



Using the delaunay triangulation and voronoi diagrams for navigation in observable environments

Uso de la triangulación de delaunay y los diagramas de voronoi para navegación en ambientes observables

Fernando Martínez Santa*, Fredy Hernán Martínez Sarmiento**, Edwar Jacinto Gómez***

Citation / Para citar este artículo: Martínez Santa, F., Jacinto Gómez, E., & Martínez, F. (2014). Using the delaunay triangulation and Voronoi diagrams for navigation in observable environments. Revista Tecnura, Edición especial, 81-87.

Fecha de recepción: 31 de marzo de 2013, **Fecha de aceptación:** 16 de mayo de 2014

Abstract

The navigation system of mobile robot should have the capability of environment cognition, action decision, motion control, and state monitoring. This paper proposes a prediction algorithm route based on Voronoi diagrams, applicable for mobile robots with range sensors operating in indoor and unknown environments. The proposed control law is a hybrid navigation strategy composed of Voronoi diagrams and Delaunay triangulation, which is used to avoid the routes that may eventually block the advance of the robot. An efficient implementation of the control law is presented. Simulation results validate the efficacy of the proposed modified control law.

Key words: Autonomous robots, obstacle avoidance, robot navigation, Voronoi diagrams.

Resumen

El sistema de navegación de un robot móvil debe tener la capacidad de reconocer su medio, la acción de decisión, el control de movimiento, y la supervisión de sus variables de estado. En este artículo se propone un algoritmo de predicción de ruta basado en los diagramas de Voronoi, aplicable a robots móviles con sensores de rango que operan en ambientes interiores y desconocidos. La ley de control propuesta es una estrategia de navegación híbrida compuesta por los diagramas de Voronoi y la triangulación de Delaunay, que se utiliza para evitar las rutas que pueden llegar a bloquear el avance del robot. Se presenta una aplicación eficaz de la ley de control. Los resultados por simulación validan la eficacia de la ley de control propuesta.

Palabras clave: Diagramas de Voronoi, evasión de obstáculos, navegación robótica, robots autónomos.

* Ingeniero en Control Electrónico e Instrumentación, magíster en Ingeniería Electrónica y de Computador. Docente, Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. Contacto: fmartinezs@udistrital.edu.co

** Ingeniero Electricista, especialista en Gestión de Proyectos de Ingeniería, candidato a doctor en Ingeniería. Docente, Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. Contacto: fhmartinezs@udistrital.edu.co

*** Ingeniero en Control Electrónico e Instrumentación, magíster (C) en Ciencias de la Información y Comunicaciones. Docente, Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. Contacto: ejacintog@udistrital.edu.co

INTRODUCTION

The route planning strategies for autonomous mobile robots have been fairly treated in many references (LaValle, 2006; Zhang *et al.*, 2007; Afzulpurkar & Thanh, 2008; Mannadiar, 2010; Bobadilla *et al.*, 2012). We seek to avoid obstacles using global strategies: the first and most used ones are artificial potential fields (Guanghui *et al.*, 2012; Bin-qiang *et al.*, 2011) or algorithms specifically purely geometric triangulation (Kallmann, 2005; Xu *et al.*, 2009; Demyen & Buro, 2006) and many other cell-based geometry such as Voronoi diagrams (Dong *et al.*, 2010; Shao & Lee, 2010), as well as the Delaunay method (Hongyang *et al.*, 2008) that is also interesting. Each one of the above methods have their own difficulties and require specific criteria, such as the method based on artificial potential fields that suggests the challenge of local minima, while triangulation algorithms have problems in the decision and prediction less than the possible routes, and in some cases the algorithms to choose the best route are necessary as Dijkstra's (Dong *et al.*, 2010). In this research, these techniques and some taken from image processing are applied, some of these navigation strategies have been studied by the research group to check its performance and behavior; in the near future comparisons will be made with each of them, in this specific job we choose a combination of Voronoi diagrams with the use of the Delaunay triangulation, implementation and simulation.

