22

Sensor-based Global Planning for Mobile Manipulators Navigation using Voronoi Diagram and Fast Marching

S. Garrido, D. Blanco, M.L. Munoz *, L. Moreno and M. Abderrahim
University Carlos III of Madrid,
Spain

** Polytechnic University of Madrid, Boadilla del Monte,*
Spain.

1. Introduction

Mobile Manipulators are special cases of mobile robots that need a careful consideration when planning their motion. Due to their higher and changing centre of gravity caused by the position of the manipulator and its configuration, abrupt motion or change of direction can have dangerous consequences and the toppling of the whole mobile manipulator. Therefore, the motion planning process has to produce smooth and safe trajectories. Since the 1980's mobile robot motion planning problems have been an important research topic that has attracted the attention of many researchers who worked extensively to obtain efficient methods to solve these problems. Such problems have been approached in two general ways: one approach has concentrated on solving motion planning problems by considering a previously known global environment or obstacles information and robot characteristics; the second approach has focused on planning motions by using local sensor information and robot characteristics. The first approach has been used extensively at a global planning level in robotics to plan trajectories from one environment location to another. This plan functions under the assumption of a perfectly controlled and modelled environment that in most situations does not exist. Unexpected obstacles, persons or moving elements make this approach difficult to utilize except in robustly controlled environments, such as in industrial manipulation environments (manufacturing). The second approach became evident in mobile robotics from the very beginning. A mobile robot's environment presents unexpected objects which makes it impossible to use the first approach exclusively. The second approach produces local plans by using local sensor information to avoid obstacles. Obviously a local plan is a solution for a local scale problem and needs to be integrated with a global planner or with global information to guarantee the existence of a solution to the global problem. This method received different names such as collision avoidance method~~s~~, local planner~~s~~, and navigation method~~s~~.
To navigate in complex environments~~,~~ an autonomous mobile robot needs to reach a compromise between the need for reacting to unexpected events and the need for efficient and optimized trajectories. Usually, path planning methods work in two steps: in the first

Source: Mobile Robots: Perception & Navigation, Book edited by: Sascha Kolski, ISBN 3-86611-283-1, pp. 704, February 2007, Plv/ARS, Germany

step~~,~~ a global path over an existing map is calculated, and in the second step~~,~~ the robot reads the sensor data and modifies the local trajectory in a reactive way. This provides the path adaptation mechanism to cope with unexpected obstacles or dynamic environments. These methods calculate the global trajectory off-line in a priori known map, and while the robot is moving local modifications are made continuously based on the sensor data.

The reason for using a two level planning approach is due to high computational cost that is required in most motion planning techniques to achieve an updated environment model (and to plan a smooth and dynamically adapted trajectory).

The use of a two level planner strategy decreases computational cost by activating the global planner occasionally (it can be done off line) while the local planner which is much faster runs on line. This two level approach affects the control architecture of the mobile robot. The first mobile robots were based on a sense-model-plan scheme referred to in the literature as planned architectures (Meystel, 1986). These architectures present some difficulties providing fast responses to environmental changes.

Posterior reactive architectures (Brooks, 1986) have the advantage of using fast response methods based on a sensor-decision-action scheme to react to environmental changes, but also show the difficulty of extending the reactivity to upper levels. Finally, hybrid deliberative/reactive architectures (Arkin, 1990), (Arkin, 1998), (Alami et al., 1998), (Chatila, 1995), (Bonasso et al., 1996) and (Low et al., 2002) have emerged as the result of recognizing the advantages provided with planning at high control levels and reactive architectures at lower control levels.

The Voronoi Fast Marching method is based on a sensor-based global path planning paradigm. This is a planning approach based on a fast sense-model-plan scheme able to integrate sensor information in a simple grid based environment model and to calculate a globally consistent, smooth and safe trajectory, fast enough to be used as a reactive navigation method. This approach presents some advantages. One is the ability of global planning methodologies to guarantee a path between a given point and the goal point, if one exists. And the other is the smoothness and safety of the solution obtained. This solution eliminates the local minima trap problem and the oscillations in narrow places present in other methods, and also indirectly eliminates the use of a supervision system able to detect local minima failures (obstructed paths), in order to initiate the search for a new and feasible global path from the current position to the goal point.

To calculate the trajectory, the proposed method combines the Voronoi Distance transform and the Fast Marching Method. The Voronoi approach to path planning has the advantage of providing the safest trajectories in terms of distance to obstacles, but because its nature is purely geometric it does not achieve enough smoothness. On the other hand, the Fast Marching Method has been applied to path planning (Sethian, 1996a), and their trajectories are of minimal distance, but they are not very safe because the path is too close to obstacles, and more importantly the path is not smooth enough. In order to improve the safety of the trajectories calculated by the Fast Marching Method, two solutions are possible. 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 before the Distance Transform. The second possibility, used in this paper, is to dilate the objects and the walls in a safely distance to ensure that the robot does not collide 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, the path obtained verifies the smoothness and safety required for mobile manipulator's path planning. The advantages of this method include the ease of implementation, the speed of the method, and the quality of the trajectories produced. The method works in 2D and 3D, and it can be used on a global or local scale, in this case operating with sensor information instead of using a priori map (sensor based planning).

The chapter is organized as follow, after discussing related work in Section 2, Section 3 explains the requirements and basic ideas of the method and in Sections 4, 5 and 6 the theoretical background of various ideas used in the method are discussed. Section 7 discusses some results of the method "Voroni Diagram and Extended Fast Marching" for an holonomic mobile robot. Sections 8 discuss the contributions of the method with respect to other methods and Section 9 concludes the chapter.

2. Related work

The objective of our work is to calculate collision-free trajectories for a mobile robot and manipulators operating in environments with unknown obstacles (dynamic or not). The technique proposed here avoids the classical partition in global based on a priori information (motion planning or global planning), and local based on sensory information (reactive navigation, collision avoidance or sensor based local planning).

