Garrido, S., L. Moreno. (2015). Mobile Robot Path Planning using Voronoi Diagram and Fast Marching. In: Luo, Zongwei (ed.), *Robotics, Automation, and Control in Industrial and Service Settings* (pp. 92-108). Hershey, PA: IGI Global.

DOI: https://doi.org/10.4018/978-1-4666-8693-9.ch003

Mobile Robot Path Planning using Voronoi Diagram and Fast Marching

S. Garrido and L. Moreno
Carlos III University of Madrid, Spain

ABSTRACT

For navigation in complicated environments, a robot must reach a compromise between efficient trajectories and ability to react to unexpected environmental events. This paper presents a new sensor-based path planner, which gives a fast local or global motion plan capable to incorporate new obstacles data. Within the first step, the safest areas in the environment are extracted by means of a Voronoi Diagram. Within the second step, the fast marching method is applied to the Voronoi extracted areas so as to get the trail. This strategy combines map-based and sensor-based designing operations to supply a reliable motion plan, whereas it operates at the frequency of the sensor. The most interesting characteristics are high speed and reliability, as the map dimensions are reduced to a virtually one-dimensional map and this map represents the safest areas within the environment. Additionally, the Voronoi Diagram is calculated in open areas with all reasonably shaped obstacles. This fact permits to use the planned trajectory methodology in complex environments wherever different Voronoi-based strategies will not work.

Key words: Path Planning, Robot Motion, Voronoi Diagram, Fast Marching.

INTRODUCTION

Mobile robot motion planning has become a very important research topic since the 1980's. Many researchers have worked extensively to get efficient ways to solve issues related to this topic. Two different ways can be followed to solve this problem. The first approach considers that the global map (surroundings and obstacle data) and the robot characteristics are known. The second one uses local sensor data and robot characteristics.

In recent years, sensor-based path planning has emerged as a compromise between efficient optimized trajectories and ability to react to unexpected environmental events. Different variations of the classical methods have been implemented to attain an operative sensor-based path planning. One among these techniques is the Voronoi-based path planning.

Many researchers have studied the Voronoi Diagrams. An advantage of Voronoi-based path planning is that it diminishes the problem to one dimension while the trajectories follow the most clearance map. This implies that the trajectories are the safest ones. Moreover, the Voronoi-based paths can be viewed as a type of topological navigation and, as a result of that, it is like the human one in some aspects. However, Voronoi-based strategies have the typical difficulties of drawing the Voronoi Diagram: finding out lines and polygons, finding the vertices and nodes, and making a tree to find the trail. As an alternative to unravel these issues, we propose the employment of image processing techniques.

In order to calculate the trajectory, the new motion planning technique relies on the mixture of a Voronoi Diagram and the expansion of a wave from the beginning point to the goal following the Voronoi Diagram. It uses local sensor data and also the known map of the surroundings.

The Voronoi Diagram is constructed using image processing methods (skeletonization) based on the Breu technique in 2D (Breu, 1995) and Gagvani in 3D (Gagvani, 1997). The Voronoi Diagram (skeleton) is dilated and inverted to get a road map. The following step is to calculate the trajectory within the image generated by the thick Voronoi Diagram employing a wave expansion. The wave expansion is calculated solving the Eikonal equation. The Fast Marching method has been implemented to solve this equation. There are other similar techniques that can also be used for the same purpose (Tsiksitlis, 1995, Mauch, 2003). After that, the obtained path verifies the smoothness and safety considerations required for mobile robot path planning.
The approach implemented in this work relies on the potential field technique of the wave expansion, which repels a robot far away from obstacles and towards the goal employing a rigorously designed artificial potential function. Since the potential field is like a Lyapunov function, the stability of the system is assured.
Finally, this strategy has been extended to be used with non-holonomic robots. It associates a vector field that permits to use the strategy with non-holonomic robots (the Voronoi Diagram thread has become a road). This is done by employing a Voronoi Diagram thicker than the robot. In this thick diagram, a repulsive potential is made using the propagation of a wave that starts at the edges. This potential field is employed as a slowness or refraction index in a second wave propagation from the goal point up to the actual position of the robot. This potential field is attractive to the goal and repulsive from the perimeters of the thick Voronoi Diagram. There are no different local minima than the destination. This potential and its associated vector field permit the motion planning of the non-holonomic robot.

The main advantages of this technique are:

- It is straightforward to implement.
- The speed is very high.
- The trajectories are smooth.
- The trajectories are the safest ones.
- It works on-line.
- It works with non-holonomic robots.
- The technique is complete, i.e., the strategy is capable of finding a path if it exists.

RELATED WORK

Voronoi Diagram

The Voronoi idea has been used for four centuries. In his 'Traite de la Lumiere', printed in 1644, Rene Descartes uses diagrams just like Voronoi to indicate the disposition of matter within the solar system and its surroundings. Algorithms to build Voronoi Diagrams have appeared since the Seventies. See the surveys by (Aurenhammer, 1991, 2000), (de Berg, 2000), and (Okabe, 1992) on numerous algorithms, applications, and generalizations of Voronoi Diagrams.
The Generalized Voronoi Diagram is the set of points where the distance to the two nearest objects is the same (Choset, 2005). Several algorithms are proposed to construct Generalized Voronoi Diagrams (GVD) using the distance data provided by diverse external sensors like sonar sensors, laser scanners, and stereo cameras. An interesting strategy for construction of a Generalized Local Voronoi Diagram (GLVD) can be found in (Mahkovic, 1998). They use measurements from a laser scanner. First, the points that belong to the same object are clustered, then the Voronoi Diagram is generated, and finally, the perimeters outside the visible region and those that both side generator points belonging to the same object are deleted. In (Sudha, 1999), they construct a Voronoi Diagram from a binary image of the workspace obtained employing a digital camera. This category of algorithms involves progressive construction procedures based on the exploration of unknown surroundings (Nagatani, 1998), (Choset, 2000) and (Choset, 2005). These algorithms do not take into account the encoder errors and they cannot be utilized in a large environment. (Bhattacharya, 2008) provides an algorithm based on the Voronoi Diagram to compute an optimal path between source and destination in the presence of simple disjoint polygonal obstacles. They evaluate the quality of the path based on clearance from obstacles, overall length, and smoothness.

