ENGINEERING MECHANICS 2004
NATIONAL CONFERENCE
with international participation
Svratka, Czech Republic, 10 - 13 May 2004

ROBOT NAVIGATION USING VORONOI DIAGRAMS

M. Šeda*

Summary: *The task of planning trajectories plays an important role in transportation, robotics, etc. In robot motion planning the robot should pass around the obstacles, from a given starting position to a given target position, touching none of them, i.e. the goal is to find a collision-free path from the starting to the target position. This task has many specific formulations depending on the shape of obstacles, allowable directions of movements, knowledge of the scene, etc. Research on path planning has yielded many fundamentally different approaches to its solution, e.g. visibility graph method or the shortest path map method. Assuming movements only in a restricted number of directions (eight directional, horizontal/vertical) the task, with respect to its combinatorial nature, must be solved by heuristic techniques. We propose an application of the Voronoi diagrams to the studied tasks and show that this approach needs only polynomial time and choosing Euclidean or rectilinear metric it can be adapted to tasks with general or directional-constrained movements.*

1. Introduction

The task of planning trajectories of a mobile robot, has received considerable attention in the research literature (de Berg *et al*., 2000), (Sugihara & Smith, 1999), (Zilouchian & Jamshidi, 2001). This task can be formulated in many ways depending on various conditions, e.g. on the fact whether the terrain contains obstacles, which shape they have, or whether the obstacles are movable. Further constraints may represent knowledge of the scene (complete or partial), the metric under consideration and so on. In this paper, we concentrate on a special case of motion planning in the 2D completely known scene with static point and polygonal obstacles that can be composed from rectangular parts and where possible movements of a robot are reduced only to horizontal, vertical and diagonal directions. This problem is usually solved by heuristics applied to a grid representation of the plane e.g. (Sugihara & Smith, 1999) and can include a case-based reasoning procedure (Kruusmaa & Svensson 1998a,b), (Šeda & Dvořák, 2003). Unfortunately the cardinality of the search space of possible paths in the grid has exponential dependence on the granularity of the plane.

Therefore we propose an entirely different approach based on an application of a rectilinear Voronoi diagram using only steps of polynomial time complexity and avoiding all the other drawbacks of the previous approach. In contrast to (Guha & Suzuki, 1997), we will

* Doc. RNDr. Ing. Miloš Šeda, Ph.D., Institute of Automation and Computer Science, Faculty of Mechanical Engineering, Brno University of Technology, Technická 2, 616 69 Brno, tel.: +420 541143332, fax: +420 541142330, e-mail: seda@uai.fme.vutbr.cz

start with the classical Voronoi diagram in the Euclidean plane and later adapt it to the rectilinear case and define a way of replacing its diagonal segments to apply it also to constructing a horizontal/vertical trajectory between the starting and target position.

2. Path planning using Voronoi diagrams

Voronoi diagrams are among the most fundamental data structures in computational geometry. These geometric structures were first discussed by Peter Lejeune Dirichlet (1805-1859) in 1850. However, they were given their name by Georgy Fedoseevich Voronoi (1868-1908), a Russian mathematician, who wrote a fundamental paper about them in 1908. Voronoi diagrams have a surprising variety of uses, e.g. nearest neighbour search, facility location, path planning, etc. In this paper, we investigate their possible use in point-to-point motion planning and propose a method for solving this problem on the Euclidean or rectilinear plane.

A Voronoi diagram of a set of sites in the plane is a collection of regions that divide up the plane. Each region corresponds to one of the sites and all the points in one region are closer to the site representing the region than to the other site.

When Voronoi diagrams are non degenerate (no four or more of its Voronoi edges have a common endpoint) then the following properties are satisfied (Aurenhammer, 1991), (de Berg *et al.*, 2000):

- Every vertex of a Voronoi diagram $V(P)$ is a common intersection of exactly three edges of the diagram.
- The bisector between points p_i and p_j defines an edge of $V(P)$ if and only if there is a point q such that $C_P(q)$ contains both p_i and p_j on its boundary but no other point.
- Voronoi diagram $V(P)$ of P is planar.
- The number of vertices in the Voronoi diagram of a set of n point sites in the plane is at most $2n-5$ and the number of edges is at most $3n-6$.

