Path Planning for Mobile Robot Navigation using Voronoi Diagram and Fast Marching

Santiago Garrido
Robotics Lab.
Carlos III University
Madrid, Spain
Email: sgarrido@ing.uc3m.es

Luis Moreno
Robotics Lab.
Carlos III University
Madrid, Spain
Email: moreno@ing.uc3m.es

Mohamed Abderrahim[1]
Robotics Lab.
Carlos III University
Madrid, Spain
Email: mohamed@ing.uc3m.es

Fernando Martin
Robotics Lab.
Carlos III University
Madrid, Spain
Email: fmartin@ing.uc3m.es

***Abstract*— This paper presents a new sensor based global Path Planner which operates in two steps. In the first step the safest areas in the environment are extracted by means of a Voronoi diagram. In the second step Fast Marching Method is applied to the Voronoi extracted areas in order to obtain the shortest path. In this way the trajectory obtained is the shortest between the safe possible ones. This two step method combines an extremely fast global planner operating on a simple sensor based environment modeling, while it operates at the sensor frequency. The main characteristics are speed and reliability, because the map dimensions are reduced to a unidimensional map and this map represents the safest areas in the environment for moving the robot.**

I. INTRODUCTION

Robot motion planing problems have been an important research topic since the 80's. A considerable number of researchers have been working extensively to develop efficient methods to overcome these problems. This issue has been dealt with in two general ways: one approach was concentrated in solving motion planing problems using a previously known global environment or obstacle information and the robot characteristics, while the second approach has concentrated in planing motion using local sensor information and the robot characteristics.

When we want to move a robot from one location to another it is necessary to use a global map to calculate a global trajectory. Mobile robot path planning approaches can be divided into five classes [1]. Roadmap methods extract a network representation of the environment and then apply graph search algorithms to find a path. Exact cell decomposition methods construct non-overlapping regions that cover free space and encode cell connectivity in a graph. Approximate cell decomposition is similar, but cells are of predefined shape (e.g. rectangles) and do not exactly cover free space. Potential field methods differ from the other four in that they consider the robot as a point evolving under the influence of forces that attract it to the goal while pushing it from obstacles. Navigation functions are commonly considered a special case of potential fields.

In order to calculate the trajectory in the global map, this paper presents a new Path Planning method based in the combination of Voronoi Diagram and the Fast Marching Method.

The Fast Marching Method has been applied to Path Planning [Sethian:1], and their trajectories are of minimal distance, but they are not very safe because the path is too close to obstacles resulting in jerky trajectories.

In order to improve the safety of the trajectories calculated by the Fast Marching Method, it is possible to give two solutions:

The first possibility, in order to avoid unrealistic trajectories, produced when the areas are narrower than the robot, the segments with distances to the obstacles and walls less than half the size of the robot need to be removed from the Voronoi Diagram.

The second possibility, used in this work, is to dilate the objects and walls in a security distance that ensures that the robot neither collides with obstacles and walls nor accepts passages narrower than the robot size.

The last step is to calculate the trajectory in the image generated by the Voronoi Diagram using the Fast Marching Method. Then, the obtained path verifies the smooth and safety considerations required for mobile robot path planning.

The advantages of this method are the easy implementation, the speed of the method and the quality of the trajectories. The method works in 2D and 3D, and it can be used at a global or local scale, in this case operating with sensor information instead of using an a priori map (sensor based planning).

II. Introduction to the Voronoi Diagram and Skeleton

It has been observed that the skeleton is embedded in the Voronoi diagram of a polygonal shape [2]. Similarly, the skeleton of a shape described by a discrete sequence of boundary points can be approximated from the Voronoi diagram of the points [3]. Both approaches yield a connected, Euclidean skeleton, but the latter is perhaps more appropriate for images since point sequences are more easily obtained than polygons. Although it is not true in general, if one restricts the shapes to those which are morphologically open and closed with respect to a finite-sized disk, the resulting skeleton approximated from the Voronoi diagram of a finite sampling of the boundary is close to the actual skeleton. In

[1]Acknowledge the funding from the Spanish Ministry of Science and Technology within the 'Ramón y Cajal' program.

this case, the approximation error can be quantified, and be made arbitrarily close to zero.