There are some path planning algorithms inspired by the Voronoi Diagram idea. An example is the MAPRM methodology (Wilmarth, 1999), which retracts sampled configurations onto the medial axis (or GVD). The EquiDistance Diagram (EDD) (Keerthi, 1999) is based on the Voronoi roadmap idea. It consists of a roadmap shaped by connecting the local maxima of a clearance function. The main disadvantage of these strategies is that the roadmap is built offline and the environment information should be provided earlier. (Bortoff, 2000) proposes a two step path-planning algorithm for UAVs. The algorithm generates a stealthy path through a set of enemy radar sites of known location, and provides an intuitive way to trade-off stealth versus path length. In the first step, a suboptimal rough-cut path is generated through the radar sites by constructing and searching a graph based on Voronoi polygons. In the second step, a set of nonlinear Ordinary Differential Equations (ODE) is simulated, using the graph solution as an initial condition. The ODE describe the dynamics of a set of virtual masses located in a virtual force field. The virtual forces push the masses away from the radars and toward one another.

(Breu, 1995) presents a linear time (and therefore asymptotically optimal) algorithmic rule for computing the Euclidean distance transform of a two-dimensional binary image. The algorithm relies on the construction and regular sampling of the Voronoi Diagram whose sites consist of the unit (feature) pixels within the image. Their algorithmic rule constructs the Voronoi Diagram wherever it intersects the horizontal lines passing through the image constituent centers. They additionally discuss the extensions to higher dimensional images and to different distance functions. The strategy proposed in this paper relies on this algorithmic rule.

Introduction to Voronoi Diagram and Skeleton

The Voronoi Diagram concept can be explained in an intuitive way. Given a finite set of objects in an area, all locations within the area are related to the nearest member of the set. The result is a partition of the area into a group of regions, Voronoi regions. The GVD is formed by the frontier between these regions. Given its widespread use, it is not stunning that this idea has been discovered persistently in many alternative places.

The close relationship between the Voronoi Diagram and the Medial Axis has already been reviewed in literature (Ogniewicz,1995). Within the field of computer vision, the skeleton of an object or image is called the Medial Axis transform. It coincides with the definition of the GVD utilized in AI. This is often to not be confused with the Voronoi Diagram of a distinct set of purposes: for every point x within the set P, there exists a boundary polygonal enclosing all the intermediate points lying nearer to x than to alternative points within the set P, and also the set of those polygons for a given purpose set is named the Voronoi Diagram.

A very illustrative definition of the skeleton is given by the prairie-fire analogy: the boundary of an object is ready on fire and also the skeleton is formed from the loci wherever the fire fronts meet and quench one another.

Based on Blum's definition of a skeleton (Blum, 1967), the skeleton S of a group D is the locus of the centers of maximal disks. A maximal disk of D could be a closed disk contained in D that is interiorly tangent to the boundary δD, which is not contained in the other disk in D. Every maximal disc should be tangent to the boundary in a minimum of two completely different points. With each skeleton point $s \in S$ we have a tendency to conjointly store the radius r(s) of its maximal disk. The skeleton σ(D) is the set of centers of maximal disks in D. The skeleton is desired to be a 'graph-like' retraction of the first set. For a good overview of skeletonization techniques, see (Smith, 1987).

A class of skeletonization approaches relies on the Distance Transform (DT). In this paper, algorithms based the DT are used to compute the skeleton (or Voronoi Diagram) of the binary image that represents the environment. This associates each pixel in the image with its distance to the closest obstacle (See an example in Fig. 3). The DT is vital for natural neighbor interpolation of a function and for morphological operations on images, such as Medial Axis and skeleton computation, and polygonal shape morphing. Several efficient algorithms exist for the computation of the DT, or approximations of this unit (meaning that the distances are not precisely the true distances), for images. The Medial Surface/Axis can be defined as the locus of greatest circles (2D) or ball (3D). A very efficient algorithm that has been used for the calculation of the Voronoi Diagram (2D) in this work is thanks to (Breu, 1995). For its approach, it is not necessary to calculate vertices and nodes of the Voronoi Diagram and it may be used for all types of walls and obstacles, even curved ones. It runs in O(m) time, wherever m is that the number of image pixels. This algorithmic rule relies on the construction and regular sampling of the Voronoi Diagram whose sites consist of the unit (feature) pixels within the image. The algorithm constructs the Voronoi Diagram wherever it intersects the horizontal lines passing through the image pixel centers. This algorithm has been used to calculate the Voronoi Diagram in 2D environments.

An important disadvantage of the skeletonization strategies is the existence of spurious branches on the generated skeletons. This unwanted fact is caused by the irregularities of the boundary. Boundary noise would cause the Voronoi Diagram to be very dense, which might result in the necessity to prune it to get the medial axis. (Ogniewicz, 1995) reduced skeletons shaped from raster boundary points to a straightforward form. This was achieved by pruning the hair nodes of the skeleton till a given minimum circumcircle was reached. On the other hand, it has been demonstrated that hair nodes correspond to a location of minimum curvature on the boundary. This implies that for a sampled boundary curve, three adjacent points are co-circular, with their center at the skeleton hair. For the simplification of the skeleton, the hairs ought to be retracted to their parent node location. The central point of the three is moved towards the parent node till it coincides with the parent node circumcircle, leading to smoothing outward-pointing salients of the boundary.

The process is iterated from the other side of the boundary, additionally retracting these salients. This could displace points involved within the first smoothing step. However, the method is convergent and a small number of iterations suffices to supply a smoothened curve having a similar number of points as the original one, but with a simplified skeleton.