Voronoi diagrams are part of the most significant route planning structures in computational geometry. These diagrams basically start from the comparison between elements near in a plan (Okabe *et al.*, 2000). In this method is essential to consider the route as far as possible of the obstacles. So in a 2-dimensional space, all points that are equidistant from two barriers are considered part of the Voronoi diagram (figure 1).

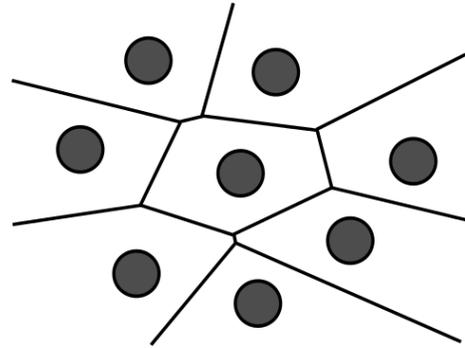


Figure 1. Voronoi diagram, cell division stage (Dong *et al.*, 2010).

The Voronoi diagram is denoted as a set of points $P = \{p_1, p_2, p_3, \dots, p_n\}$ in the plan, or in any m -dimensional space. In this set a particular classification is done, which consists in verify the Euclidean distance of each point regarding to the other points, calculating the distance $d(p, q)$, distance defined as shown in equation (1):

$$d(p, q) = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2} \quad (1)$$

Furthermore, it is defined $V(p_i)$ as the Voronoi cell for p_i , which becomes the set of q points in the plan which are closer to p_i than any other element. Therefore, the Voronoi cell of p_i is defined as shown in equation (2):

$$V(p_i) = \{q \mid \|p_i q\| \leq \|p_j q\|, \forall j \neq i\} \quad (2)$$

This defines the Euclidean distance between p and q . Finally, it is defined the Voronoi diagram of P , called $Vor(P)$, as the plan which has been divided into n Voronoi cells $V(p_i)$, one for each point in P (Okabe *et al.*, 2000).

This paper is organized as follows: Section 2 presents the basic concepts of Voronoi diagrams as base navigation algorithm. In Section 3, the steps to implement the algorithm are presented, and in Section 4, the results and simulations with different environments are shown. Section 5, the paper is concluded.

PROBLEM FORMULATION

The problem is to discard the use of onboard sensors (on the robot). The use of images to the environment taken with a camera is considered, instead. From these images, the information of the robots and obstacles are extracted; then, this information to implement the routing prediction algorithm is used. Therefore, the central control unit is responsible to calculate the navigation path and to control the movement of the robot, that is always possible to sense and known the environment. This means that the position of the robot and the obstacles in the environment are always known.

Let $W \subset \mathbf{R}^2$ be the closure of a contractible open set in the plan that has a connected open interior with obstacles that represent inaccessible regions. Let \mathcal{O} be a set of obstacles, in which each $O \subset \mathcal{O}$ is closed with a connected piecewise-analytic boundary that is finite in length. Furthermore, the obstacles in \mathcal{O} are pairwise-disjoint and countably finite in number. Let $E \subset W$ be the free space in the environment, which is the open subset of W with the obstacles removed, and the boundary, ∂E , of E is the image of a piecewise-analytic closed curve.

The goal is to plan optimal navigation routes through the free space E using a global algorithm. We also use an internal algorithm to determine the best path; i.e., an algorithm that eliminates local minima, paths that block the movement of the robot, and very long paths.

The path planning should be done by image processing (figure 2). The proposed navigation algorithm (which is supported on Voronoi diagrams) must calculate the optimal route from the navigation information extracted from the images, and with the starting and arrival (or target) points.

An improvement to the design of paths with Voronoi diagrams are proposed, particularly addressing the question of the possible closed paths using Delaunay triangulation as a solution, in addition a recursive algorithm that verifies that the selected path is the shortest of the paths defined by Voronoi is implemented.