From a theoretical point of view, the motion planning problem is well understood and formulated, and there is a set of classical solutions able to compute a geometrical trajectory avoiding all known obstacles. Mobile robot path planning approaches can be divided into five classes according to Latombe (Latombe, 1991). 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 pre-defined shape (e.g. rectangles) and not exactly cover free space. Potential field methods differ from the other four in such a way that they consider the robot as a point evolving under the influence of forces that attract it to the goal while pushing it away from obstacles. Navigation functions are commonly considered as special case of potential fields. Most of these general methods are not applicable if the environment is dynamic or there are no modelled obstacles. Hence, to avoid the problem of executing an unrealistic geometrical trajectory which can collide with obstacles, obstacle avoidance algorithms have been developed to provide a robust way of coping with this problem.

These methods are based on a perception-action process that is repeated periodically at a high rate. The process has two steps: first, the sensory information is collected and then a motion command is calculated to avoid collisions while moving the robot toward a given goal location (that is while maintaining the global plan previously determined)

Due to fast operation rate of the collision avoidance algorithms they can deal robustly with unknown obstacles and dynamic scenarios. However, it is difficult to obtain optimal solutions and to avoid trap situations since they use only local sensory information. This classical approach to motion planning is demonstrated schematically in figure 1. Next, we describe some of these related methods.

One of the classical methods for dynamically coping with the collision avoidance problem is the potential field approach developed by Khatib (Khatib, 1986 and Khatib & Chatila, 1995). This approach is based on the creation of an artificial potential field in which the target is an attractive pole and the obstacles are repulsive surfaces. The robot follows the gradient of this potential toward its minimum. The derived force induces a collinear and proportional acceleration enabling easy dynamics and kinematic control actions. This technique can be used at a global or local level depending on the information used to generate the potentials. The major advantage of this method is its simplicity, and the capability of being used dynamically due to easy treatment of fixed and mobile obstacles. Its major disadvantage is the possible existence of local minima and the oscillations for certain configurations of obstacles. Despite of such problem this technique has been used extensively in reactive architectures because of its ability to encapsulate specific robot behaviours.

A variation of this idea is the vector field histogram (VFH) method (Borenstein & Koren, 1991). VHF uses a method consisting of a local heuristic choice based on a bi-dimensional grid modelling of the environment. A polar histogram is constructed around the robot representing the polar densities of the obstacles. The sector associated with the least density closest to the goal direction is chosen. This technique is simple, but may lead to oscillations and can also be trapped in certain configurations. Posteriorly, several improvements have been proposed in the VFH+ to improve the security distance (object enlarging), to reduce oscillations between valleys (hysteresis), and to obtain path smoothness (Ulrich & Borenstein, 1998). A third version, VHF* has been proposed that includes a "look-ahead" verification of the robots motions in order to avoid local traps (Ulrich & Borenstein, 2000). A projected position tree is built by predicting each possible movement of the robot and searches using A* classic method to choose a motor command.

Between the obstacle avoidance (sensor-based local planners) we can find the Curvature-Velocity method (Simmons, 1996), which treats obstacle avoidance as a constrained optimization in velocity space. Vehicle dynamics and obstacle information are converted into constrains and used in the optimization process.

The Dynamic Window approach (Fox et al., 1997) uses a similar approach to the Curvature-Velocity method. It also uses a constrained search in the velocity space to determine convenient speed commands. The use of a grid based representation on one hand is simple but on the other hand increases the memory requirements and the computational cost. The Nearness Diagram approach (Minguez & Montano, 2004) is a situated-activity reasoning system that based on the sensor information identifies the situation and determines the corresponding action. It uses a reactive navigation method designed by a symbolic level decision tree based on a set of complete and exclusive definitions of the possible situations.

The Elastic Bands proposed in (Quinlan & Khatib, 1993) represent connected bubbles of the free space subject to repulsive forces from obstacles and attractive forces from neighbouring bubbles. The elastic band iteratively smoothes the plan and adapts it to moving or unexpected obstacles.

The majority of the above methods have commutation difficulties, due to the fact that local avoidance algorithms provide modifications of global trajectories, but do not offer solutions when the global trajectory is blocked or trapped in local minima. The classical solution to this problem is to include a supervisor to analyze the global path execution. In the case where a local trap or a blocked trajectory is detected, it begins a search for a new global path from the current location to the goal point.

The objective of the present work is to unify the global motion planner and the obstacle avoidance planner in a single planner. This provides a smooth and reliable global motion path that avoids local obstacles from the current position to the goal destination. The motion planning structure with the proposed planner is shown in figure. 2.

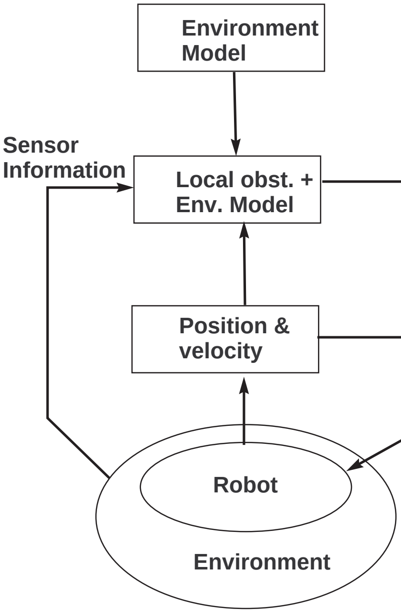

Fig. 2. Sensor-based global planning approach.