A. *Voronoi Diagram and Skeleton.*

Consider the set F, closed in R^2. Associated with each point in F is its Voronoi region.

$$V_F(p) = \{q : d(q,p) \leq d(q, F/\{p\})\} \tag{1}$$

The Voronoi diagram of F is the union of the boundaries of all the Voronoi regions.

$$VD(F) = \bigcup_{p \in F} \partial V_F(p). \tag{2}$$

A maximal disk in G is one which is contained in G while not being contained by any other disk in G. Assume that all maximal disks in G are bounded. The skeleton $\sigma(G)$ is the set of centers of maximal disks in G. One desires the skeleton to be a "graph-like" retraction of the original set. In general, this cannot be assured due to the presence of infinitesimal detail. However, it is possible to eliminate these fine structures by assuming a reasonable subclass: the regular sets.

A compact set, K, is said to be $r - regular$ [4] if it is morphologically open and closed with respect to a disk of radius $r > 0$. It is possible to show that ∂K is a disjoint union of closed simple C^2 curves with curvature magnitude no greater than $1/r$. The skeleton of the interior of K is well-behaved and graph-like.

B. *Skeleton-based generalization algorithm*

One issue that needs improvement is the existence of spurious hairs on the skeletons generated. This is a well-known artifact of skeleton generation, where any irregularities in the boundary generate unwanted skeleton branches. Ogniewicz [**?**] attempted to reduce skeletons formed from raster boundary points to a simple form by pruning the leaf nodes of the skeleton until a specified minimum circumcircle was achieved, but with the development of the one-step crust and skeleton algorithm this process may be greatly simplified. Alt and Schwartzkopf [5], as well as Blum [6] showed that leaf nodes of a skeleton correspond to locations of minimum curvature on the boundary. For a sampled boundary curve this means that three adjacent sample points are cocircular, with their centre at the skeleton leaf. If we wish to simplify the skeleton we should retract leaf nodes to their parent node location. This means that we now have four cocircular points instead of three. The retraction is performed by taking the central point of the three defining the leaf node, and moving it towards the parent node of the skeleton until it meets the parent node circumcircle. This smooths outward-pointing salients in the boundary of the object. The same should be done from the other side of the boundary, retracting those salients also. This may displace some of the points involved in the first smoothing step, but as the process is convergent a small number of iterations suffices to produce a smoothed curve having the same number of points as the original, but with a simplified skeleton.

III. Introduction to the Level Set Method and the Fast Marching Method

The level set method was devised by Osher and Sethian as a simple and versatile method for computing and analyzing the motion of the interface in two or three dimensions. The goal is to compute and analyze the subsequent motion of the interface under a velocity field. This velocity can depend on position, time, the geometry of the interface and the external physics. The interface is captured for later time as the zero level set of a smooth (at least Lipschitz continuous) function. Topological merging and breaking are well defined and easily performed.

The original level set idea of Osher and Sethian (Osher [7]) for tracking the evolution of an initial front γ_0 as it propagates in a direction normal to itself with a given speed function V . The main idea is to match the one-parameter family of fronts $\{\gamma_t\}_{t \geq 0}$, where γ_t, is the position of the front at time t, with a one-parameter family of moving surfaces in such a way that the zero level set of the surface always yields the moving front. To determine the front propagation, we then need to find and solve a partial differential equation for the motion of the evolving surface. To be more precise, let γ_0 be an initial front in R^d, $d \geq 2$ and assume that the so-called *level set function* $\phi : \mathrm{R}^d \times \mathrm{R}_+ \to \mathrm{R}$ is such that at time $t \geq 0$ the zero level set of ϕ is the front γ_t. We further assume that $\phi(\boldsymbol{x}; 0) = \pm d(\boldsymbol{x})$; where $d(\boldsymbol{x})$ is the distance from $\boldsymbol{x}$ to the curve γ_0. We use plus sign if $\boldsymbol{x}$ is inside 0 and minus if $\boldsymbol{x}$ is outside. Let each level set of ϕ along its gradient field with speed V . This speed function should match the desired speed function for the zero level set of ϕ. Now consider the motion of, e.g., the level set

$$\left\{\boldsymbol{x} \in \mathrm{R}^d : \phi(\boldsymbol{x}; t) = 0\right\}. \tag{3}$$