In order to extract the skeleton in 3D, the algorithm presented in (Gagvani,1997) has been implemented. The skeletonization methodology for 3D objects relies on a quasi-Euclidean DT. Once the DT value at each object point is known, a linked list of all the object points is made. One pass is completed over the list to check if a local maximum of it satisfies the condition:

$$MNTp < DTp - TP, \quad (1)$$

where TP is a thinness parameter, DTp is the DT of the voxel p, and MNTp is the mean of the neighbors' DT.

Selecting a thinness parameter that removes most of the 'hairs' within the skeleton reduces the impact of boundary noise.

INTRODUCTION TO FAST MARCHING

The Fast Marching Method (FMM) is a computational algorithm to find the time of arrival to every point of the area of an expanding wave that starts in a point or a set of points. Conceptually, it may be thought as a continuous version of the Dijkstra's algorithm, (Dijkstra, 1959). It is based on the assumption that the information solely flows outward from the seeding area (wave source).

The FMM was proposed by J. A. Sethian (Sethian, 1996) to approximate the solution of the Eikonal equation, a non-linear partial equation encountered in the propagation of an electromagnetic wave, such as light.

Let assume a 2D map, where x = (x, y) is a point on the map with its coordinates defined in a Cartesian reference system, T(x) is a front wave arrival time function for every point of the map, and F(x) is the velocity of the wave propagation in each point x. Let also assume that a wave starts propagating at $x_0 = (x_0;y_0)$ at time $T(x_0) = 0$ with velocity $F(x) \geq 0$. The Eikonal equation defines the time of arrival of the propagating wavefront T(x) at each point x of the map, in which the propagation speed depends on the point F(x), according to:

$$|\nabla T(x)|\, F(x) = 1 .$$

With the discretization of the gradient $\nabla T(x)$ (Chiang, 1992), it is possible to solve the Eikonal equation at each point x. Simplifying the notation, the last equation becomes :

$$T_1 = min\left(T\left(i-1, j\right); T\left(i+1; j\right)\right)$$
$$T_2 = min(T\left(i; j-1; T\left(i; j+1\right)\right)$$
$$\left(\frac{T(i,j)-T_1}{\Delta x}\right)^2 + \left(\frac{T(i,j)-T_2}{\Delta y}\right)^2 = \frac{1}{F(i,j)^2}$$

The FMM consists on the solution the last equation, when all the parameters are known except T(i; j). This method is iterative, beginning at the source point of the wave (or waves) where T(i0; j0) = 0. The next iteration solves the value T(i; j) for the neighbors of the points solved in the previous iteration. Using as an input a binary grid map, in which velocity F(i; j)=0 (black) means obstacle and F(i; j)=1 (white) means free space, the output of the algorithm is a map of distances. These distances are equivalent to the time or arrival of the expanding wave at each point of the map, as shown in Fig. 1.
The FMM is often directly used as a path planner algorithm. By applying gradient descent from any point of the distance map, a path can be obtained with the source of the wave as a goal point. The paths provided by the FMM are optimal in distance terms, but they do not accomplish the smoothness and safety constraints that most robotic applications need. These paths run too near to obstacles and walls and have sharp curves.

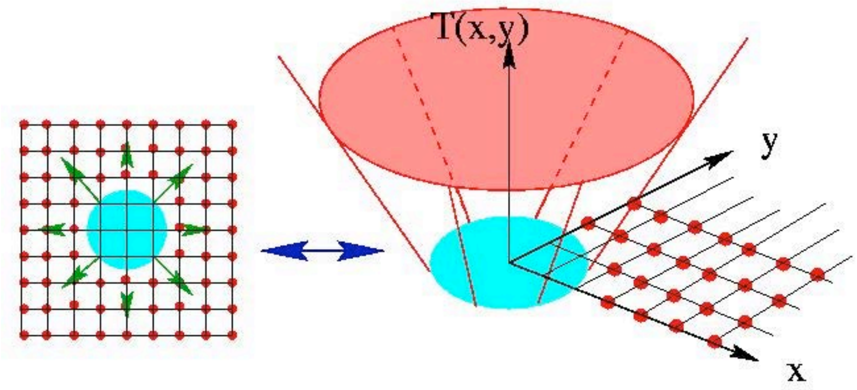


Fig. 1 The function T(x,y) gives a cone-shaped surface. The height T gives the set of points reached at time T. The surface on the right below is called the arrival time surface, because it gives the arrival time.

IMPLEMENTATION OF THE METHOD

This methodology operates in two main steps. The first step consists on the calculation of the Voronoi Diagram of the 2D or 3D map of the environment (that is, the cells located equidistant to the obstacles). This method is completed by means of morphological operations on the image of the environment map. In the second step, the FMM is applied to make a potential T(x) that represents the propagation of an electromagnetic wave. This potential is employed to calculate the trajectories based on the potential surface defined by the slowness map.
To be more precise, the planner follows the subsequent steps:

- **Modeling.** A grid-based map of the environment is the model to be used. The a priori known map (if it is known) and, therefore, the sensor data, are integrated in an updated image of the environment with black and white cells. The sole criteria to select the grid size is the computation time. The grid does not need to be uniform, being attainable to use a triangular mesh.