3. The sensor-based global planner algorithm

The sensor-based global planner algorithm proposed in this study is based on three key points:

- *Fast response*. The planner needs to be fast enough to be used reactively in case unexpected obstacles make it necessary to plan a new calculation. To obtain such a fast response, a fast planning algorithm and a fast-simple treatment or modelling of the sensor information is required. This requires a low complexity order algorithm to obtain a real-time response to unexpected situations.
- *Smooth trajectory*. The planner must be able to provide a smooth motion plan which can ~~to~~ be executed by the robot motion controller. This avoids the need for a local refinement of the trajectory.
- *Reliable trajectory*. The planner will provide a safe (reasonably far from a priori and detected obstacles) and reliable trajectory (free from local traps). This avoids the coordination problem between the local avoidance planners and the global planners when local traps or blocked trajectories exist in the environment.

To satisfy all these requirements the sensor based global planner approach follows these steps:

1. The first step of the algorithm integrates sensor measurements into a grid based map by updating the corresponding occupied cells. The sensory information is included in the environment map avoiding complex modelling or information fusion.
2. In the second step the objects are enlarged in the radius of the mobile robot, which will eliminate unfeasible paths~~,~~ and avoid additional path verifications.

3. The updated environment model is then converted to an environment safety image by using the Euclidean distance transform in the environment image. In the new image, each position contains the distance to the closest object that produces a distance map based on the Euclidean distance transform where maxima are obtained in the Voronoi diagram of the environment model (skeleton). This reveals the collision risk of a given trajectory in terms of the distance to obstacles at each point of the trajectory. The step starts with the calculation of the Voronoi Diagram of 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, a skeletonisation is done by using image techniques in order to obtain the Voronoi Diagram. Simultaneously, the method calculates the Extended Voronoi Transform of the map to obtain the Euclidean distances of each cell to the closest obstacle in the environment. This part of the method is also very because it is done by efficient image processing techniques.
4. Based on this distance map to obstacles, the Level Set Method (Fast Marching), is used to calculate the shortest trajectories in the potential surface defined by the Extended Voronoi Transform of the map. The calculated trajectory is geodesic in the potential surface, i.e. with a viscous distance. This viscosity is indicated by the grey level. If the Level Set Method were used directly on the environment map, we would obtain the shortest geometrical trajectory, but then the trajectory would be neither safe nor smooth.

The main reason to separate the trajectory calculation into two parts (in classical approaches) is because the global path calculation is very slow and it is impossible to re-calculate it continuously as the robot is moving. The method proposed here is extremely fast, where the whole process (sensing-model-plan) takes only 0.15 seconds (for a medium length trajectory), letting the algorithm to re-calculate a path from the current robots position to the goal location reactively.

4. Voronoi Diagram and Skeleton determination

It has been observed that the skeleton is embedded in the Voronoi diagram of a polygonal shape (Lee, 1982). Similarly, the skeleton of a shape described by a discrete sequence of boundary points can be approximated from the Voronoi diagram of the points. 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 this case, the approximation error can be quantified, and made arbitrarily closer to zero.

Consider the set *F*, closed in R^2. A Voronoi region is associated with each point in *F*.

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

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* and not contained in any other disk in *G*. Assume that all maximal disks in *G* are bounded. The skeleton σ(*G*) is the set of centres of maximal disks in *G*. One desires the skeleton to be "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* if it is morphologically open and closed with respect to a disk of radius $r > 0$ (Serra, 1982). 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.

4.1. Skeleton-based generalization algorithm

One issue that needs improvement is the existence of spurious "hairs" on the generated skeletons. This is a well-known artefact of skeleton generation, where any irregularities in the boundary generates unwanted skeleton branches. Ogniewicz (Ogniewicz & Kubler, 1995) 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 circum-circle was achieved. However, with the development of the one-step crust and skeleton algorithm this process may be greatly simplified. Blum (Blum, 1967), as well as Alt (Alt & Schwarzkopf, 1995) 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 co-circular, with their centre at the skeleton leaf. If we wish to simplify the skeleton we should retract leaf nodes to their parent node location, creating four co-circular 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 circum-circle, which smoothes outward-pointing salient in the boundary of the object. The same process is repeated from the other side of the boundary, which may displace some of the points involved in the first smoothing step, but as the process is convergent a small number of iterations suffice to produce a smoothed curve having the same number of points as the original, but with a simplified skeleton. Figure 3 shows the Voronoi diagram obtained by skeletonization of a room.

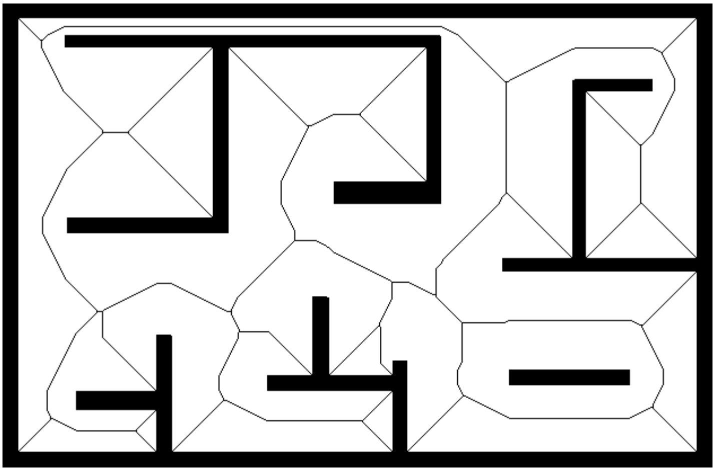

Fig. 3. Map of the room with the Voronoi made by skeletonization.