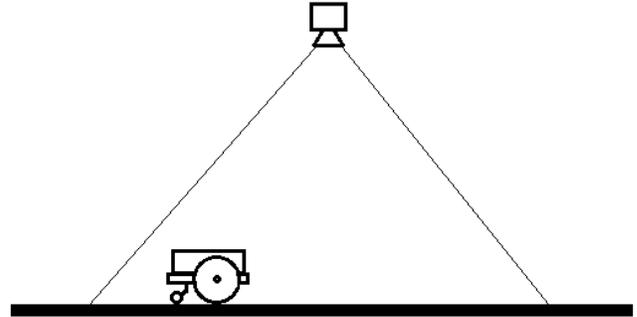


Figure 2. Camera as environment sensor.

NAVIGATION ALGORITHM

Voronoi diagrams as paths prediction method have a lower risk of collision for robot navigation within an environment, since it has a projection of free space E on a one-dimensional curves network. Formally, it is defined as a retraction (Ortega, 2010) with preservation of continuity. If the set C_l defines the unobstructed positions of an environment, the retraction function R_T builds a continuous subset C_v of C_l (equation (3)) (Ollero, 1995).

$$RT(q): C_l \rightarrow C_v / C_v \subset C_l \quad (3)$$

Thus, it is said that there is a path from a starting point q_a to another end point q_f , which is free of obstacles, if and only if there is a continuous curve from $RT(q_a)$ to $RT(q_f)$.

Obstacles represented as points

Considering the obstacles of a map as points (figure 3), the following properties can be defined (Ortega, 2010):

- Two points p_i and p_j are neighbors if they share an edge. An edge is the perpendicular bisector of the segment $p_i p_j$.
- A vertex is a point equidistant from three generators (if it is more than three, then degenerate cases are had) and it is the intersection of three edges.

- A Voronoi region is a convex polygon or an unbounded region.
- A Voronoi region is unbounded if its generating point belongs to the convex hull of the point cloud.

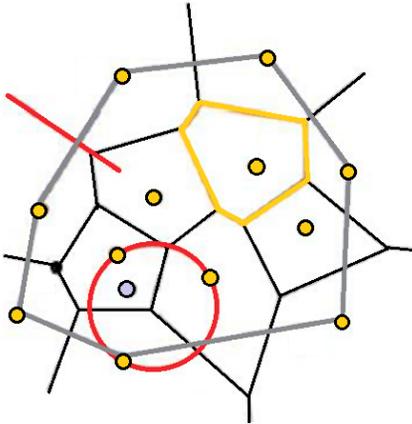


Figure 3. Voronoi diagram and properties (Ortega, 2010).

3.2 Obstacles represented as polygons

The retraction function RT definition involves construction of the Voronoi diagram (Rombaut *et al.*, 1991). In order to make more realistic the obstacles representation, and according to the formulation of the problem, the obstacles are interpreted not as points but as polygons. Thus, the Voronoi diagram is formed by two kinds of segments: rectilinear and parabolic.

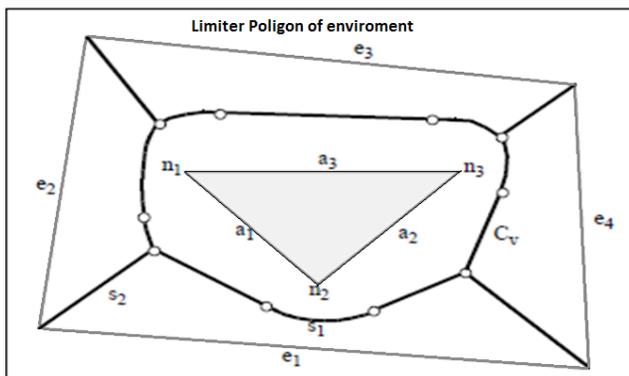


Figure 4. Retraction of free space E on Voronoi diagram (Ollero, 1995).