- **Object enlarging.** To avoid impractical trajectories, the objects and walls are enlarged by the radius of the mobile robot before the calculation of the Voronoi Diagram to make sure it neither collides with obstacles or walls nor accepts passages narrower than the robot size. This permits robot to be considered as as a point from now on.
- **Filter.** In order to get rid of the tiny 'hairs' of skeletons, an averaging filter is applied. This filter is applied over the binary image to eliminate the corners and avoid the appearance of hairs.
- **Voronoi Diagram.** Morphological image process techniques are applied to the map to create the Voronoi Diagram. To be precise, a skeletonization of the image based mostly in the (Breu, 1995) methodology (2D) and (Gagvani, 1997) (3D) is applied to get the Voronoi Diagram as shown in Fig. 3(a). This skeleton is an image in which the points of the Voronoi diagram are black and the rest of the points are white. There are no vertices/nodes. It is possible to calculate the nodes, however it's more sensible to search the path with the image with the FMM, because the calculation time is smaller and it works even with curved obstacles and walls.
- **Dilatation.** Afterwards, a dilatation is completed to have a thick Voronoi Diagram (see Fig. 3(b)) wherein to calculate the propagation of the wave front of the FMM. This is often done in order to get 2 advantages. First, the sharp angles between segments of the diagram are smoothened, which improves the continuity of the generated trajectories and frees them from sharp angles. Second, the Voronoi Diagram is thickened to eliminate excessive narrowing, and thereby to permit a far better propagation of the wave front after applying the FMM. In this way, the path is calculated. This thickening of all the lines of the Voronoi Diagram is completed within the same amount. However, the grade of dilatation is the solely design parameter of the method. It is small enough to have no isolated points or large enough to have a tube in which non-holonomic vehicles maneuver.
- **Viscosity Map**. In order to use the FMM, it is necessary to invert the image (see Fig. 4(a)) to get a difficulty (or slowness) map since the wave travels faster within the clearer zones and slower within the darker ones.
- **Fast Marching methodology.** The next step is to calculate the flight in the image generated by the Voronoi Diagram employing a wave expansion. The wave equation is approximated using the paraxial approximation utilized in optics. This way, the Eikonal equation is obtained. To solve the Eikonal equation, the FMM has been used. The FMM is employed to make a potential T(x) that represents the propagation of an electromagnetic wave in a viscosity map. The time is added as a third axis in 2D or the fourth axis in 3D. The origin of the wave is the goal point, which continues propagating till it reaches the present position of the robot. The FMM produces a funnel-shaped surface in which the time is the last axis, i.e., T (x).
- **Path Calculation.** This potential field is employed to calculate the trajectories based on the potential surface defined by the slowness map using the gradient technique with the present position of the robot as the starting point and the goal position as the final point (see Fig. 4(b)). The gradient determines the direction of travel once the FMM has calculated T (x).

The algorithm steps are summarized in the flowchart of Fig. 2. The path is obtained inside the safest areas provided by the thickened Voronoi Diagram and is correctly smooth, particularly at the angles, since the fast marching technique selects a continuous path in gradient terms. Moreover, trajectory extraction is extremely fast as a result of the fast marching technique propagates in a nearly one-dimensional curve (although this is often an approximation since the Voronoi Diagram is thickened within the perpendicular direction to the diagram curves). The proposed method can also be used for sensor-based planning, working directly on a raw laser sensor image of the environment, as shown in Fig. 5 and 6. The data shown corresponds to the corner between two perpendicular aisles of the Carlos III University to the left of the robot, which is located in the center of the bottom of the scene. The image obtained from raw data may contain scattered pixels (see Fig. 5(a)), and the dilatation of the image achieves the continuity and thickening of the lines representing the walls and obstacles (see Fig. 5(b)). There are similar spurious lines behind the walls because the Voronoi Diagram is made with an open image, but the path followed by the robot is correct. At this point the path planning steps described above are applied in the same sequence. The capability of using raw sensor data combined with the speed of the calculation allows this methodology to be used online to recalculate the trajectory at any time, avoiding this way dynamic obstacles and objects not present in the original map.

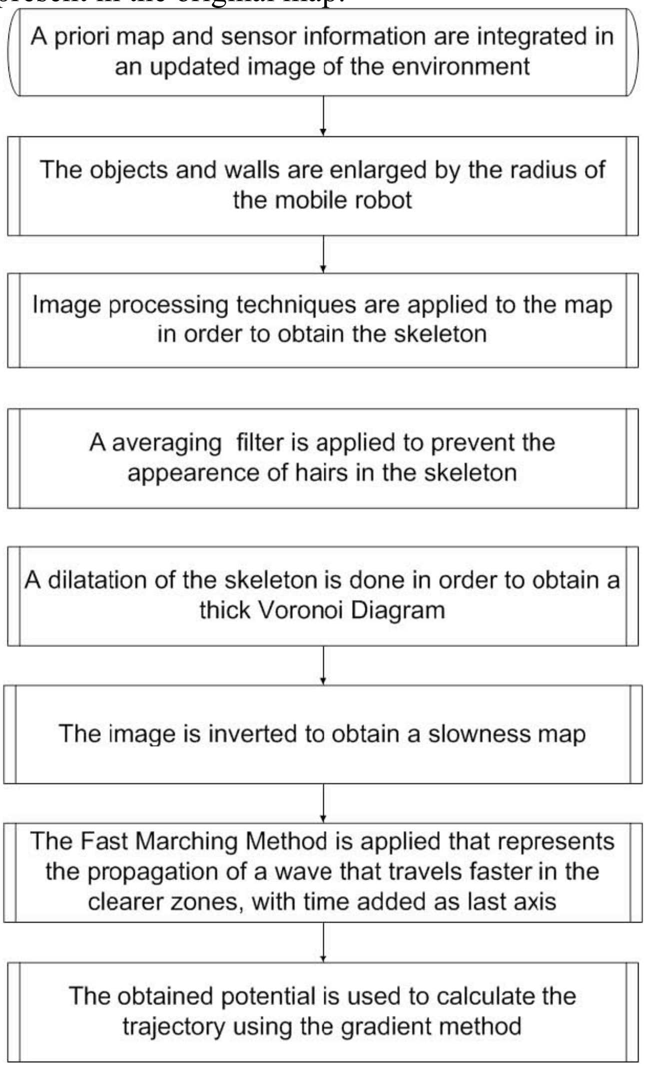
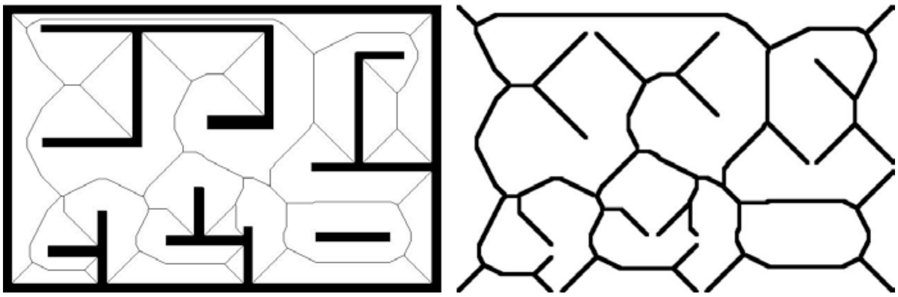
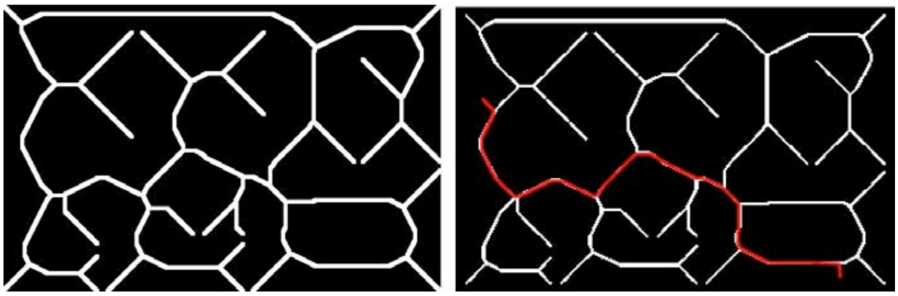


