ROBOT NAVIGATION USING VORONOI DIAGRAMS

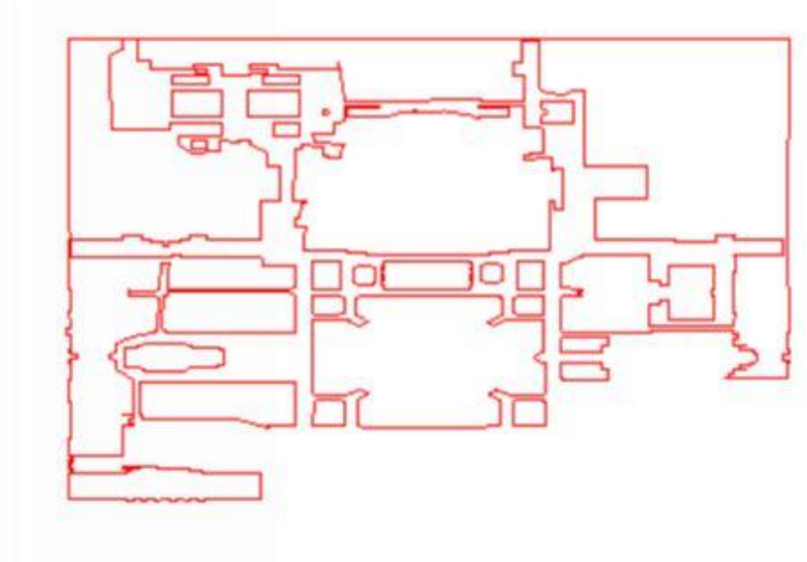
The Team –

1. IMT2013035 - Rishabh Manoj - Rishabh.Manoj@iiitb.org
2. IMT2013005 - Anirudh Ravi - Anirudh.Ravi@iiitb.org
3. IMT2013029 - Penmetsa Murali Krishnam Raju - raju.pmk@iiitb.org
4. IMT2013039 - Samyak Upadhyay - Samyak.Upadhyay@iiitb.org
5. IMT2013051 - Tadigadapa Abhiram - Abhiram.Tadigadapa@iiitb.org
6. IMT2013027 - Nigel Steven Fernandez - NigelSteven.Fernandez@iiitb.org

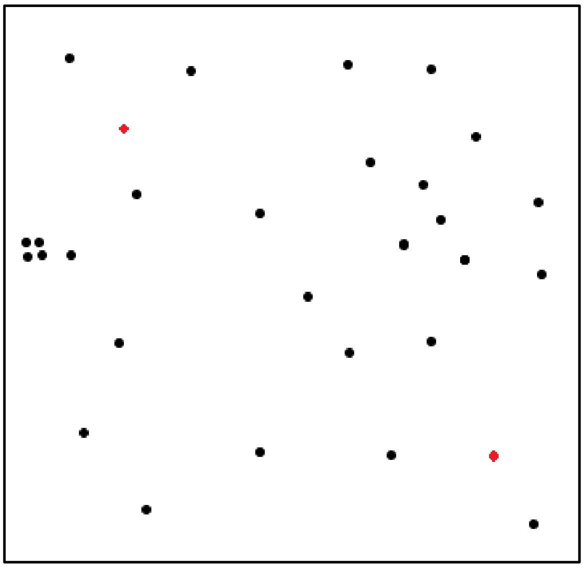
An Overview –

1. <INPUT> The software will require a map specifying the location of the various obstacles, point A (starting point) and point B (stopping point). This can be obtained from the user by specifying the co-ordinates or an image can be scanned.

2. The obstacles can either be treated as a simplistic version of discrete points or can be retained as irregular distorted polygons.

