C. Polygon Decomposition

Polygon Decomposition achieves the purpose of spatial segmentation by extending each edge of the obstacle to the boundary. The linear equation of the obstacle edge is found in the literature and stored in the matrix *P* and the vector *Q*. Each line can divided the whole space into two halves, and then through linear inequality to find whether the interior is empty intersection. If it is an empty intersection, find the vertices. In polygon triangulation, any two adjacent polygons will share a facet, which makes the connection relationship established relatively simple, as shown in Figure 1-(c).

D. Quadtree Decomposition

Quadtree Decomposition [36] is an approximate unit subdivision, which covers free space with arbitrary precision. This method is inspired by the quaternary tree, which is mainly used in computer science to organize data or decomposition of compressed images . A quaternary tree is a tree whose nodes are leaves or have four subtrees. Each node in the quaternary tree corresponds to a rectangle in the map. It is divided into four rectangles in the space. If any of them are occupied, continue cutting down into four rectangles until all the rectangles are not occupied, as shown in Figure 1-(d).

E. Raster Decomposition

Raster Decomposition [37-39] is a process of turning geometric figures into unit images. The process consists of two parts. The first part determines which grid areas in the coordinates are occupied by obstacles and the second part assigns a binary value to each area to distinguish the free space and obstacles, as shown in Figure 1(e).

III. ALGORITHM AND ILLUSTRATION

This section introduces the proposed path planning algorithm based on Delaunay triangulation. The algorithm is described as follows.

3.1 Algorithm:

INPUT: Start point, end point, vertex coordinates of polygonal obstacles.

OUTPUT: The shortest free path from the start point to the end point obtained after the space is divided by DT.

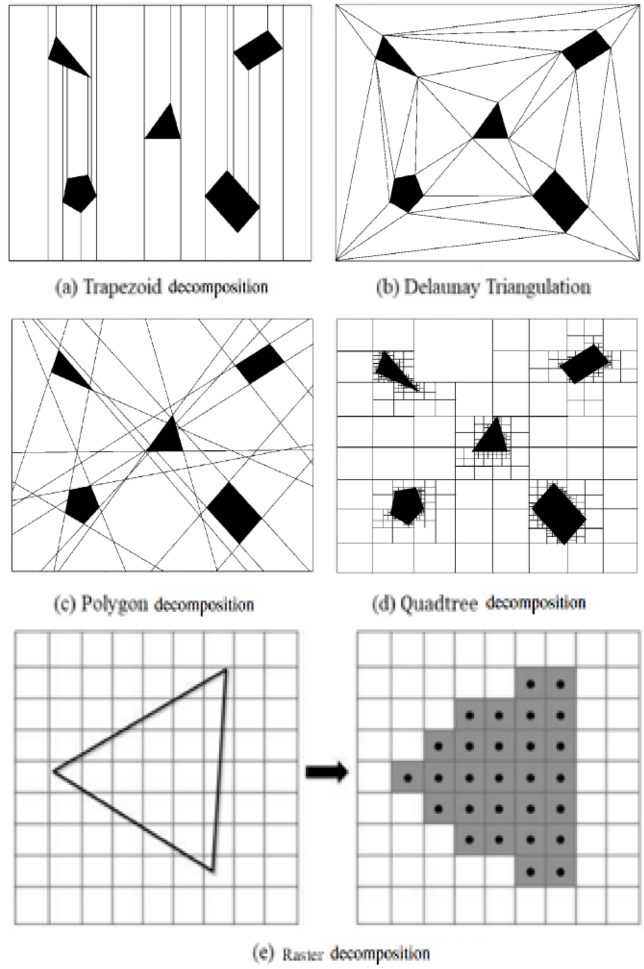


Figure 1. Different Types of cell Decompositions

Step 1: Call constrained Delaunay triangulation (polygons) to perform spatial division.

Step 2: Calculate the centroids of triangles.

Step 3: Connect the centroid points of all adjacent free spaces to establish a connection graph.

Step 4: Apply the Dijkstra’s algorithm to find the shortest path in the connected graph obtained by Step 3.