Figure 2: Flowchart of the proposed planning algorithm

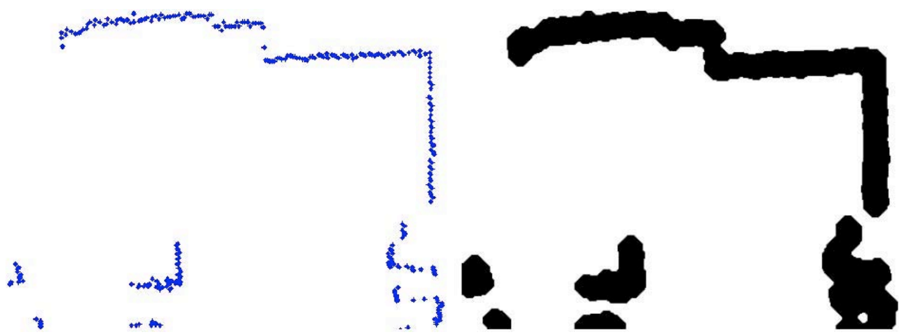


(a)Voronoi Diagram (b) Thickened Voronoi Diagram
Figure 3: Map of the room used in the first experiment

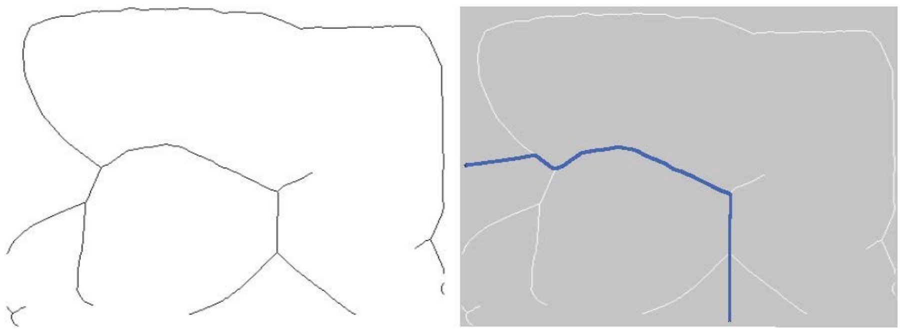


(a) Inverted image of the thickened Voronoi Diagram (b) Trajectory calculated by Fast Marching Method

Figure 4: Trajectory calculated by Fast Marching Method in the inverted image of the thickened Voronoi Diagram of the room



(a) Raw laser data (b) Thickened laser data
Figure 5: Data read by the robot (Local map)



(a) Voronoi Diagram (VD) (b) Trajectory calculated with FM over the VD
Figure 6: Voronoi Diagram and Trajectory of the Local map

RESULTS

The proposed method has been tested using the manipulator robot MANFRED. It has a coordinated control of all the Degrees Of Freedom (DOF) in the system (the mobile base has 2 DOF and the manipulator has 6 DOF) to achieve smooth movement. This mobile manipulator uses a sensorial system based on vision and 3D laser telemetry to perceive and model 3D environments. The mobile manipulator will include all the capabilities needed to localize and avoid obstacles and to navigate safely through the environment.

Figure 7: Trajectory calculated with Fast Marching over the Voronoi Diagram (Global map)

To illustrate the capabilities of the proposed method, different tests are presented in this section. In the first test (room test with a resolution of 628 ×412 pixels), a difficult test room environment and the floor of the laboratory environment have been used (Fig. 7).
In the second test (laser test), the method is applied to a local environment path planning task where the laser scanner measurement data are used to generate the trajectories, as it was explained in the previous section. Fig. 5 and 6 illustrate the achieved good trade-off between trajectory length, distances to obstacles and smooth changes in the trajectory.
The method has capabilities to generate adequate paths on a global scale as shown in Figs. 7 and 8. The results for the complete lab floor are shown in fig. 7 top (the environment map of the Robotics Lab of the Carlos III University) and Fig. 8 bottom (the path obtained after applying the FMM to the previous Voronoi Diagram image). The method provides smooth trajectories that can be used at low control levels without any additional smooth interpolation process.
The last test (lab test) is dedicated to illustrate the capability of the proposed method of adapting to a changing environment taking into account possible dynamic features of the environment such as moving obstacles and persons in the vicinity of the robot, as shown in Fig. 8. During motion, the robot observes the environment with its laser scanner, introducing new information in the map and planning a new trajectory. Local observations (obstacle in the middle of the corridor) generate modified trajectories in order to avoid the detected obstacles. In the bottom map of the 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 m (the cell resolution is 12 cm). The image resolution is 966 ×120 pixels. For this environment (lab floor) the first step (the Voronoi extraction) takes 0.05 s in a Pentium 4 at 2.2 Ghz, and the second step (Fast Marching) takes 0.15 s for a long trajectory, which makes a total of 0.20 s to apply the Voronoi FM method.
The proposed method is highly efficient from a computational point of view because it operates directly over a 2D image map (without extracting adjacent 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.