Let $\boldsymbol{x}(t)$ be trajectory of a particle located at this level set so that

$$\phi(\boldsymbol{x}(t); t) = 0. \tag{4}$$

The particle speed $d\boldsymbol{x}/dt$ in the direction n normal to the level set is given by the speed function V, and hence

$$\frac{d\boldsymbol{x}}{dt} \cdot n = V. \tag{5}$$

where the normal vector n is given by

$$n = -\frac{\nabla \phi}{|\nabla \phi|}. \tag{6}$$

This is a vector pointing outwards, giving our initialization of u. By the chain rule

$$\frac{\partial \phi}{\partial t} + \frac{d\boldsymbol{x}}{dt} \cdot \nabla \phi = 0. \tag{7}$$

Therefore $\phi(\boldsymbol{x}; t)$ satisfies the partial differential equation (the level set equation)

$$\frac{\partial \phi}{\partial t} - V\left|\nabla \phi\right| = 0, \tag{8}$$

and the initial condition

$$\phi(\boldsymbol{x}; t = 0) = \pm d(\boldsymbol{x}). \tag{9}$$

This is called an Eulerian formulation of the front propagation problem because it is written in terms of a fixed coordinate system in the physical domain.

If the speed function V is either always positive or always negative, we can introduce a new variable (the arrival time function) $T(\boldsymbol{x})$ defined by

$$\phi(\boldsymbol{x}, T(\boldsymbol{x})) = 0. \tag{10}$$

In other words, $T(\boldsymbol{x})$ is the time when $\phi(\boldsymbol{x}; t) = 0$. If $\frac{d\boldsymbol{x}}{dt} \neq 0$, then T will satisfy the stationary Eikonal equation

$$V\left|\nabla T\right| = 1, \tag{11}$$

coupled with the boundary condition

$$T|_{d(\boldsymbol{x})=0} = 0. \tag{12}$$

The advantage of this formulation 11 is that we can solve it numerically by the fast marching method [8], which is precisely what we will do in this paper.

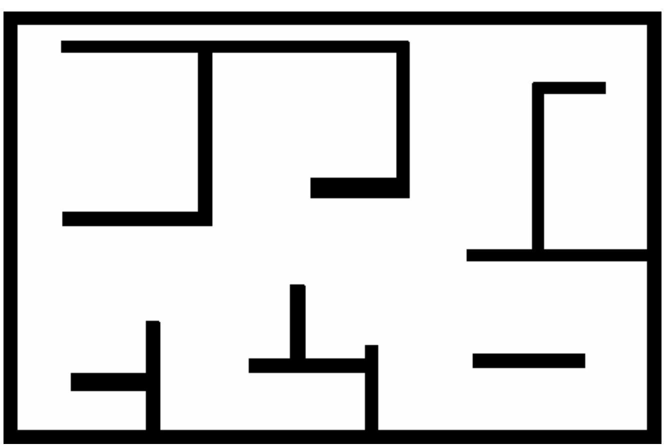

Fig. 1. Map of the room used in the first experiment with its Voronoi Diagram.

Summing up, the central mathematical idea is to view the moving front γ_t as the zero level set of the higher-dimensional level set function $\phi(\boldsymbol{x}; t)$. Depending on the form of the speed function V , the propagation of the level set function $\phi(\boldsymbol{x}; t)$ is described by the initial problem for a nonlinear Hamilton-Jacobi type partial differential equation 7 of first or second order.

If $V > 0$ or $V < 0$, it is also possible formulate the problem in terms of a time function $T(x)$ which solves a boundary value problem for a stationary Eikonal equation 11.