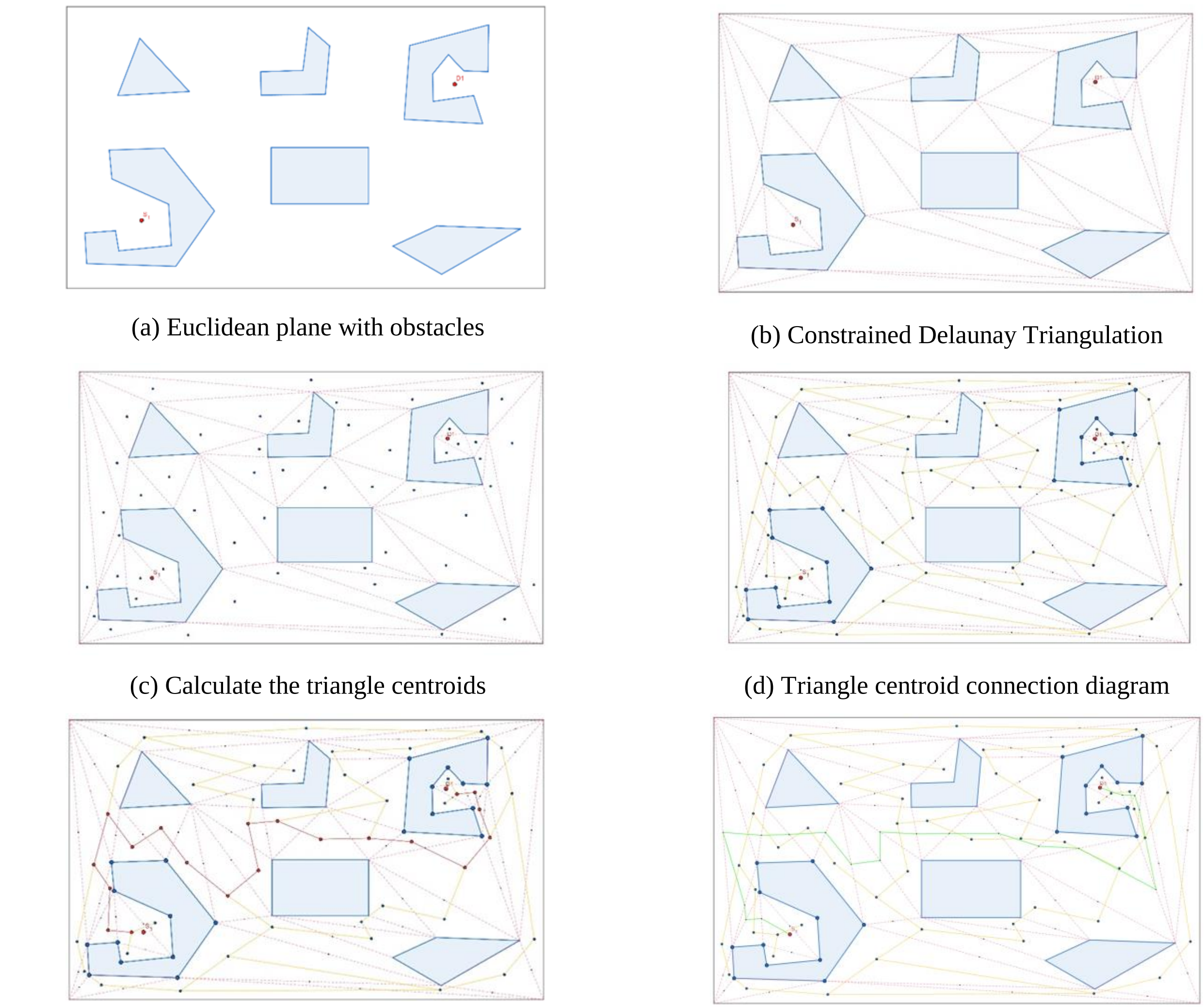
Step 5: The midpoint of the line segment after DT is generated.

Step 6: Find the free path through the midpoints of line segments of triangles in the path obtained in step 4.

END{Algorithm}

3.2 Illustration of algorithm

The illustration of the algorithm are shown in Figures 2. Figure 2(a) shows the open space with irregular obstacles with varying numbers. Then perform DT calculations on the start point, end point, end point and obstacle. Subsequently, calculate the triangle centroid, as shown in Figures 3(c) and (d). Then connect the centroid points and apply Dijkstra's algorithm to calculate the shortest path between the centroid points. Finally, call the Path Shortening function to re-plan the route. The comparison of different algorithm in Euclidian space is depicted in table 1.



(a) Euclidean plane with obstacles

(b) Constrained Delaunay Triangulation

(c) Calculate the triangle centroids

(d) Triangle centroid connection diagram

(e) Apply Dijkstra's algorithm to generate the shortest path

(f) Free path in green

Figure 2. Illustration of algorithm

IV. CONCLUSIONS

This research proposes a path planning algorithm based on Delaunay triangulation, then apply Dijkstra's algorithm for the global routing and finally obtained the shortest free path in $O(n\log n)$ time. Compared with the traditional path planning algorithms of cell decompositions, the method proposed in this study can either reduce the required time and space complexities or obtain the shorter free path.