5. The Extended Voronoi Transform

A k-dimensional binary image is a function I from the elements of an $n_1 \times \ldots \times n_k$ array to [0,1]. The elements are called pixels when $n = 2$ and voxels when $n \geq 3$. Voxels of value are called background and foreground or feature voxels, respectively. For a given distance metric, the Extended Voronoi Transform (also called Image Distance Transform) of an image I is an assignment to each voxel x of the distance between x and the closest feature voxel in I, i.e. it consists of associate grey levels for each cell. As a result of this process, a kind of potential proportional to the distance to the nearest obstacles for each cell is obtained. Zero potential indicates that a given cell is part of an obstacle and maxima potential cells correspond to cells located in the Voronoi diagrams (which are the cells located equidistant to the obstacles). This function introduces a potential similar to a repulsive electric potential.

At each dimension level, the transform is computed by constructing the intersection of the Voronoi diagram whose sites are the feature voxels with each row of the image. This construct is performed efficiently by using the transform in the next lower dimension.

The algorithm calculates the Voronoi and the Extended Voronoi Transform simultaneously, therefore, saving time. The algorithm used (Maurer *et al.*, 2003) for a k-dimensional binary image is linear in the total number of voxels N, i.e. it has a $O(N)$ computational complexity. A parallel version of the algorithm runs in $O(N/p)$ time with p processors. Figure 4, shows the potential image generated by the Extended Voronoi Transform for the previous room example.

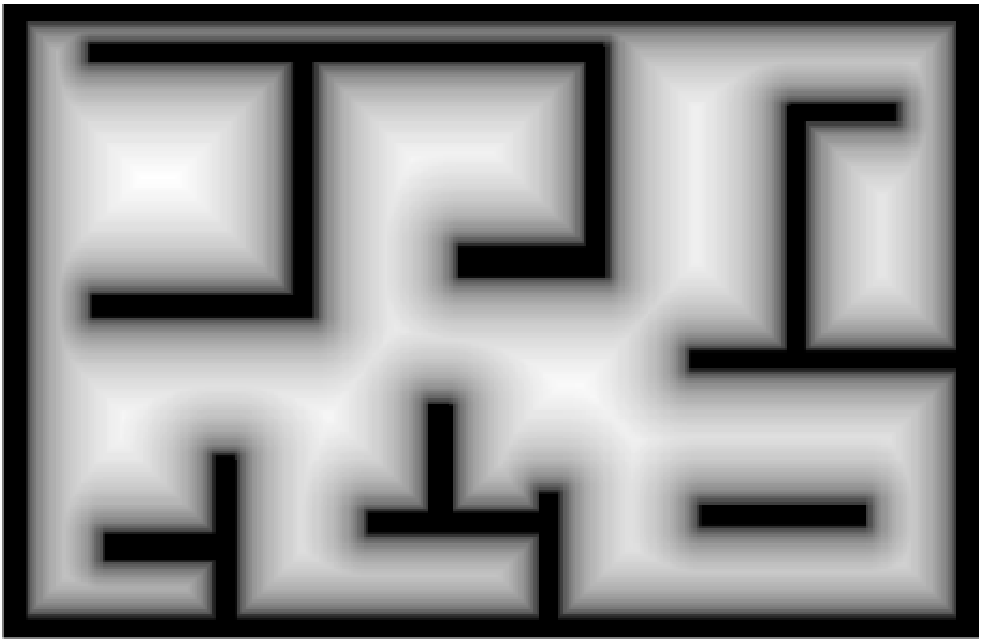

Fig. 4. Potential of the Extended Voronoi Transform.

6. Level Set Method and the Fast Marching Planning Method

The level set method was devised by Osher and Sethian (Osher & Sethian, 1988) 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 subsequent motion of the interface under a velocity field. This velocity can depend on position, time, and the geometry of the interface and on 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 involves 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, a partial differential equation must be solved 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_0. We further assume that $\phi(x;0) = \pm d(x)$; where $d(x)$ is the distance from x to the curve γ_0. We use a plus sign if x is inside γ_0 and minus if it 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

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

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

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

The particle speed *dxdt* in the direction n normal to the level set is given by the speed function *V*, and hence

$$\frac{dx}{dt} \cdot n = V. \tag{5}$$

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

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

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

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

and the initial condition

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

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) $T(x)$ defined by

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

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

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

coupled with the boundary condition

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

The advantage of formula (11) is that we can solve it numerically by the fast marching method (Sethian, 1996c), which is exactly what will be done in this study.

The direct use of the Fast Marching method does not guarantee a smooth and safe trajectory. Due to the way that the front wave is propagated and the shortest path is determined, the trajectory is not safe because it selects the shortest path, resulting in un-safe trajectory that touches the corners and walls, as is shown in figure 5. This problem can easily be reduced by enlarging the obstacles, but even in this case the trajectory tends to go close to the walls.