Fast Marching Methods are designed for problems in which the speed function never changes sign, so that the front

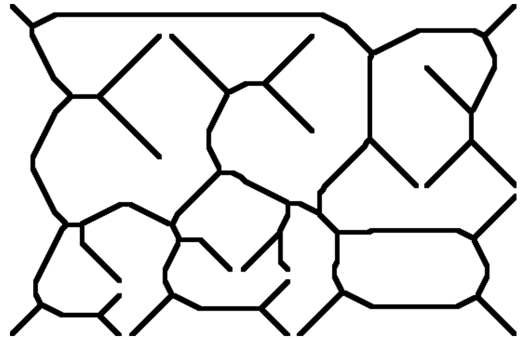

Fig. 2. Enlarged Voronoi Diagram of the room.

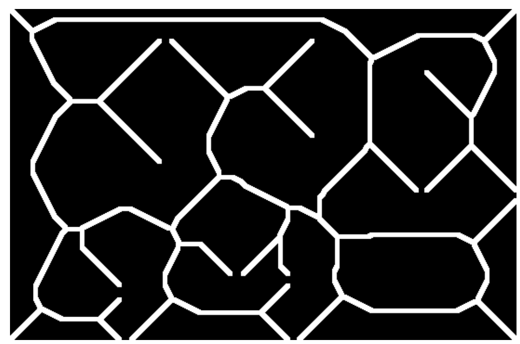

Fig. 3. Inverted image of the Enlarged Voronoi Diagram of the room.

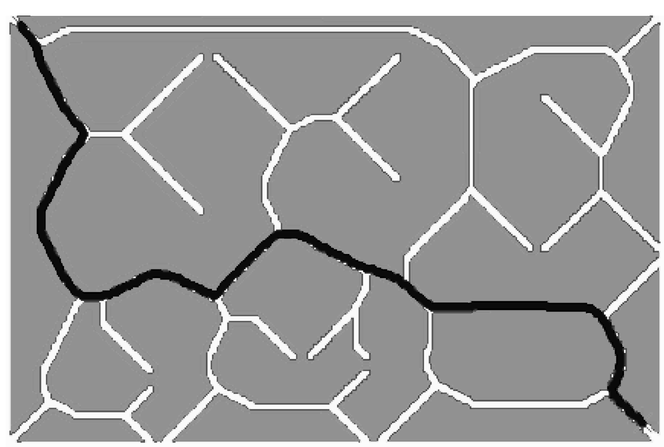

Fig. 4. Trajectory calculated by Fast Marching Method in the inverted image of the Enlarged Voronoi Diagram of the room.

is always moving forward or backward. This allows us to convert the problem to a stationary formulation, because the front crosses each grid point only once. This conversion to a stationary formulation, plus a whole bunch of numerical tricks, gives it its tremendous speed

Level Set Methods are designed for problems in which the speed function can be positive in some places are negative in others, so that the front can move forwards in some places and backwards in others. While significantly slower than Fast Marching Methods, embedding the problem in one higher dimension gives the method tremendous generality.

Because of the nonlinear nature of the governing partial differential equation 7 or 11, solutions are not smooth enough to satisfy this equation in the classical sense (the level set function and the time function are typically only Lipschitz). Furthermore, generalized solutions, i.e., Lipschitz continuous functions satisfying the equations almost everywhere, are not uniquely determined by their data and additional selection criteria (entropy conditions) are needed to pick out the (physically) correct generalized solutions. The correct mathematical framework in which to treat Hamilton-Jacobi type equations is provided by the notion of viscosity solutions (Crandall [9], [10]).

After its introduction, the level set approach has been successfully applied to a wide collection of problems that arise in geometry, mechanics, computer vision, and manufacturing processes, see (Sethian [11]) for details. Numerous advances have been made to the original technique, including the adaptive narrow band methodology (Adalsteinsson and Sethian [12]) and the fast marching method for solving the static Eikonal equation (Sethian [13], [11]). For further details and summaries of level set and fast marching techniques for numerical purposes, see (Sethian [11]). The Fast Marching Method is an $O(n\log(n))$ algorithm.