Figure 4 shows the Voronoi diagram C_v , and is represented by the demarcated lines. The environment W is delimited by polygon edges $\{e_1, e_2, e_3, e_4\}$, the triangular obstacle vertices $\{n_1, n_2, n_3\}$ and the triangular obstacle edges $\{a_1, a_2, a_3\}$. In the figure two kinds of lines that compose the Voronoi diagram can be seen. The segment s_1 is the locus of equidistant points between the edge e_1 , and the vertex n_2 . Similarly, the rectilinear segment s_2 is defined in the same way, but with regard to the edges e_1 and e_2 . Given a configuration that does not belong to C_v , there is a unique nearest point p belonging to a vertex or edge of an obstacle O . The function $RT(q)$ is defined as the first cut with C_v of the line linking p with q (figure 5) (Ollero, 1995).

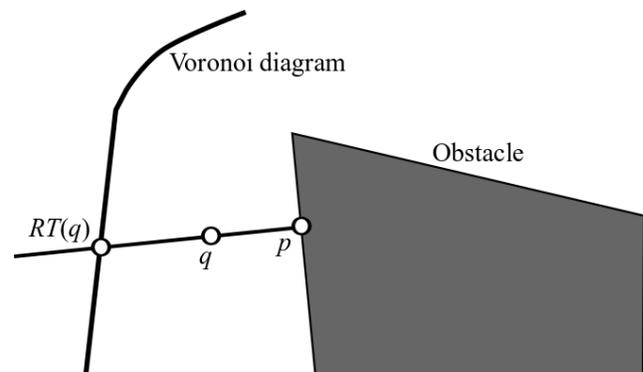


Figure 5. A configuration q in the Voronoi diagram (Ollero, 1995).

For path planning, the Voronoi algorithm tries to find the sequence of line segments for connecting $RT(q_a)$ with $RT(q_p)$. The algorithm (Ollero, 1995):

- Calculate the Voronoi diagram.
- Calculate $RT(q_a)$ and $RT(q_p)$.
- Find the sequence of segments $\{s_1, \dots, s_p\}$ such that $RT(q)$ belongs to s_1 and $RT(q_p)$ belongs to s_p .
- If it finds such a sequence, then it delivers the path. Otherwise it indicates error condition.

METHODOLOGY AND IMPLEMENTATION OF THE ALGORITHM

Application was performed in MatLab, using some techniques of basic image processing (Cuevas *et al.*, 2010). As a first step, the image is required from a file or from a camera located on the environment. The image is pre-processed and converted to a binary matrix. When having this initial file, the obstacles within the environment are labeled and listed, the position of these obstacles are stored as well as its dimensions. From this moment the central unit has all the information about E and the boundary ∂E of E .

The next step is the construction of the Voronoi diagram. The environment the function that calculates the Voronoi cells is applied, this function gives as a result a series of navigation solutions. With the Voronoi diagram, and with the information about the obstacles, the Voronoi diagram the starting and target points are added. Using Euclidean distance, these points are determined on where they are connected.

Finally, the optimal paths for navigation must be determined. The Delaunay triangulation to verify the robot to navigate at a distance r from obstacles is applied. This distance r is calculated as the radius of a circle that circumscribes the robot. For the selection of the final path, a recursive method that compares the different routes using the Euclidean distance is applied (figure 6).

RESULTS

The algorithm was tested on a total of 20 different lab environments, making a total of 40 trials. In each of these tests, the start and target points were randomly selected. In the algorithm, the robot was taken as a circle of radius $r = 10.5$ cm according to the geometry of the laboratory robots. The environment used was a rectangle of 7 m high and 5.5 m wide.

Some of the most significant tests are shown in figure 7. In this figure the Voronoi diagram with the path chosen by the algorithm for four different cases is shown.