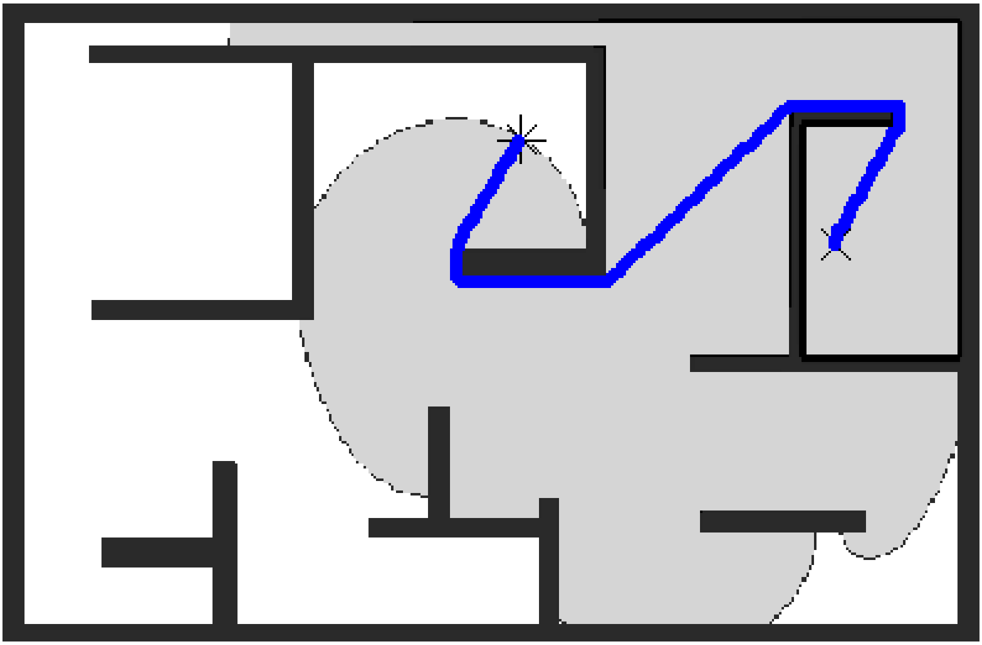

Fig. 5. Trajectory calculated by Fast Marching Method directly, without the use of the Extended Voronoi Transform.

The use of the Extended Voronoi transform together with the Fast Marching method improves the quality of the calculated trajectory considerably. On the one hand, the trajectories tend to go close to the Voronoi skeleton because of the optimal conditions of this area for robot movement; on the other hand the trajectories are also considerably smoothened. This can be observed in figure 6, where safe and smooth quality of the trajectory can be noted.

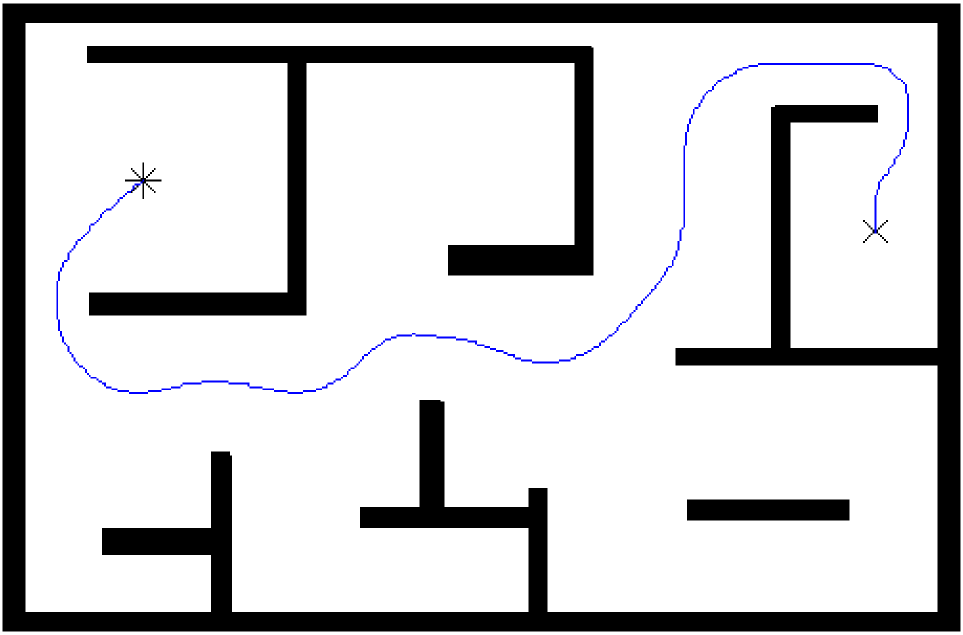

Fig. 6. Trajectory calculated using Fast Marching with the Extended Voronoi Transform.

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 $\phi(x;t)$. Depending on the form of the speed function V, the propagation of the level set function $\phi(x;t)$ 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 to 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 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 set of numerical tricks, gives it tremendous speed.

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 required to select the (physically) correct generalized solutions. The correct mathematical framework to treat Hamilton-Jacobi type equations is provided by the notion of viscosity solutions (Crandall & Lions, 1983; Crandall et al., 1992).

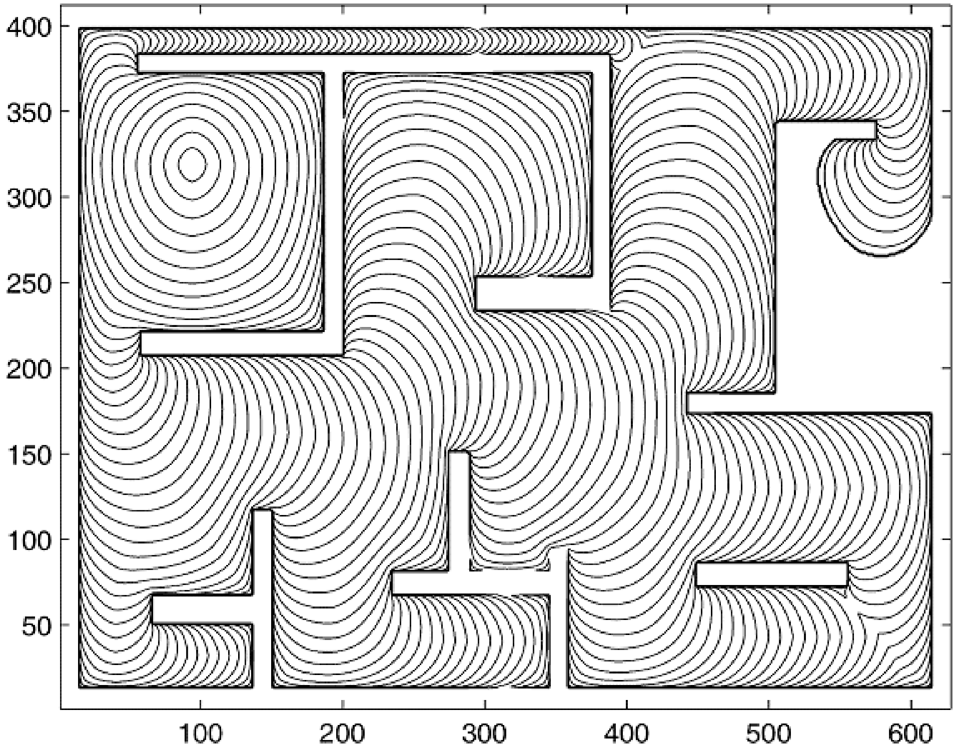

Fig. 7. Front propagation of the Fast Marching Method with the Extended Voronoi Transform.