IV. Implementation of the Method

This method operates in two steps. The first step starts with the calculation of the Voronoi Diagram of the 2D or 3D a priori map of the environment (which are the cells located equidistant to the obstacles). This process is done by means of morphological operations on the image of the environment map. To be more precise, it is done a skeletonization with the image techniques previously mentioned, in order to obtain the Voronoi Diagram. After that, a dilatation is done in order to have a thick Voronoi Diagram where to calculate the propagation of a wave front. This is done in order to obtain two characteristics, on one side the Voronoi Diagram has abrupt changes of gradient, especially in nodes and in other side Fast Marching Method used in next step, is a method designed for more than one dimension.

This way, the enlarged Voronoi Diagram will let the Fast Marching Method in the second step to plan the shortest trajectory. This trajectory is obtained inside the most safe areas provided by the enlarged Voronoi Diagram and properly smooth because the Fast Marching Method selects a continuous path in gradient terms.Besides the path extraction is very fast because the Fast Marching Method propagates in an almost unidimensional curve (it is not completely unidimensional due to the Voronoi Diagram is enlarged in the perpendicular direction to the Diagram curves in some cells).

After that,it is necessary to invert the image because it is necessary to have a viscosity map where the wave goes faster in the clearer zones and slower in the darker ones. The calculation of the evolution of the wave front is done with the Fast Marching Method.

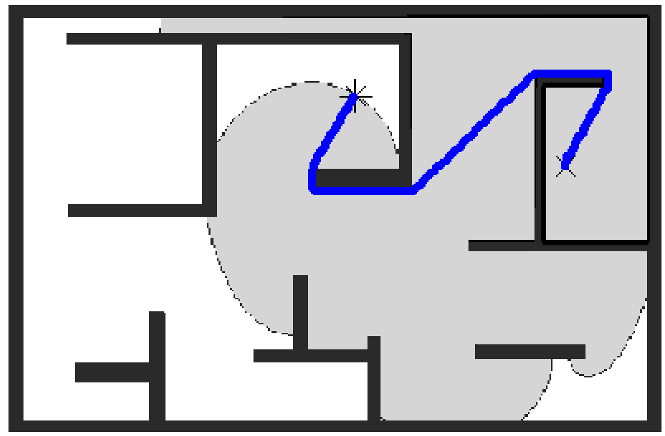

Fig. 5. Trajectory calculated by Fast Marching Method directly, without the previous Voronoi Diagram. The trajectory is not safe because touches the corners and walls.

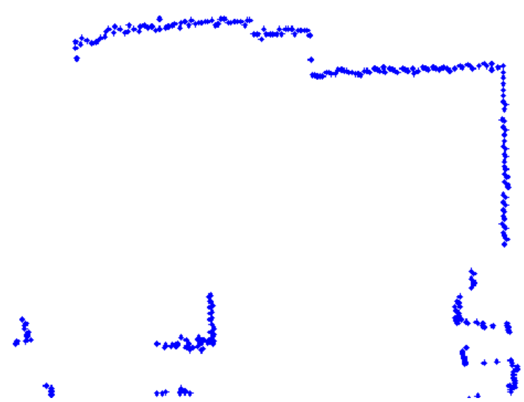

Fig. 6. Raw Laser data read by the robot (Local map).

The method proposed, can also be used for sensor based planning, working directly on a raw sensor image of the environment, as shown in figures 7 and 8.

V. Results

To illustrate the capabilities of the proposed method three test are presented in this section. In the first test, the method is

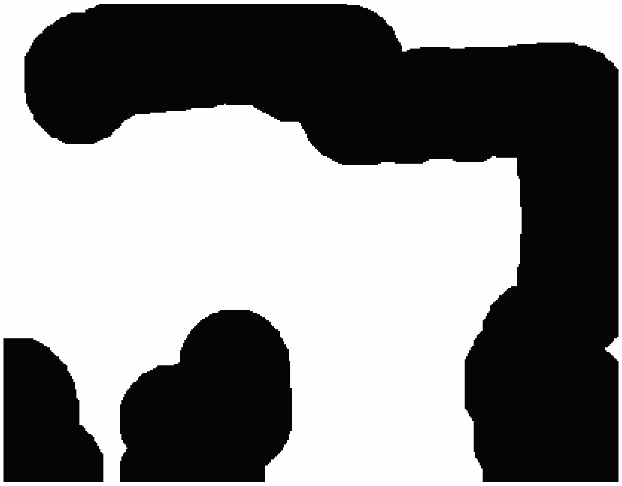

Fig. 7. Enlarged Laser data read by the robot (Local map).