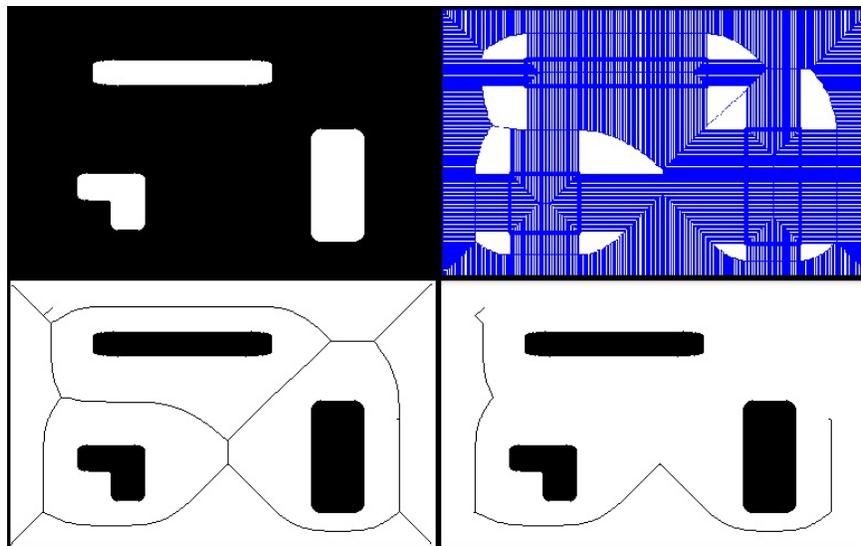


Figure 6. Steps in the proposed algorithm.

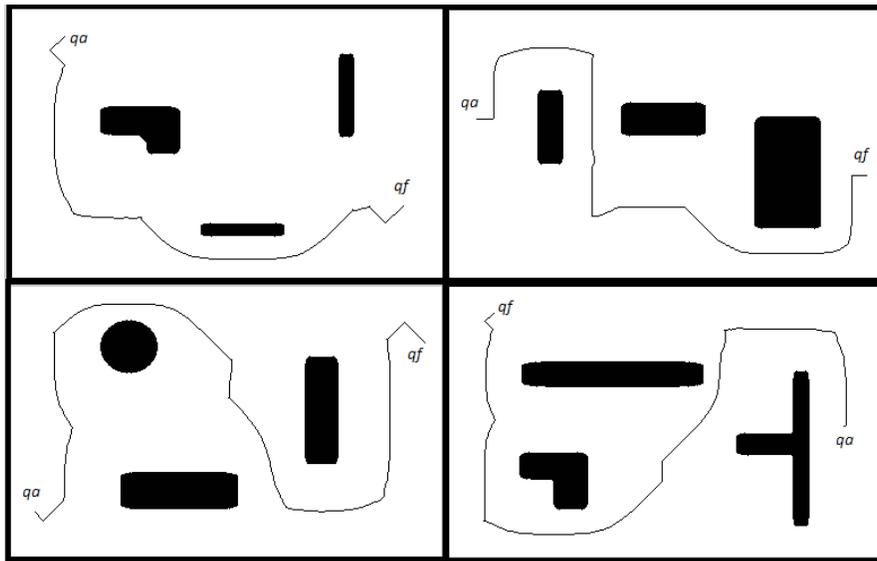


Figure 7. These are four different environments with the navigational paths calculated by the proposed algorithm.

CONCLUSIONS

This paper proposes a hybrid scheme for robot navigation combining two geometric navigation strategies. The modification and verification of Voronoi diagrams with the Delaunay triangulation algorithm avoids possible routes impending collisions, and leaves the road ready for implementing an algorithm using a single Euclidean metric. This gives as a general result that the algorithm chooses, in most cases, the obstacle peripheral roads within the scene, eventually shipping lanes are full of obstacles away with smooth curves and a reasonable distance.

This paper presents an implementation of predicting cell routes, which makes the computational cost be low and these results can be easily measured. The measured times with different scenarios are between 1 and 3 s, using an Intel Core I7 with 8 Gigabytes Ram running Windows 7 64 bits and Mat Lab R2011a. This is a time extremely low, but plotted against delivery routes are not the most optimal, often routes are entirely peripheral.

Such algorithms are based on a central unit which makes its implementation easy because it simply performed without using external sensors, only a low-resolution camera and a personal computer with basic characteristics, in this case only scenarios were applied to static, but as future work it could be implemented for dynamic applications.

The performance of the proposed algorithm was verified through several simulations.