After its introduction, the level set approach has been successfully applied to a wide array of problems that arise in geometry, mechanics, computer vision, and manufacturing processes, see (Sethian, 1996b). Numerous advances have been made in the original technique, including the adaptive narrow band methodology (Adalsteinsson & Seth, 1995) and the fast marching method for solving the static Eikonal equation ((Sethian, 1996a), (Sethian, 1996b)). For further details and summaries of level set and fast marching techniques for numerical purposes, see (Sethian, 1996b). The Fast Marching Method is an $O(n\log(n))$ algorithm.

7. Results

To illustrate the capability of the proposed method three tests are shown. In the first test, the method proposed is applied directly to the data obtained from a laser scan around the robot. Note how the method obtains a good trade off between trajectory distance, distances to obstacles and smooth changes in the trajectory (figures 8 and 9).

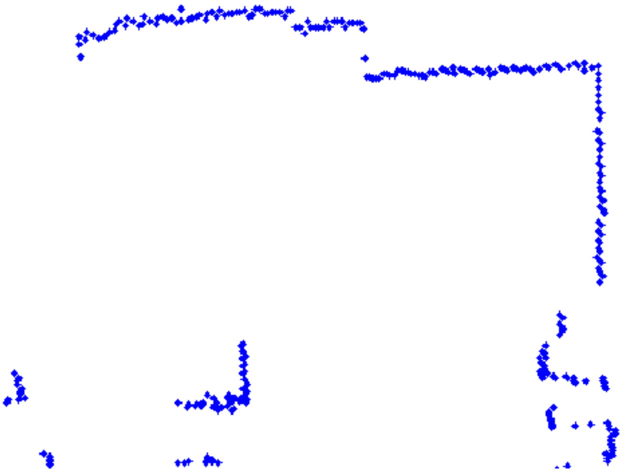

Fig. 8. Laser scan data.

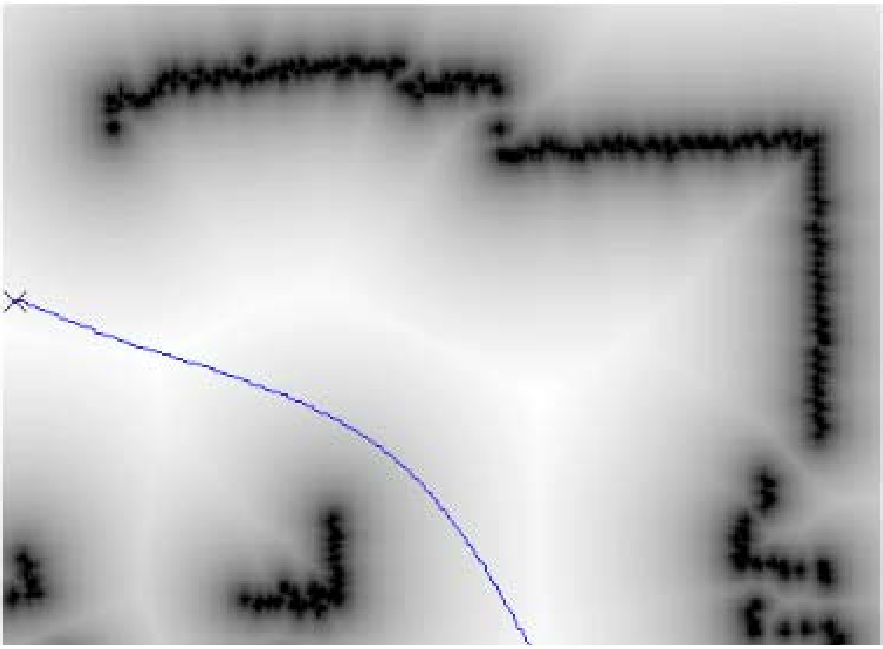

Fig. 9. Extended Voronoi transform of the scanned data and trajectory obtained with the Fast Marching Method

In the second test, in order to show global plan capabilities, the method is applied to the whole plant of the building where our laboratory is located. The laboratory plant is around 2000 square meters (medium size). The results are shown in figures 10 and 11.

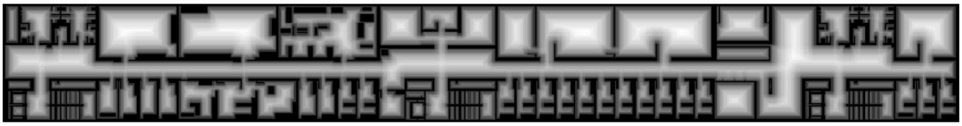

Fig. 10. Extended Voronoi transform of the UC3M Robotics Lab plan

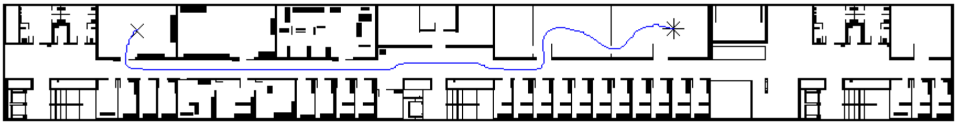

Fig. 11. Motion trajectory obtained with the Extended Fast Marching Method

The last test shows the combination of the global and local properties of this method. In this case a simple trajectory motion is determined from an initial position to the goal position. During the robot motion the robot observes the environment with its laser scan, introduces it in the map, and plans a new trajectory. Local observations (obstacles located in the middle of the corridor) that originate may slightly modify the trajectories to avoid detected obstacles (figure 12). In the last image of figure 12 the obstacles detected blocked the corridor and the sensor based global planner finds a completely different trajectory.