After simulation, the proposed algorithm can effectively plan the moving route of objects with a given plane and obstacles. In the future, it can be applied to different fields such as mobile robots, bypass planning and traffic navigation.

REFERENCES

[1] Paull, L., Saeedi, S., Seto, M., and Li, H. (2012). Sensor-driven online coverage planning for autonomous underwater vehicles. *IEEE/ASME Transactions on Mechatronics*, 18(6), 1827-1838.

[2] Gallo, G., and Pallottino, S. (1988). Shortest path algorithms. *Annals of operations research*, 13(1), 1-79.

[3] Golden, B. (1976). Shortest-path algorithms: A comparison. Operations Research, 24(6), 1164-1168.

[4] Gene Eu Jan, T. Y. Juang, J. D. Huang, C. M. Su, and C. C. Cheng, "A Fast Path Planning Algorithm for Piano Mover's Problem on Raster," *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Monterey, U.S.A., pp. 24~28 , July 2005.

[5] Gene Eu Jan, Ming-Bo Lin and Yung-Yuan Chen, "Computerized Shortest Path Searching for Vessels," *Journal of Marine Science and Technology*, Vol. 5, No. 1, June 1997, pp. 95-99.

[6] Ki-Yin Chang, Gene Eu Jan and Ian Parberry, "A Method for Searching Optimal Routes with Collision Avoidance on Raster Charts," *Journal of Navigation*, Vol. 56, No. 3, pp. 371~384, 2003.

[7] Ki-Yin Chang, Gene Eu Jan, C. M. Su and Ian Parberry, "Optimal Interceptions on Two-Dimensional Grids with Obstacles," *Journal of Navigation*, Vol. 61, pp. 31~43, Jan. 2008.

[8] Wu, Y., Sun, D., Huang, W., and Xi, N. (2012). Dynamics analysis and motion planning for automated cell transportation with optical tweezers. *IEEE/ASME Transactions on Mechatronics*, 18(2), 706-713.

[9] Chi-Chia Sun, Gene Eu Jan, Shao-Wei Leu, Kai-Chien Yang and Yi-Chun Chen, "Near-Shortest Path-Planning on a Quadratic Surface with $O(n \log n)$ Time," *IEEE Sensors Journal*, Vol. 15, No. 11, pp. 6079- 6080, Nov. 2015.

TABLE I. COMPARISON OF DIFFERENT ALGORITHMS, WHERE *n* DENOTES THE NUMBER OF OBSTACLES

Euclidean	Algorithms						
	Cell decomposition			Potential field	Roadmap		
	Trapezoidal decomposition	Polygon decomposition	Ours		Visibility Graph	Voronoi Diagram	Delaunay Triangulation
Time Complexity	$n \log n$	n^2	$n \log n$	Not available	n^2	$n \log n$	$n \log n$
Space Complexity of constructing connected graph	n	n^2	n	Not available	n^2	n	n
Find path if one exists?	yes	yes	yes	yes	yes	yes	yes
Is the path free or semi free?	free	free	free	free	semi	semi	semi

[10] Gene Eu Jan, and Ki-Yin Chang, "An Improved Lee's Algorithm on Electronic Maps," 2002 *International Computer Symposium*, National Dong Hwa Univ., Hualien, Taiwan, Dec. 2002, pp. 776~786.

[11] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, *1*(1), 269-271.

[12] Johnson, D. B. (1973). A note on Dijkstra's shortest path algorithm. *Journal of the ACM (JACM)*, *20*(3), 385-388.

[13] Russell, S., and Norvig, P. (2002). Artificial intelligence: a modern approach.

[14] Delling, D., Sanders, P., Schultes, D., and Wagner, D. (2009). Engineering route planning algorithms. In *Algorithmics of large and complex networks* (pp. 117-139). Springer, Berlin, Heidelberg.

[15] Latombe, J. C. (2012). *Robot motion planning* (Vol. 124). Springer Science & Business Media.

[16] Choset, H. M., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., and Arkin, R. C. (2005). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.

[17] Peyer, S., Rautenbach, D., and Vygen, J. (2009). A generalization of Dijkstra's shortest path algorithm with applications to VLSI routing. *Journal of Discrete Algorithms*, *7*(4), 377-390.

