Path Planning by Using Generalized Voronoi Diagrams and Dijkstra Algorithm

by

Lai Hon Lin

Project Proposal

of

Computational Geometry

Assessed by:

Dr. Deng Jun Hui

1. Abstract:

This proposal is a description of my computational geometry project, which is a program of path planning. It is based on the Voronoi Diagram algorithm and the Dijkstra shortest path-searching algorithm, written in Microsoft Visual J++ 6.0, appear as an applet that can be put on the web. So far, I had finished the about half of such program, and seems practically work.

It is a 2D map path-planning program that allows users to draw their own map on specified area. Then, it can decompose the edges of all the obstacles on the map into lots of points. After that, the program can generate the Voronoi Diagrams based on the points. With appropriate eliminations, the generalized Voronoi Diagrams shows the walkable paths on the map. Finally, user can specify a start and a destination point, then the program can find a shortest path by the Dijkstra algorithm. Part of TsingHua University will be drawn in the program for a default map.

2. Introduction:

Nowadays, robotics is a rapidly developing field. And path planning is one of the hottest topics in this field. Basically, robot's path planning can be divided into 2 main directions, one is planning in certain areas and the other is in uncertain region. Actually the path-planning algorithm in certain region is more useful in daily life. For example, autonomous vehicles, information systems in sites and buildings, transportation management systems, etc. Traditionally, this kind of system will have good performance in path planning but have a common problem that is the map is fixed and such system cannot give an immediate reaction on the case that an obstacle is appear on the given path. Also if there is any change in the map, e.g. new buildings, many systems may need a difficult algorithm on calculating the new situation. So I would like to develop a software that can be used for path planning with more simplex algorithm results in faster reactions on exceptional case and easier on map modifications.

The basic idea of traditional path planning systems is the connections between different paths, so the main concept is the walkable paths. A suitable path is given by appropriate algorithm. Different from traditional idea, my project's main concept is obstacles, all the unwalkable region are treated as obstacles and all the calculations are based on the obstacles.

The program is divided into 3 main parts, they are the map generation, the path generation and the shortest path planning.

3. Objectives:

To develop a software for path planning by using generalized Voronoi diagrams for the map and path generation. And use Dijkstra algorithm for shortest path searching. If possible, use the campus of TsingHua University as the default map of such software.

4. Methodology:

2D-Voronoi Diagram has a special property that makes path planning can base on obstacles. There will always a perpendicular bisection line lying between two neighbour points. Suppose all the points are come form the edges of the obstacles, then the bisection lines will only have two possible property. If the two points are come from same obstacle edge, the bisection line will definitely has at least one endpoint lies inside the obstacle. On the other hand, if the two points are come form different obstacles, then the bisection line will lies between these two obstacles, that is the path we can go.

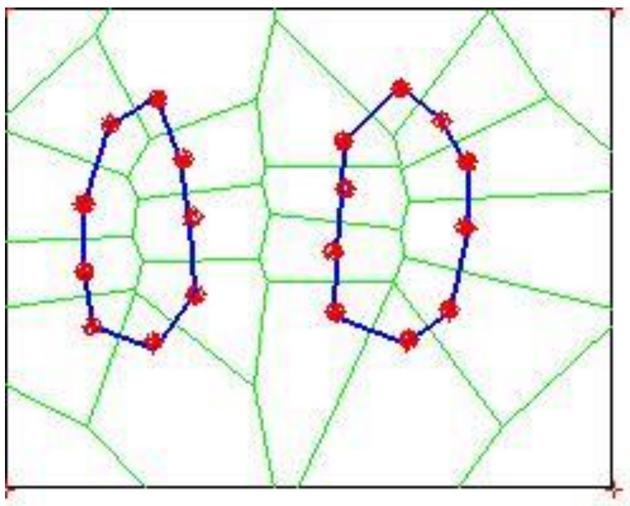

Figure 1: Voronoi Diagrams form from points on two obstacles

Figure 1 shows an example of such property. Assume there are two obstacles (blue) and the red points are coming from the edges of the obstacles. The green lines are the Voronoi edges generated by the red points. It is easily to notice that the Voronoi edges that lie between two neighbour points on same obstacle will have at least one endpoint inside the obstacle. Obviously, points on different obstacles form the outside Voronoi edges, which are the paths.

Map Generation:

In the general problem, one is given a two-dimensional map of the complicated region in which a robot will be operating. Such a map includes a variety of polygonal obstacles that are to be avoided. If possible, the default map that being investigated will be the campus of TsingHua University, or may be part of it. In addition, user can also draw his map on the specified area.