This technique shows the advantage by not only being local, but also global, and combines sensor based local planning capabilities with global planning capabilities to react to the obstacles rapidly, while maintaining reliability in the planned trajectory. The method always finds a solution, if it exists.

8. Discussion

We present here a discussion regarding alternative planning approaches, the limitations of the proposed method, and the potential effect on mobile robot's control architecture.

8.1. Comparison with existing methods

The common limitation of all the reactive navigation methods analyzed in this section is that they cannot guarantee global convergence to the goal location, because they use only a fraction of the information available (the local sensory information). Some researchers have worked on introducing global information into the reactive collision avoidance methods to avoid local traps situations. This approach has been adopted in (Ulrich & Borenstein, 2000) which utilizes a look-ahead verification in order to analyze the consequences of a given motion a few steps ahead avoiding trap situations. Other authors exploit the information about global environment connectivity to avoid trap situations (Minguez & Montano, 2001). Such solutions still maintain the classical two-level approach, and require additional complexity at the obstacle avoidance level to improve the reliability.

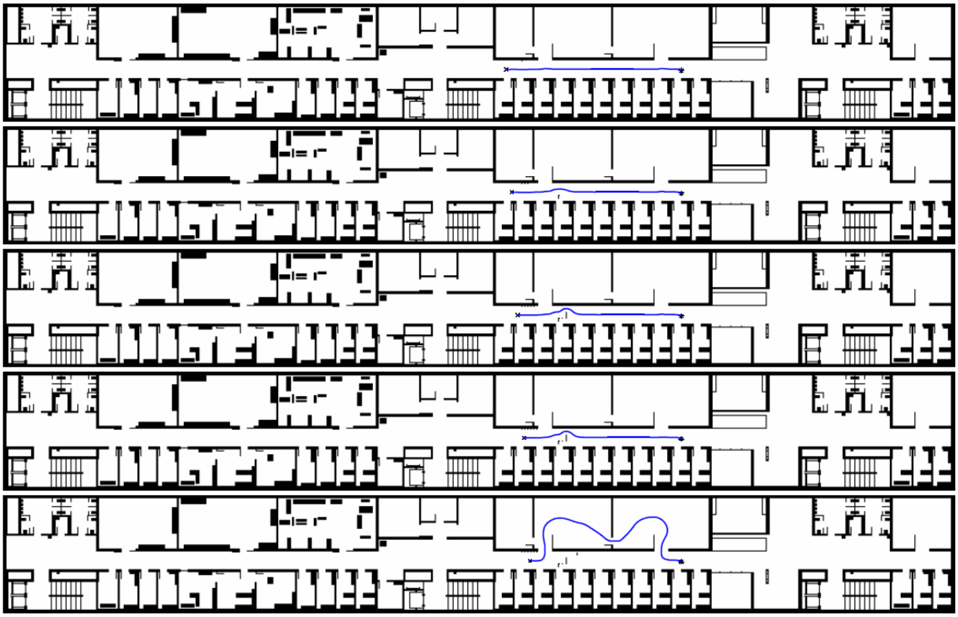

Fig. 12. Dynamical evolution of the path when the robot reads information about the new obstacles that are not in the previous map and the robot cannot pass through the corridor.

The Voronoi Fast Marching method is consistent at local and global scale, because it guarantees a motion path (if it exists), and does not require a global re-planning supervision to re-start the planning when a local trap is detected or a path is blocked. Additionally, the path calculated has good safety and smoothness characteristics.

8.2. Method limitations

The main limitation of the Voronoi Fast Marching method is its computational complexity and the computational differences between the initial and the final stages of task implementation. Because the fast marching method is a wave propagation technique, the area explored by the algorithm can be quite different. This causes the algorithm to be faster at the end of the task. Despite of its computational efficiency for medium size environments, the computational cost increases with the environment size. In our test site, the laboratory plant is about 2000 square meters (medium size) and the slowest computation time detected was 150 milliseconds (2.2 Ghz Pentium 4 processor). This time is acceptable for most applications, but because the algorithm has to operate in a reactive way in case of bigger environments, the computational cost has to be controlled. One possible solution is to increase the cell size, another is to use a multi-resolution map system which uses a big cell size to calculate the motion path in areas outside of a window located around the robot.

The method can be extended to non-holonomic robot structures by modifying the algorithm to operate in the robot's configuration space instead of the environment map. However, the computational cost increases and its capability to operate in real time decreases.

8.3. Architectural effect

Due to the fact that the VFM method integrates global motion planning and obstacle avoidance capabilities in an algorithm, which lets us simplify the mobile robot control architecture. This technique has been originally designed for mobile manipulators, which requires a smoother and safer motion plan compared to mobile robots. From an architectural point of view, this uncovers an old discussion in the mobile robot research community. Can a planned architecture be as efficient and as responsive as a reactive architecture? This difficult question we think may be answered by utilizing sensor-based global planning method that can lead us to develop efficient and responsive sensor-based planned architectures, and to also simplify the current hybrid architectures which operate deliberatively at upper architectural levels and reactively at lower levels.

9. Conclusions

We have addressed here a sensor-based global planner. We have presented a method able to deal simultaneously with global and local planning requirements using a combination of the extended Voronoi transform and the Fast Marching method. The advantage is that our design is able to provide a smooth and safe motion trajectory, and at the same time guarantee that the trajectory obtained from the current location to the goal location is free from local traps (with the information available at that moment).