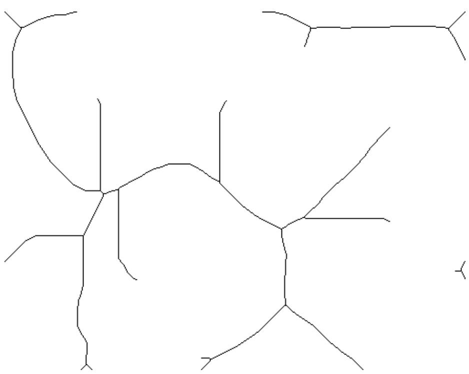

Fig. 8. Voronoi Diagram of the Enlarged Laser data read by the robot (Local map).

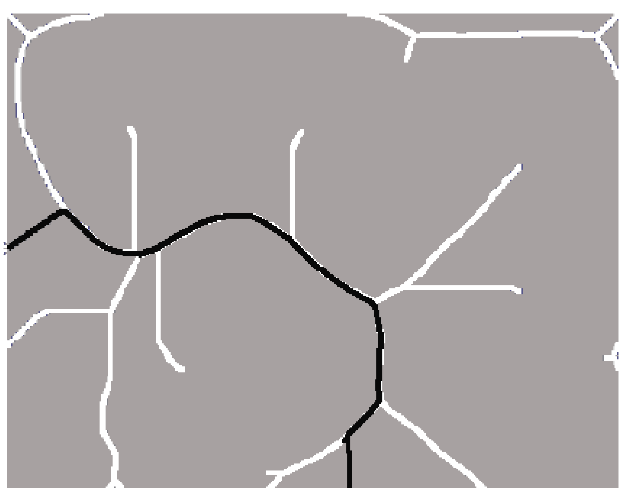

Fig. 9. Trajectory calculated with Fast Marching over the Voronoi Diagram of the Enlarged Laser data read by the robot (Local map).

applied to local environment path planing task where the laser scanner measurement data are used to generate the trajectories. Figures 7 to 10 illustrate the achieved good trade off between trajectory length, distances to obstacles and smooth changes in the trajectory. In the second test, a difficult test room environment and the floor of the laboratory environment have been used. Figures 1 to 5 and 11 to 12 show the capabilities of the method to generate adequate paths on a global scale.

The last test is dedicated to illustrate the capability of the proposed method to adapt to changing environment to accommodate possible dynamic features of the environment such as moving obstacles and persons in the vicinity of the robot. During the motion, the robot observes the environment with its laser scanner, introduces the new information in the map and plan a new trajectory. Local observations (obstacle in the middle of the corridor) generate modified trajectories in order to avoid the detected obstacles. In the last figure the obstacles detected block the corridor and the sensor based global planner generates a completely different safe trajectory. The dimensions of the environment are 116x14 meters (the cell resolution is 12 cm). For this environment the first step (the Voronoi extraction) takes 0.06 seconds in a Pentium 4 at 2.2 Ghz, and the second step (Fast Marching) takes 0.20 seconds for a long trajectory.

The proposed method is highly efficient from a computational point of view because it operates directly over a 2D image map (without extracting adjacency maps), and due to the fact that Marching complexity is $O(n)$ and the Voronoi path calculation is also of complexity $O(n)$, where n is the number of cells in the environment map.

The method provides smooth trajectories that can be used at low control levels without any additional smooth interpolation process.

The results are shown in fig 10 (The environment map of the Robotics Lab.) 11 and fig 12 (the path obtained after applying the Fast Marching method to the previous Voronoi diagram image).