In such map, I will use an approach based on the generalized Voronoi diagram as mentioned before for a planar region with specified obstacles.

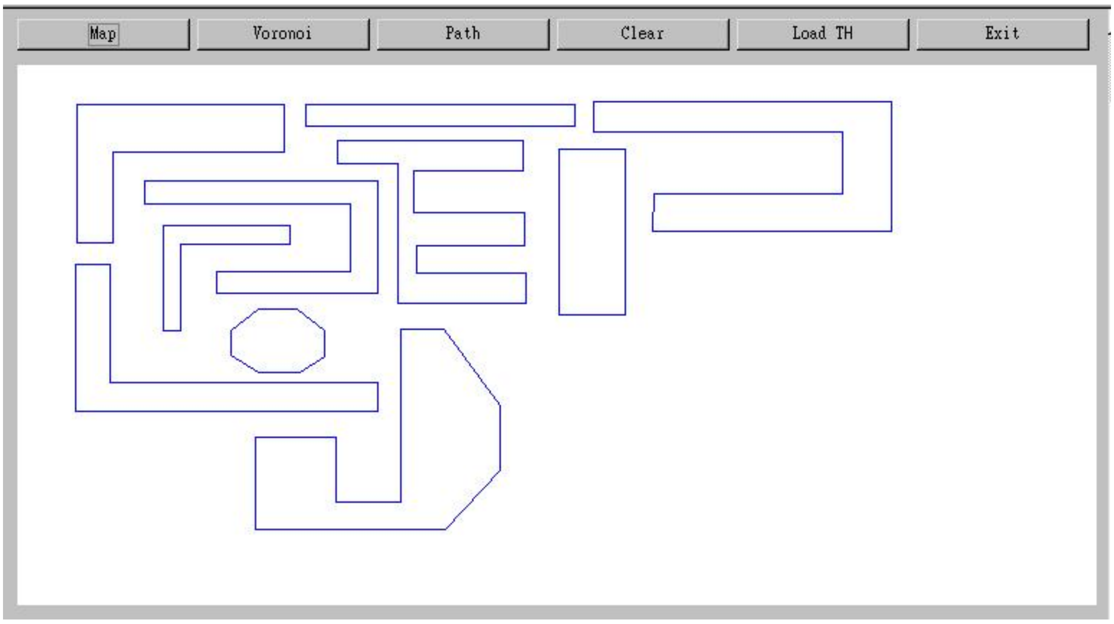

Figure 2: A sample map drawn with the software

The two-dimensional region in which the robot moves will contain buildings and other types of barriers, each of which can be represented by a convex or concave polygonal obstacle. To find the generalized Voronoi diagram for this collection of polygons, I use an approximation that decompose the edges of all the obstacles into lots of discrete points in order to compute the Voronoi diagram for such map.

Generalized Voronoi Diagram:

After the map had been generated and the discrete points had been confirmed, then the Voronoi diagram can be generated by common algorithms. As mentioned before, the Voronoi edges will only have two possible properties since all the points are come from the edges of obstacles. With appropriate eliminations on those Voronoi edges have one or both endpoints lie inside the obstacles. The remain Voronoi edges form a good approximation of the generalized Voronoi diagram for the original obstacles in the map.