ACKNOWLEDGEMENT

This work was supported by the Universidad Distrital Francisco José de Caldas, in part through CIDC, and partly by the Faculty of Technology. The views expressed in this paper are not necessarily endorsed by Universidad Distrital. The authors thank the research groups DIGITI and ARMOS for the evaluation carried out on prototypes of ideas and strategies.

FINANCIAMIENTO:

Universidad Distrital Francisco José de Caldas

REFERENCES

- Afzulpurkar, N. & Thanh, N. T. (2008). Path planning for a mobile robot in a dynamic environment. In IEEE International Conference on Robotics and Biomimetics ROBIO 2008, 2115-2120.
- Bin-qiang, Y.; Ming-fu, Z. & Yi, W. (2011). Research of path planning method for mobile robot based on artificial potential field. In 2011 International Conference on Multimedia Technology ICMT, 3192-3195.
- Bobadilla, L.; Martínez, F.; Gobst, E.; Gossman, K. & LaValle, S. M. (2012). Controlling wild mobile robots using virtual gates and discrete transitions. In American Control Conference ACC2012, 743-749.
- Cuevas, E.; Zaldívar, D. & Pérez, M. (2010). Procesamiento digital de imágenes con Matlab y Simulink. México: Alfaomega Grupo Editor.
- Demyen, D. & Buro, M. (2006). Efficient triangulation-based pathfinding. In Proceedings of the 21st National Conference on Artificial Intelligence, AAAI American Association for Artificial Intelligence, 942-947.
- Dong, H.; Li, W.; Zhu, J. & Duan, S. (2010). The path planning for mobile robot based on Voronoi diagram. In 2010 3rd International Conference on Intelligent Networks and Intelligent Systems ICINIS, 446-449.
- Guanghui, L.; Yamashita, A.; Asama, H. & Tamura, Y. (2012). An efficient improved artificial potential field based regression search method for robot path planning. In 2012 International Conference on Mechatronics and Automation ICMA, 1227-1232.
- Hongyang, Y.; Huifang, W.; Yangzhou, Ch. & Guiping, D. (2008). Path planning based on constrained Delaunay triangulation. In 7Th World Congress on intelligent control and automation 2008, 5168-5173.
- Kallmann, M. (2005). Path planning in triangulations. In Proceedings of the Workshop on Reasoning, Representation, and Learning in Computer Games, International Joint Conference on Artificial Intelligence IJCAI, 49-54.
- LaValle, S. M. (2006). Planning algorithms. Cambridge, U.K.: Cambridge University Press.
- Mannadiar, R. (2010). Optimal coverage of a known arbitrary environment. In 2010 IEEE International Conference on Robotics and Automation ICRA, 5525-5530.
- Okabe, A.; Boots, B.; Sugihara, K. & Chiu, S. N. (2000). Spatial Tessellations: Concepts and Applications of Voronoi Diagrams. England: John Wiley & Sons, Ltd.
- Ollero, A. (1995). Planificación de trayectorias para Robots Móviles. España: Universidad de Málaga.
- Ortega, L. (2010). El Diagrama de Voronoi. España: Universidad de Jaen.
- Rombaut, M.; Segovia, A.; Meziel, D. & Preciado, A. (1991). Displacements of a mobile robot in a known environment. In IMACS-IFAC Symposium MCTS.
- Shao, M. & Lee, J. Y. (2010). Development of autonomous navigation method for nonholonomic mobile robots based on the generalized Voronoi diagram. In 2010 International Conference on Control Automation and Systems ICCAS, 309-313.
- Zhang, L.; Kim, Y. J. & Manocha, D. (2007). A hybrid approach for complete motion planning. In IEEE/RSJ International Conference on Intelligent Robots and Systems IROS 2007, 7-14.
- Xu, D.; Zhang, F. & Yao, Y. (2009). Constructing visibility graph and planning optimal path for inspection of 2D workspace. In *IEEE International Conference on Intelligent Computing and Intelligent Systems ICIS 2009*, 693-698.