VI. Conclusion

A new global sensor based path planner is presented in this paper. The proposed method is able to deal simultaneously with both global and local planing requirements. The advantages of the approach can be summarized in the fact that obtained trajectories are smooth and safe and at the same time free of local traps due to the integration of the instant sensor information in the recalculation of the path.

The algorithm complexity is $O(n)$, where n is the number of cells in the environment map, which let us use the algorithm on line.

References

[1] J.-C. Latombe, *Robot motion planning*. Dordrecht, Netherlands: Kluwer Academic Publishers, 1991.

[2] D. T. Lee, "Medial axis transformation of a planar shape," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 4, no. 4, pp. 398–401, 1982.

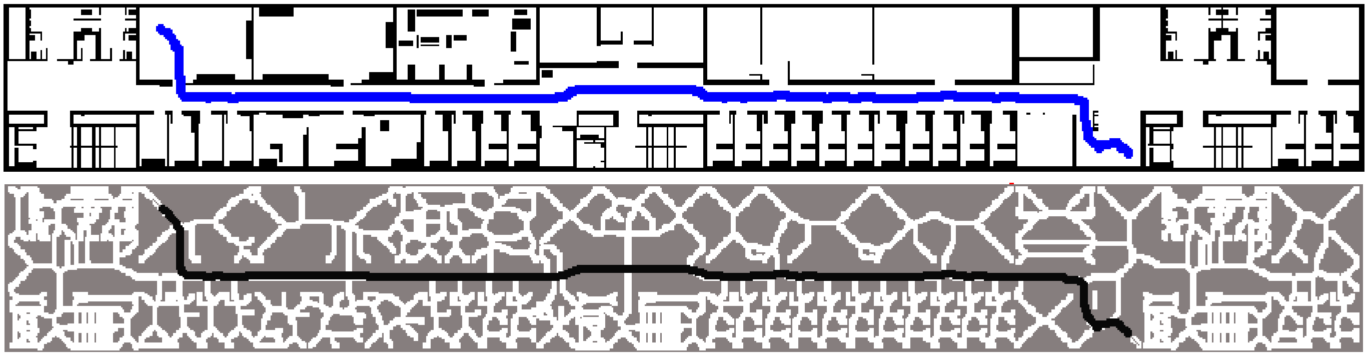

Fig. 10. Trajectory calculated with Fast Marching over the Voronoi Diagram (Global map).