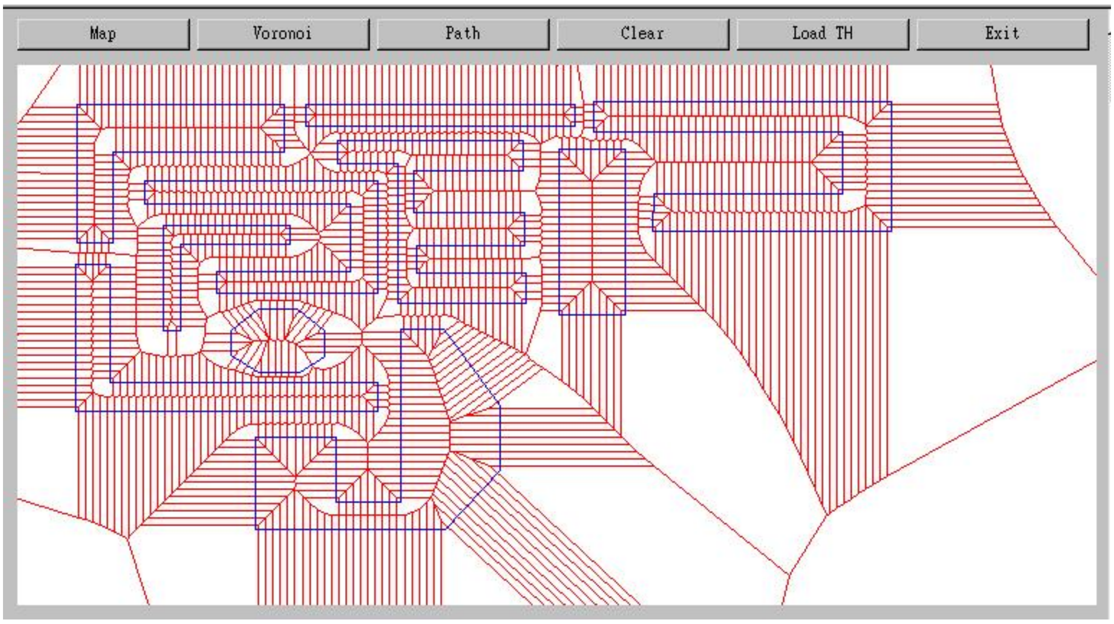

Figure 3: Voronoi Diagrams form from the points of obstacles

Figure 2 shows the Voronoi diagrams of the points that decomposed from the obstacles' edges. The red lines are the Voronoi edges. After eliminations, the generalized Voronoi diagram in black colour is shown in Figure 4, which forms the paths that can be pass through.

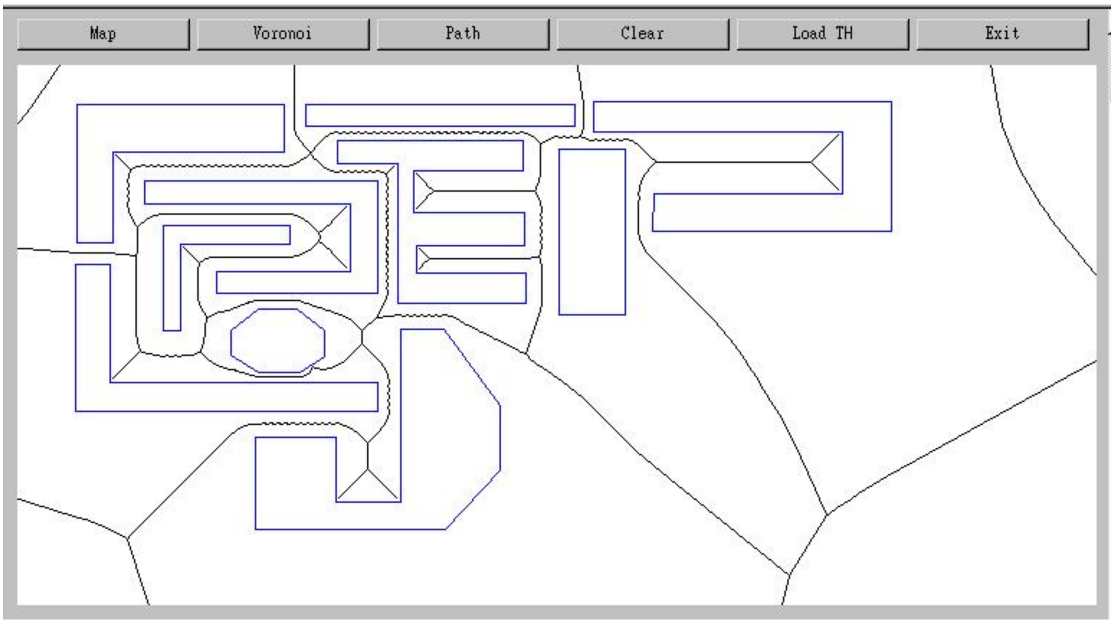

Figure 4: Generalized Voronoi Diagrams formed after eliminations

Path Planning

In the generalized Voronoi diagram, then locate the starting and stopping points and then compute the Voronoi vertices that are closest to these two points. After that use straight lines to connect the starting and stopping points to these closest vertices.

Once the starting and stopping Voronoi Vertices are confirmed, I use the Dijkstra Algorithm to find out a shortest path, which is a subset of the Voronoi edges that connects the starting and stopping vertices. Then this path can be expanded to a path between the original starting and stopping points.

This method generates a route that for the most part remains equidistant between the obstacles closest to the robot and gives the robot a relatively safe path along which to travel.

5. Conclusion:

This project is aimed to develop a path planning system that using generalized Voronoi diagram and Dijkstra algorithm. The software was divided into 3 parts and developed modularly. It allows user to draw their own map and add obstacles flexibly. The traditional concept is concern much on the paths, and this software is different from it that concerns more on the obstacles, thus leads the flexibility on map modifications. Then use the Dijkstra algorithm to find the shortest path once the starting point and stopping point are confirmed.