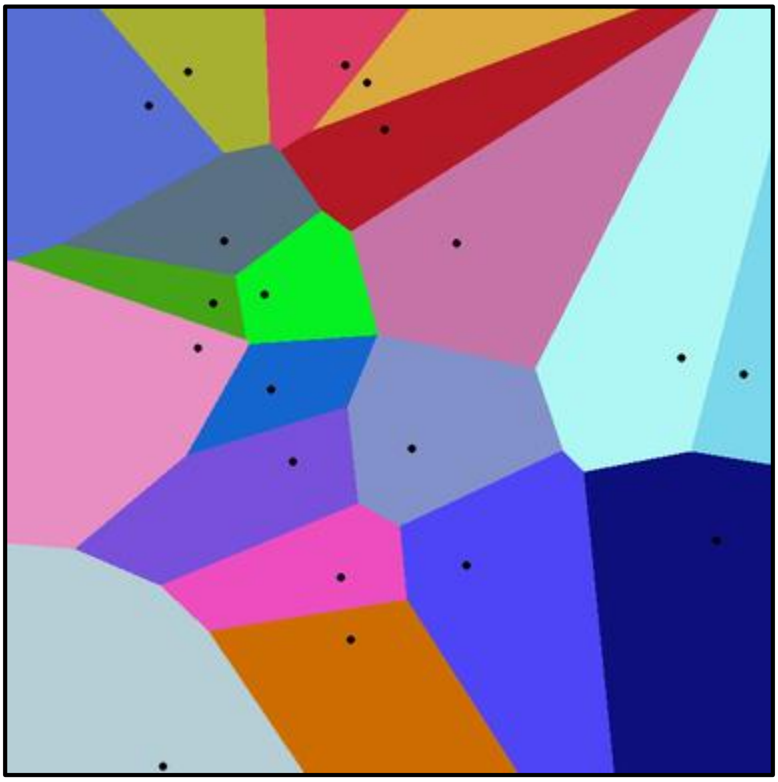


The 2-dimensional map retaining the obstacles shape.



The 2-dimensional map treating obstacles as discrete points. The red points are A and B.

3. A Voronoi diagram is computed for the various points on the map.
   <BACKGROUND KNOWLEDGE> A Voronoi diagram is a way of dividing space into a number of regions. A set of points is specified beforehand and for each point there will be a corresponding region (Voronoi cells) consisting of all points closer to that point than to any other.

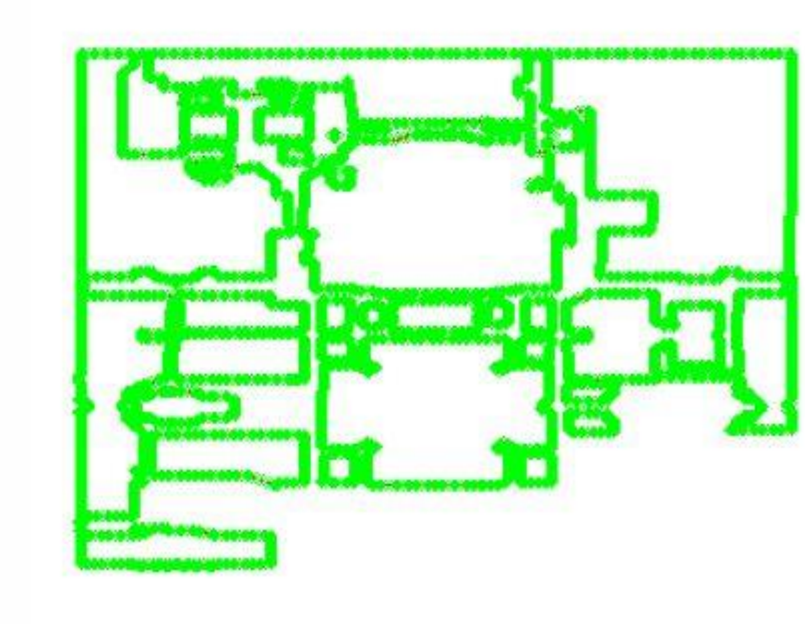


20 points and their Voronoi cells. The nearest neighbour of any point in the plane is the point contained in the region where it lies.

4. The path from point A to point B is then calculated using the edges of the Voronoi diagram. Non – obstacle intersecting, straight lines are used to connect points A and B to appropriate vertices. This results in the safest path as the points on the edges of the graph are at maximum distance (or equidistant) from the corresponding obstacles.