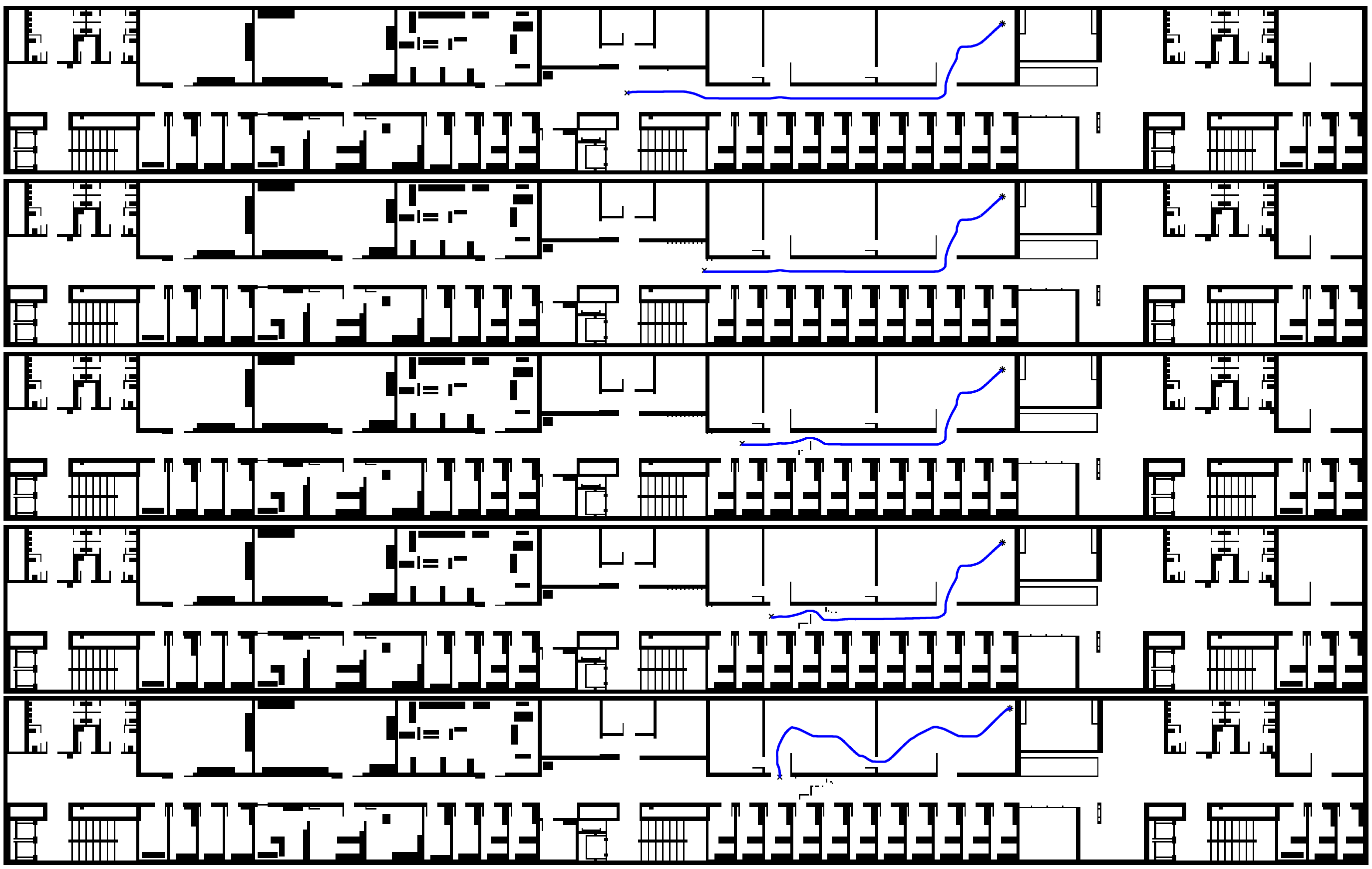

Fig. 11. Evolution of the path when the robot reads information about the new obstacles that are not in the previous map and the robot can't pass trough the corridor.

[3] P. K. J. D. Boissonnat, "Use of the delaunay triangulation for the identification and localization of objects," *IEEE Conf. Computer Vision and Pattern Recognition*, pp. 398–401, 1985.

[4] J. Serra, *Image Analysis and Mathematical Morphology*. London: Academic Press, 1982.

[5] O. S. H. Alt, "The voronoi diagram of curved objects," in *Proc. 11th Annual ACM Symposium on Computational Geometry*, 1995, pp. 89–97.

[6] H. Blum, "A transformation for extracting new descriptors of shape," in *Models for Preception of Speech and Visual Form*, W. W. Dunn, Ed. MIT Press, Cambridge, Mass., 1967, pp. 153–171.

[7] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations," *Journal of Computational Physics*, no. 79, pp. 12–49, 1988.

[8] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nat. Acad Sci. U.S.A.*, vol. 93, no. 4, pp. 1591–1595, 1996.

[9] P. L. L. M. G. Crandall, H. Ishii, "User's guide to viscosity solutions of second order partial differential equations," *Bull. Amer. Math. Soc.*, vol. 27, no. 1, pp. 1–67, 1992.

[10] P. L. L. M. G. Crandall, "Viscosity solutions of hamilton jacobi equations," *Trans. Amer. Math. Soc.*, vol. 277, pp. 1–42, 1983.

[11] J. A. Sethian, "Theory, algorithms, and aplications of level set methods for propagating interfaces," *Acta numerica*, pp. 309–395, 1996, cambridge Univ. Press.

[12] D. Adalsteinsson and J. Sethian, "A fast level set method for propagating interfaces," *J. Comput. Phys.*, vol. 118, no. 2, pp. 269–277, 1995.

[13] J. Sethian, *Level set methods*. Cambridge University Press, 1996.