As can be seen in the Fig. 8, the algorithm is perfectly able to make a correct planning of trajectories.

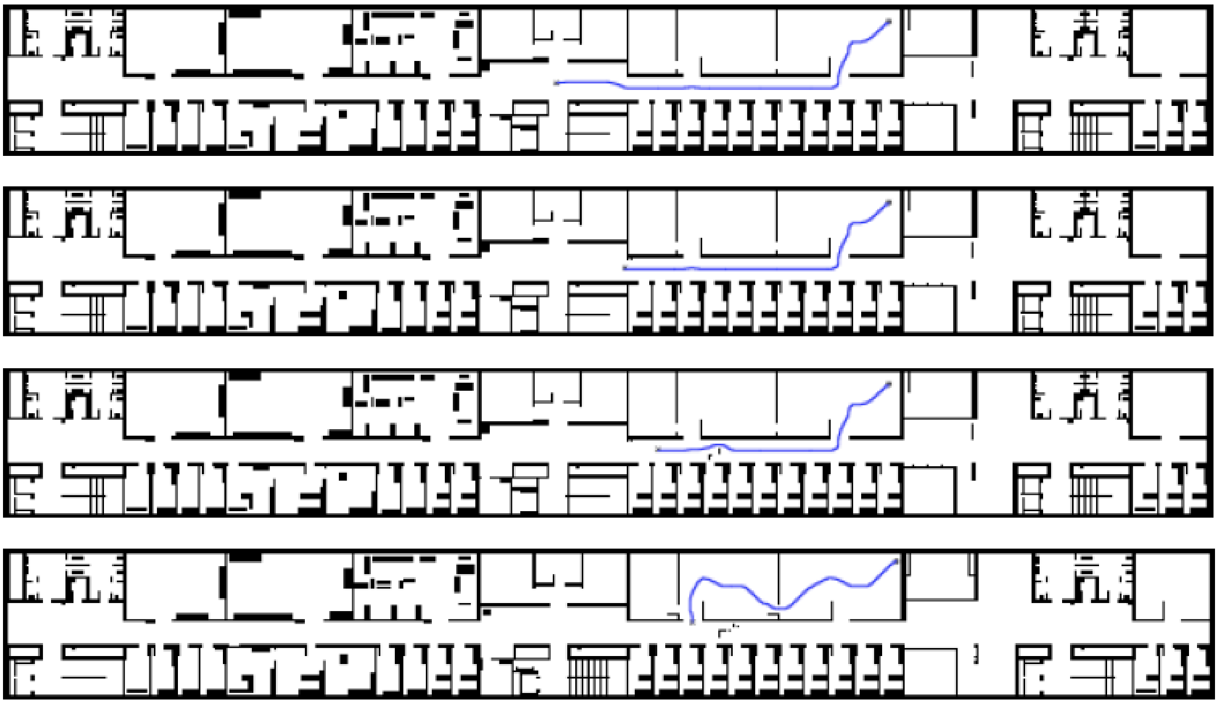


Figure 8. Evolution of the path when the robot reads information about the new obstacles that are not in the previous map and the robot can no pass through the corridor

A great advantage of this method is that it can work with any kind of shaped objects, as demonstrated in Fig. 9. The curvature in the shape of the obstacle is not a restriction for the application of the method. In Fig. 10, the proposed method has been applied to an open environment not limited totally by walls, in which there are circular obstacles. This map represents a corner with columns. As can be seen in the figure, the algorithm is perfectly able to make a correct planning of trajectories.
Fig. 11 shows the application of the proposed method to find a path in 3D. The FMM can even work in more dimensions, but the method must undergo the problem of the dimensionality.