2.1. Algorithms for constructing Voronoi diagrams

Fundamental algorithms for constructing Voronoi diagrams are the *incremental algorithm*, *random incremental algorithm*, *divide and conquer algorithm* and *plane sweep algorithm* (or *Fortune's algorithm*). More details can be found e.g. in (Aurenhammer, 1991), (de Berg *et al.*, 2000), (Fortune, 1992) and (Shamos & Hoey, 1975). The time complexity of the incremental algorithm is $O(n^2)$ in the worst case, and $O(n \log n)$ for the other three algorithms. We will briefly describe the first two of them.

The incremental algorithm inserts the points one at a time into the diagram. First, we need to find the current Voronoi region into which the new point p falls. Let q be the point defining this region; the separator of p and q then will contribute an edge, e, to p's region. Second, we need to "walk around" the boundary of the new point's Voronoi region. This boundary is created edge by edge, starting with e. Finally, we delete all the old edges sticking into the new region.

The second and third steps are more time consuming. It is possible that each new point's Voronoi region will touch all the old regions. Thus, in the worst case, we end up spending linear time on each region, or $O(n^2)$ time overall.

One approach to speeding up the insertion is its *randomisation*. It has been proven that inserting the points in random order yields an $O(n \log n)$-time performance with high probability, regardless of which set of points is given.

2.2. Motion planning in the scene with point obstacles

Consider a disc-shaped robot in the plane. It should pass among a set P of point obstacles, getting from a given starting position to a given target position and touching none of the obstacles. If such a passage is possible at all, the robot always walks along the edges of the Voronoi diagram of P, which define the possible channels that maximise the distance to the obstacles, except for the initial and final segments of the tour. This allows us to reduce the robot motion problem to a graph search problem: we define a subgraph of the Voronoi diagram consisting of the edges that are passable for the robot. However, in Fig. 1 some of the edges of the Voronoi diagram are not passable and therefore these edges must be omitted from the diagram. If the graph after this reduction is still connected, we can use the previous approach again. In the case when initial and final segments are connected to different components of the graph, then the path among the obstacles does not exist.

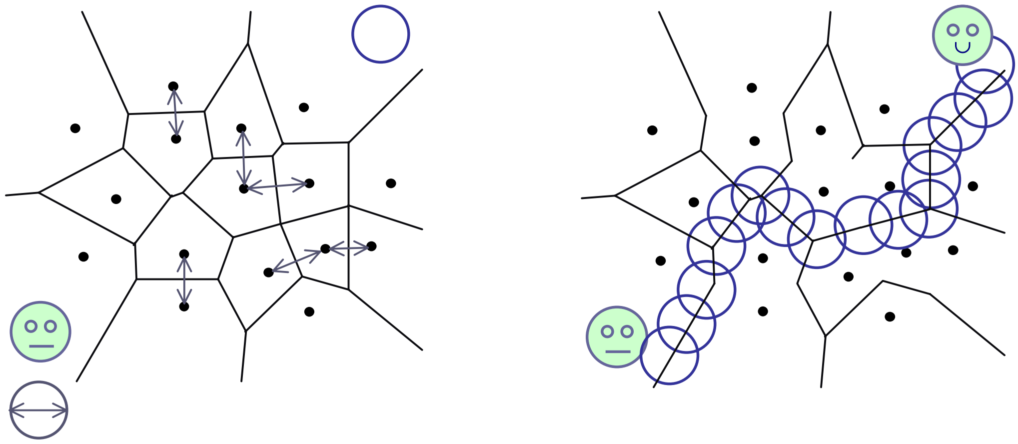

Fig. 1. Motion planning using Voronoi diagrams with impassable edges

The lengths of paths along the edges of the diagram are given by the sums of the edge lengths computed as Euclidean distances of the corresponding pairs of vertices. The shortest path can be easily solved by the Dijkstra algorithm. Using a binary heap implementation, its time complexity is given by $O(|E| \log |V|)$, where E is a set of edges and V is a set of vertices. Note that the number of vertices in the Voronoi diagram of a set of n point sites in the plane is at most $2n-5$ and the number of edges is at most $3n-6$ (de Berg *et al.*, 2000), i.e. both are bounded by $O(n)$.

2.3. Motion planning in eight directions

Let us assume that we wish to plan motion planning constrained by movements in eight directions. If we use the rectilinear metric for the Voronoi diagram then due to the rectilinearity, each straight-line segment of a bisector in rectilinear Voronoi diagrams will be either horizontal, vertical, or inclined at 45° or 135° to the positive direction of the x-axis.