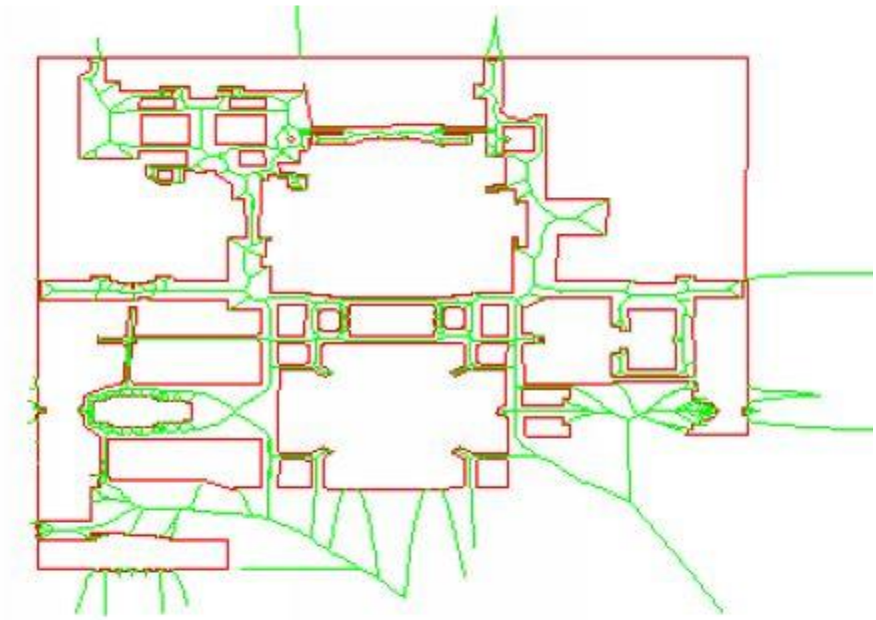
5. The shortest path is then calculated from point A to point B using suitable algorithms like the Dijkstra's Algorithm.

This is illustrated as:

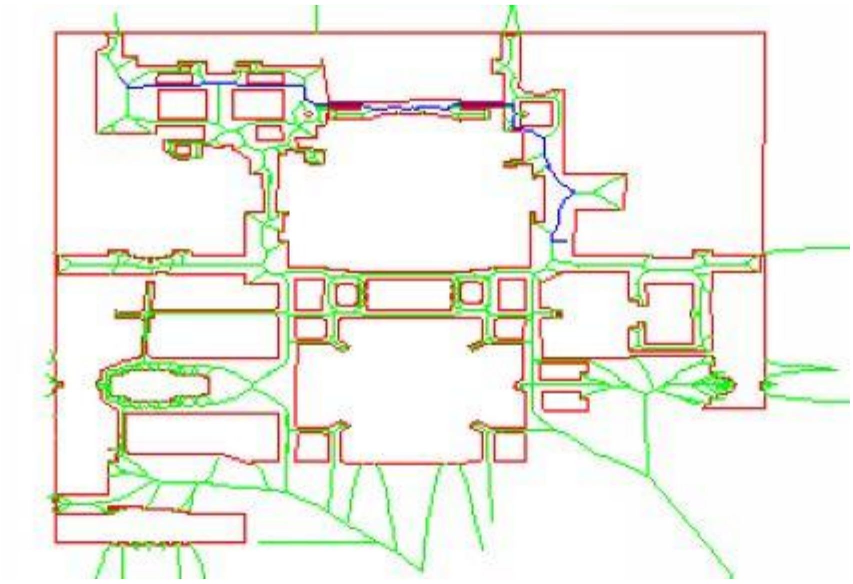


The points that approximate the polygonal obstacles

.



Corresponding simplified Voronoi diagram.



A sample path chosen along the Voronoi diagram by the search algorithm

.

(Note – The obstacles may be treated as discrete points or as above.)

6. The blue line depicted is the shortest, safest path for the robot to travel between point A and point B. This will be displayed on the map by depicting it on the monitor. Further an actual robot may be used to travel between points A and B along the blue path overcoming the obstacles. This practical demonstration will involve the map along with its obstacles, point A and point B being projected on the floor with a real robot navigating along the blue path.

References:

1. For opening up the world of Voronoi diagrams which sparked off the final idea in us – Prof G Srinivasaraghavan.

2. For the diagrams and implementation ideas -http://www.cs.columbia.edu/~pblaer/projects/path_planner/

3. For making Voronoi diagrams fun – http://alexbeutel.com/webgl/voronoi.html

4. For a brief description of Voronoi diagrams - http://en.wikipedia.org/wiki/Voronoi_diagram#Illustration