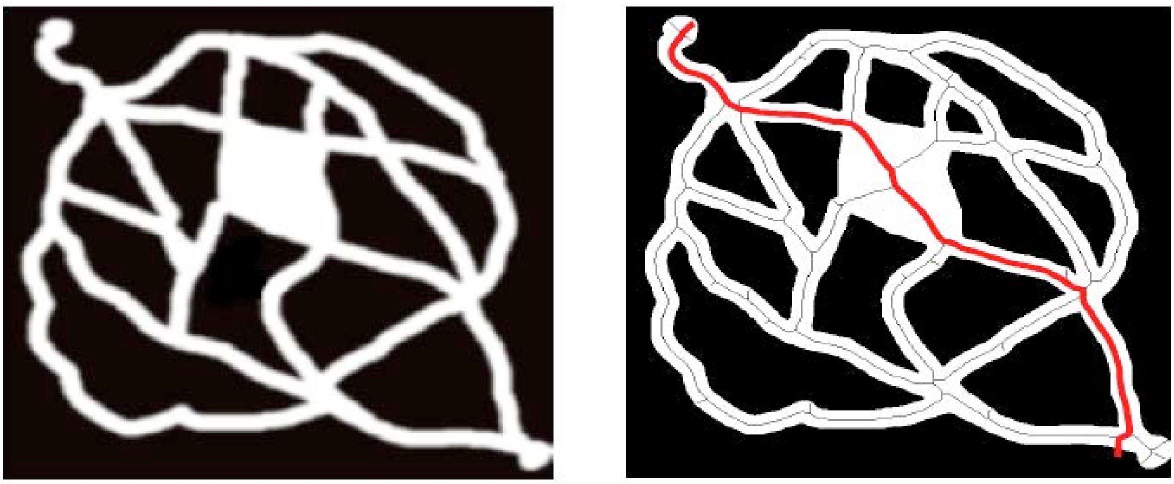


Figure 9. Voronoi Diagram and trajectory in a environment with curved shapes

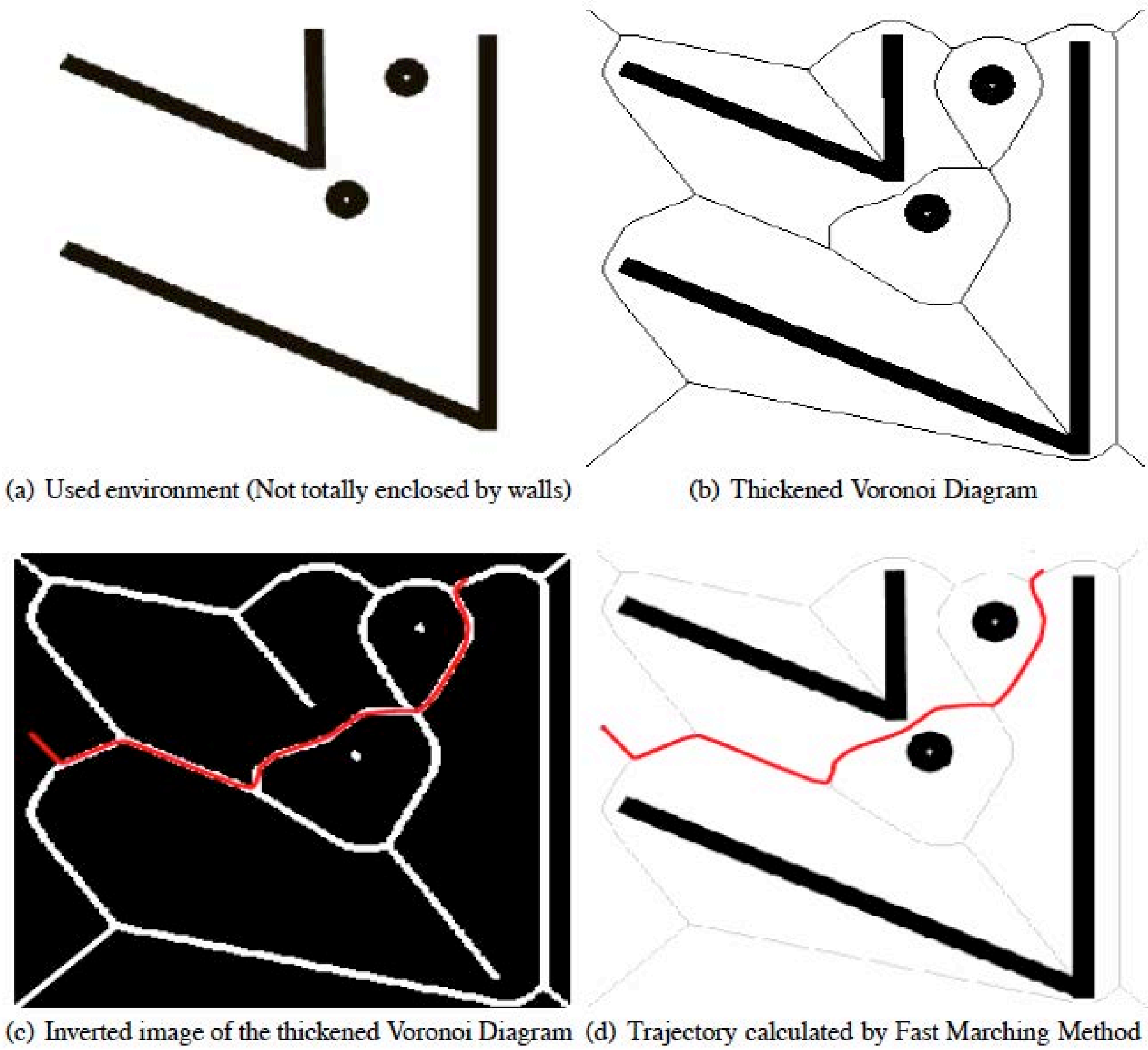


Figure 10. Trajectory calculated by Fast Marching Method in the Voronoi Diagram of an open aisle with columns

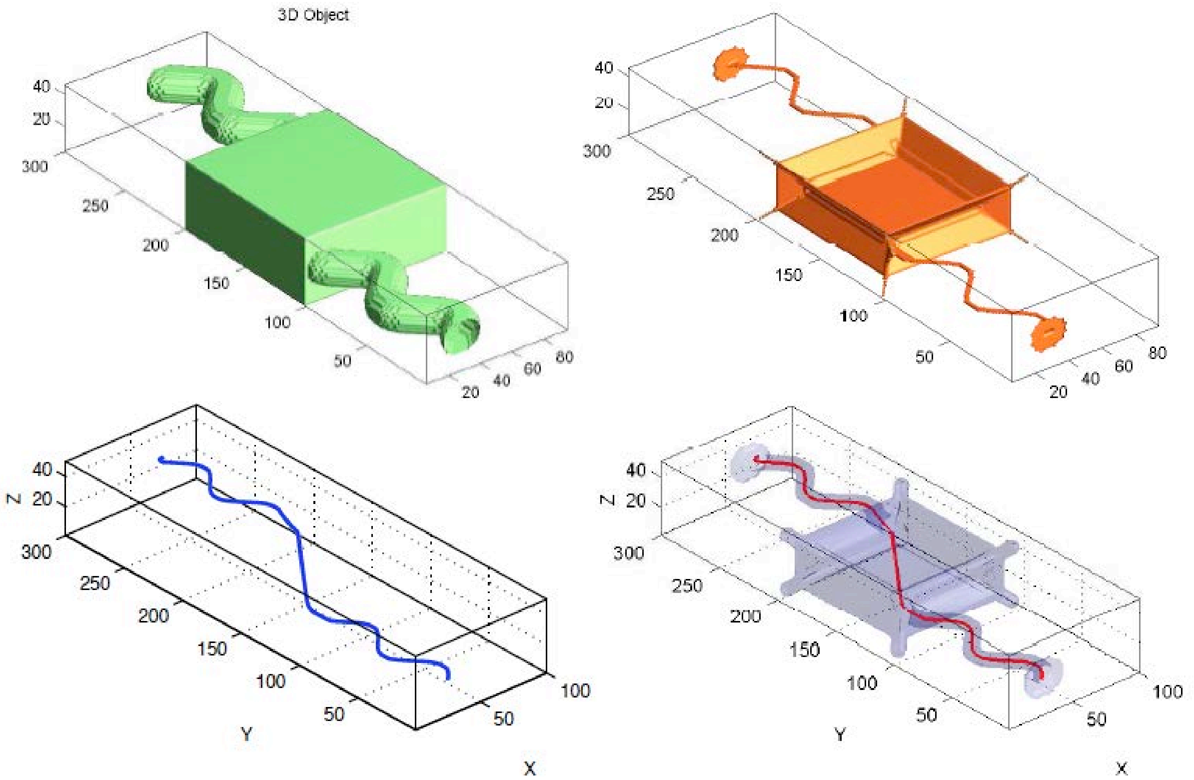


Figure 11. Trajectory calculated with the proposed method in 3D

CONCLUSIONS

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

The method is easy to implement, the algorithm is very fast and it can work online. It works in cluttered and changing environments with moving obstacles. The method is complete, i.e., the method is capable of finding a trajectory if it exists.