[18] Karas, I. R., and Demir, S. (2011). Dijkstra algorithm interactive training software development for network analysis applications in GIS. *Energy Education Science and Technology Part A: Energy Science and Research*, *28*(1), 445-452.

[19] Djokic, D., and Maidment, D. R. (1993). Application of GIS network routines for water flow and transport. *Journal of Water Resources Planning and Management*, *119*(2), 229-245.

[20] Chaomin Luo, Mohan Krishnan, Mark Paulik and Gene Eu Jan, "An Effective Trace-Guided Wave-front Navigation and Map Building Approach for Autonomous Mobile Robots," *Proc. of SPIE Conference on Intelligent Robots and Computer Vision: Algorithms and Techniques*, San Francisco, Feb 2014.

[21] Preparata, F. P., and Shamos, M. I. (2012). *Computational geometry: an introduction*. Springer Science & Business Media.

[22] Lozano-Pérez, T., and Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM*, *22*(10), 560-570.

[23] Rohnert, H. (1986). Shortest paths in the plane with convex polygonal obstacles. *Information Processing Letters*, *23*(2), 71-76.

[24] Ghosh, S. K., and Mount, D. M. (1991). An output-sensitive algorithm for computing visibility graphs. *SIAM Journal on Computing*, *20*(5), 888-910.

[25] Michel Pocchiola and Gert Vegter. Computing the visibility graph via pseudotriangulations. In *Proceedings of the 11th annual Symposium on Computational Geometry (SoCG)*, pages 248–257. ACM Press, 1995.

[26] Zhang, J., Papadias, D., Mouratidis, K., and Manli, Z. (2005). Query processing in spatial databases containing obstacles. *International Journal of Geographical Information Science*, *19*(10), 1091-1111.

[27] Gao, Y., Yang, J., Chen, G., Zheng, B., and Chen, C. (2011, November). On efficient obstructed reverse nearest neighbor query processing. In *Proceedings of the 19th ACM SIGSPATIAL international conference on advances in Geographic Information Systems* (pp. 191-200).

[28] Li, Z., Gao, Y., and Lu, Y. (2011, October). Continuous obstructed range queries in spatio-temporal databases. In 2011 *IEEE International Conference on System science, Engineering design and Manufacturing informatization (Vol. 2, pp. 267-270).*

[29] Van Kreveld, M., Schwarzkopf, O., de Berg, M., and Overmars, M. (2000). *Computational geometry algorithms and applications*. Springer.

[30] LaValle, S. M. (2006). *Planning algorithms*. Cambridge univ. press.

[31] Chazelle, B. (1985). Approximation and decomposition of shapes. *Algorithmic and Geometric Aspects of Robotics*, *1*, 145-185.

[32] Gene Eu Jan, Wei Chun Tsai, Chi-Chia Sun and Bor-Shing Lin, "A Delaunay triangulation-based shortest path algorithm with O(n log n) time in the Euclidean plane," 2012 *IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, Kaohsiung, Taiwan, pp. 186- 189, July 2012.

[33] Jan, G. E., Sun, C. C., Tsai, W. C., and Lin, T. H. (2013). An $O(n\log n)$ Shortest Path Algorithm Based on Delaunay Triangu-lation. *IEEE/ASME Transactions on Mechatronics*, *19*(2), 660-666.

[34] Fortune, S. (1995). Voronoi diagrams and Delaunay triangulations. *Computing in Euclidean geometry*, 225-265.

[35] L. P. Chew, "Constrained Delaunay triangulations," in *Proceedings of the third annual symposium on Computational geometry*, ser. SCG '87, 1987, pp. 215-222.

[36] Finkel, R. A., and Bentley, J. L. (1974). Quad trees a data structure for retrieval on composite keys. *Acta informatica*, *4*(1), 1-9.

[37] Gene Eu Jan, Ki-Yin Chang,Su Gao, "A 4- geometry maze router and its application on multiterminal nets ," *ACM Transactions on Design Automation of Electronic Systems*, 2005.

[38] Gene Eu Jan, and Ki-Yin Chang, "A New Cell Decomposition Approach to Automatic Path-Planning for a Mobile Robot," *The 7th International Conference on Automation Technology*, National Chung Cheng Univ., Chia Yi, Taiwan, Sep. 2003, pp. 894~899.

[39] Gene Eu Jan, and Ki-Yin Chang, 〝A New Maze Routing Approach for Path-Planning of a Mobile Robot,〞 *Proceedings of the IEEE/ASME International Conf. on Advanced Intelligent Mechatronics*, Kobe, Japan, July 2003, pp. 552~557. (EI)