This finding straightforwardly offers to use the rectilinear Voronoi diagram for the 8-directional motion planning. The rectilinear Voronoi diagram can be constructed by a simple modification of the random incremental algorithm for the Euclidean metric.

2.4. Motion planning in the plane with point and rectangular obstacles

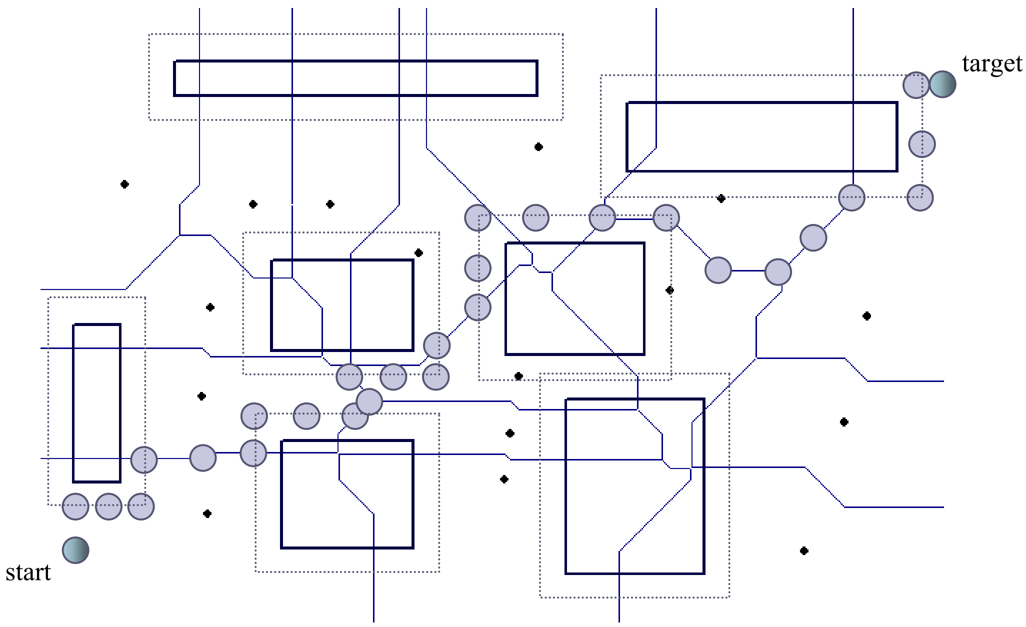

Fig. 2. Scene with point and rectangular obstacles

Now consider a more general case of a scene where, besides point obstacles, rectangular obstacles occur. Fig. 2 shows such a scene with the rectilinear Voronoi diagram constructed for point obstacles only. After the construction of the rectilinear Voronoi diagram for point obstacles, we increase the width and height of the rectangular obstacles by the diameter of the disc-shaped robot and a certain small number representing a reserve for finding a collision-free path. For the optimal path between the starting and target position, we search in the graph whose edges create the edges of the "extended" rectangular obstacles and edges of the rectilinear Voronoi diagram without their parts inside the extended obstacles. This is demonstrated in Fig. 3. The example shown in Fig. 2 and Fig. 3 demonstrates the case of the 8-directional movements. It is obvious that this approach can also be applied for general motion planning using Euclidean version of the Voronoi diagram.

3. Conclusions and future work

In this paper, we proposed applications of the Voronoi diagrams to general and 8-directional motion planning. As algorithms for constructing the Voronoi diagrams run in polynomial time, the number of their edges is linearly dependent on the number of obstacles, algorithms for searching the shortest paths in graphs are also polynomial, and this holds for all additional operations for finding a collision-free path of a robot (replacements, extensions of the rectangular obstacles), the overall time complexity of all proposed algorithms is polynomial. This approach avoids all the drawbacks of classical methods (combinatorial explosion, low boundaries for grid representation and generating many infeasible solutions).

In future, we will try to generalize this approach for cases of more complex shapes of obstacles and movable obstacles. Further investigating will also include the case when the environment is totally or partially unknown, varying over time or a combination of both.