As demonstrated along this work, the method can perform in all types of environments without restrictions in the form of the obstacles. The planner works with curved forms, open environments (not totally enclosed by walls), and concavities.

The other main advantage of this method is that non-holonomic restrictions, such as steering angle limits, can be easily incorporated into the algorithm and it still generates smooth trajectories.

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

Adalsteinsson, D., & Sethian, J. (1995). A fast level set method for propagating interfaces. *J. Comput. Phys.*, *118*(2), 269-277.

Aurenhammer, F. (1991). Voronoi diagrams: A survey of a fundamental geometric data structure. *ACM Computing Surveys*, *23*, 345-405.

Aurenhammer, F., & Klein, R. (2000). Handbook of computational geometry. In J. Sack & J. Urrutia (Eds.), 201-290.

Bhattacharya, P., & Gavrilova, M. L. (2008). Roadmap-based path planning-Using the Voronoi diagram for a clearance-based shortest path. *Robotics & Automation Magazine, IEEE, 15(2),* 58-66.

Berg, M. de, Krefeld, M. van, Overmars, M., & Schwarzkopf, O. (2000). *Computational geometry: Algorithms and applications, 2nd rev.* Springer.

Bortoff, Scott A. (2000). "Path planning for UAVs." *Proceedings of the 2000 American Control Conference,.. Vol. 1. No. 6. IEEE.*

Breu, H., Gil, J., Kirkpatrick, D., & Werman, M. (1995). Linear time euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *17*(5), 529-533.

Blum, H., (1967). “A transformation for extracting new descriptors of shape,” *in Models for Perception of Speech and Visual Form, W. W. Dunn, Ed. MIT Press, Cambridge, Mass*., pp. 153–171.

Chiang, C. S., (1992). “*The Euclidean Distance Transform*,” Ph. D. Thesis, Dept. Comp. Sci. Purdue University.

Choset, H. (1996). *Sensor based motion planning: The hierarchical generalized Voronoi graph*. Doctoral dissertation, California Institute of Technology,Pasadena, California.

Choset, H. (2005). *Principles of robot motion: Theory, algorithms, and implementations*. The MIT Press.

Davis, J. L. (1988). *Wave propagation in solids and fluids*. Springer.

Dijkstra, E. (1959). A note on two problems in conexion with graphs. *Numerische Mathematik*, *1*, 269-271.

Gagvani, N., & Silver, D. (1997). *Parameter controlled skeletonization of three dimensional objects* (Tech. Rep.). CAIP-TR-216, Rutgers State University of New Jersey.

Garrido, S., Moreno, L., Abderrahim, M., & Martin, F. (2006). Path planning for mobile robot navigation using Voronoi diagram and Fast Marching. In *Proc of IROS’06. Beijing. China.* (p. 2376-2381).

Garrido, S., Moreno, L., & Blanco, D. (2006). Voronoi diagram and Fast Marching applied to path planning. In *Proc. of ICRA* (p. 3049-3054).

Garrido, S., Moreno, L., & Blanco, D. (2007). Sensor-based global planning for mobile robot navigation. *Robotica*, *25*, 189-199.

Keerthi, S., Huang, C., & Gilbert, E. (1999). Equidistance diagram- a new roadmap method for path planning. In *Proc. ieee int. conf. on Robotics and Automation* (p. 682-687).

Mahkovic, R., & Slivnik, T. (1998). Generalized local Voronoi diagram of visible region. In *In proc.*

IEEE int. conf. on Robotics and Automation, pp. 349-355, Leuven, Belgium.

Mauch, S. (2003). *Efficient algorithms for solving static hamilton-jacobi equations*. Doctoral dissertation, California Inst. of Technology.

Nagatani, K., Choset, H., & Thrun., S. (1998). Towards exact localization without explicit localization with the generalized Voronoi graph. In *Proc. IEEE int. conf. On Robotics and Automation* (p. 342-348). Leuven, Belgium.

Ogniewicz, R., & Kubler, O. (1995). Hierarchic Voronoi skeletons. *Pattern Reconition*.

Okabe, A., Boots, B., & Sugihara, K. (1992). *Spatial tessellations: Concepts and aplications of Voronoi diagrams". , 1992.* (J. Wiley & Sons, Eds.). Chichester, UK.

Sethian, J. (1996). *Level set methods*. Cambridge University Press.

Smith, R. W. (1987). Computer processing of line images: A survey. *Pattern Recognition*, *20*, 7-15.

Sudha, N., Nandi, S., & Sridharan, K. (1999). A parallel algorithm to construct Voronoi diagram and its vlsi architecture. In *Proc. IEEE int. conf. on Robotics and Automation, pp. 1683-1688.*

Tsitsiklis, J. N. (1995). Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, *40*, 1528-1538.

Wilmarth, S. A., Amato, N. M., & Stiller, P. F. (1999). MAPRM: A probabilistic roadmap planner with sampling on the medial axis of the free space. In *Proc. IEEE int. conf. on Robotics and Automation* (p. 1024-1031).

Yatziv, L., Bartesaghi, A., & Sapiro, G. (2005). A fast O(n) implementation of the Fast Marching algorithm. *Journal of Computational Physics*, *212*, 393-399.