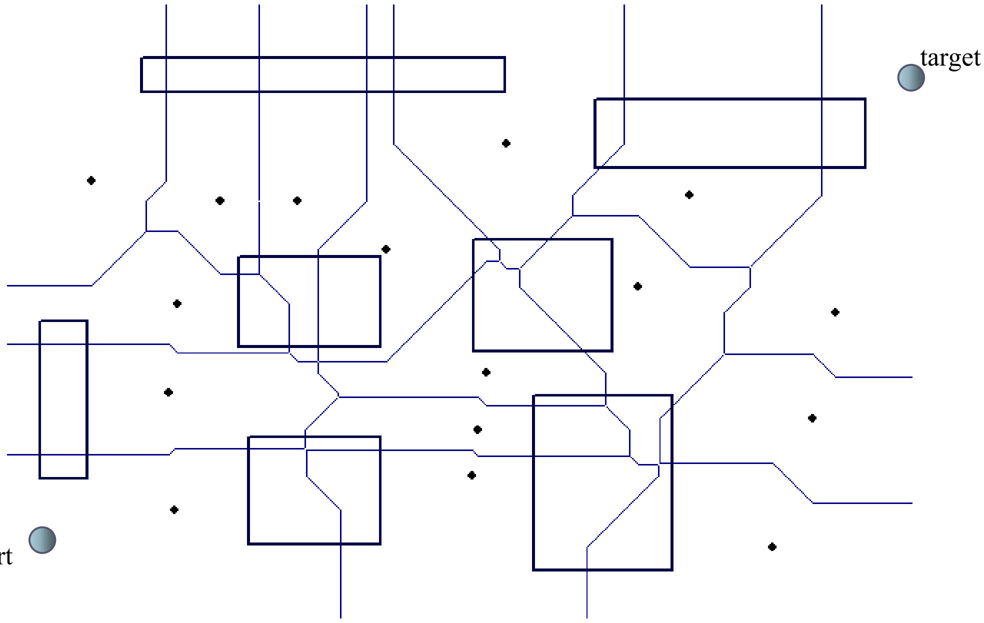

Fig. 3. Motion planning in 8 directions in a scene with point and rectangular obstacles

4. Acknowledgement

This paper was supported by the research projects MSM 261100009 "Non-Traditional Study Methods of Complex and Uncertain Systems", MSM 261100013 "Automation of technologies and production processes" and MSM 262100024 "Research and Development of Mechatronic Systems".

5. References

Aurenhammer, F. (1991) Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure. *ACM Computing Surveys*, Vol. 23, No. 3, pp. 345-405.

de Berg, M., van Kreveld, M., Overmars, M. & Schwarzkopf, O. (2000) *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Berlin.

Fortune, S. (1992) Voronoi Diagrams and Delaunay Triangulations. In Du, D.A. and Hwang, F.K. (eds.): *Euclidean Geometry and Computers*. World Scientific Publishing, pp. 193-233, Singapore.

Guha, S. & Suzuki, I. (1997) Proximity Problems for Points on a Rectilinear Plane with Rectangular Obstacles. *Algorithmica*, Vol. 17, pp. 281-307.

Kruusmaa, M. & Svensson B. (1998a) Combined Map-Based and Case-Based Path Planning for Mobile Robot Navigation, in: *Proceedings of International Symposium of Intelligent Robotic Systems*, Jan 10-12, 6 pp.

Kruusmaa, M. & Svensson B. (1998b) Using Case-Based Reasoning for Mobile Robot Path Planning, in: *Proceedings of the 6th German Workshop on Case-Based Reasoning*, March 6-8, 8 pp. Berlin, Germany.

Šeda, M. & Dvořák, J. (2003) Robot Navigation Using Genetic Algorithm and Case-Based Reasoning, in: *Book of Extended Abstracts of the National Conference with International Participation Engineering Mechanics 2003.* Association for Engineering Mechanics, pp. 328-329 + 8 pp. on CD-ROM. Svratka, Czech Republic.

Shamos, M.I. & Hoey, D. (1975) Closest Point Problems, in: *Proceedings of the 16th Annual Symposium on Foundations of Computer Science FOCS '75*, pp. 151-162. Berkeley, California.

Sugihara, K. & Smith, J. (1999) Genetic Algorithms for Adaptive Planning of Path and Trajectory of a Mobile Robot in 2D Terrains. *IEICE Transactions on Information and Systems*, Vol. E82-D, No. 1, pp. 309-317.

Zilouchian, A. & Jamshidi, M. (2001) *Intelligent Control Systems Using Soft Computing Methodologies*. CRC Press, Boca Raton, Florida.