To illustrate its possibilities, the method has been used for planning a trajectory in different situations: In the first case operating only with sensor information (in this case only as a local planner); in the second case working as a global planner by using a priori map only; in the third situation operating as a sensor-based global planner using a priori information and sensor information, to re-plan dynamically the trajectory from the current position to the goal position as soon as the sensor originates information changes around the robot. The dimensions of the laboratory environment are 116x14 meters (the cell resolution used in the a priori information map are 12 cm). For this environment the first step (the Extended Voronoi Transform) takes 0.05 seconds in a Pentium 4 at 2.2 Ghz, and the second step (Fast Marching) takes between 0.05 and 0.15 seconds, depending on the length of the trajectory. For a medium distance trajectory on the map, the value is around 0.10 seconds.

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 Fast Marching complexity is $O(n\log(n))$ and the Extended Voronoi Transform complexity is $O((n))$, where n is the number of cells in the environment map. The main contribution of the method is that it robustly achieves smooth and safe motion plans during dynamic scenarios in real time that can be used at low control levels without any additional smoothing interpolations. This permits the method to fuse collision avoidance and global planning in only one module, which can simplify the control architecture of the mobile robots and manipulators.

10. Acknowledgment

The authors would like to thank the Spanish Government for the funds provided by the MICYT project DPI2004-00594.

11. References

Adalsteinsson, D. &. Sethian, J.A. (1995). A fast level set method for propagating interfaces, *J. Computational Physics*, Vol. 11, N^{o} 2, pp. 269–277, 1995. ISSN: 0021-9991

Alami, R. ; Chatila, R.; Fleury, S. ; Ghallab M. & Ingrand, F. (1998). An Architecture for Autonomy, *Int. Journal of Robotics Research,* Vol. 17(4), pp. 315–337, 1998, pp. 315–337, ISSN: 0278-3649

Alt, H. & Schwarzkopf, O. (1995). The Voronoi diagram of curved objects, *Proc. 11th Annu. ACM Sympos. Comput. Geom.*, pp. 89–97, 1995

Arkin, R. C. (1998). *Behaviour-Based Robotics*, MIT Press, Cambridge, MA

Arkin, R. C. (1990). Integrating Behavioral, Perceptual and World Knowledge in Reactive Navigation, *Robotics and Autonomous Systems*, vol. 6, N^{o}12, June 1990, pp. 105–122, ISSN: 0921-8890.

Blum, H. (1967). A transformation for extracting new descriptors of shape, *Models for Perception of Speech and Visual Form*, MIT Press, pp. 153–171, 1967.

Borenstein, J. & Koren, Y. (1991). The vector field histogram - fast obstacle avoidance for mobile robots, *IEEE Trans. on Robotics and Automation*, Vol. 3, N^{o} 7, June 1991, pp. 78–288, ISSN: 1042-296X

Bonasso, R.P.; Kortenkamp, D. ; Miller, D.P. & Slack, M. (1996). Experiences with Architecture for Intelligent, Reactive Agents, *Proc. IJCAI'95 Workshop (ATAL), Lecture Notes in Computer Science 1037*, pp. 187–202, ISBN: 3 540 60805 2

Brooks, R. (1986). A Robust Layered control System for a Mobile Robot, *IEEE Trans. on Robotics and Automation*, Vol. 2(1), pp. 14–23, ISSN: 1042-296X

Chatila, R. (1995). Deliberation and reactivity in autonomous mobile robot, *Journal of Robotics and Autonomous Systems*, Vol. 16, N^{o}2-4, Dec. 1995, pp. 197–211, ISSN: 0921-8890

Crandall, M.G. & Lions, P.L. (1983). Viscosity solutions of Hamilton Jacobi equations, *Trans. Amer. Math. Soc.*, Vol. 277, pp. 1–42, 1983

Crandall, M.G.; Ishii, H. & Lions, P.L. (1992). User’s guide to viscosity solutions of second order partial differential equations, *Bull. Amer. Math. Soc.*, Vol. 27, N^{o}. 1, pp. 1–67, 1992

Fox, D.; Burgard, W. & Thrun, S. (1997). The dynamic window approach to collision avoidance, *IEEE Robotics & Automation Magazine*, Vol. 4, Mar. 1997, pp. 23–33, ISSN: 1070-9932

Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots, *Int. J. Robot. Res.*, vol. 5, spring 1986, pp. 90–98, ISSN: 0278-3649

Khatib, M & Chatila, R. (1995). An extended potential field approach for mobile robot sensor-based motions, *Proceedings of the Int. Conf. on Intelligent Autonomous Systems.* pp. 490-496, Karlsruhe Germany, 27-30 March 1995, IOS Press, Amsterdam

Latombe, J.C. (1991). Robot motion planning, *Kluwer Academic Publishers*, 1991

Lee, D.T. (1982). Medial axis transformation of a planar shape, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 4, N^{o} 4, Jul 1982, pp. 363–369, ISSN: 0162-8828

Low, K. H. ; K. Leow, W. & Ang, M. H. (2002). A Hybrid Mobile Robot Architecture with Integrated Planning and Control, *Proceedings of the 1st International Joint Conference on: Autonomous Agents and Multiagent Systems*, N^{o} 2, pp 219-226, Bologna-Italy, July 2002, ACM press NY

Maurer, C. R., Qi, R. & Raghavan, V. (2003). A Linear Time Algorithm for Computing Exact Euclidean Distance Transforms of Binary Images in Arbitrary Dimensions, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25 , N° 2, Feb. 2003, pp. 265 – 270, ISSN:0162-8828

Meystel, A. (1986). Planning in a Hierarchical Nested Autonomous Control System, *Mobile Robots, SPIE Proc.*, Vol. 727, pp. 42–76

Minguez, J. & Montano, L. (2001). Global Nearness Diagram navigation, *Proc. IEEE Int. Conf. on Robotics and Automation,* Vol. 1, pp.33–39, ISSN: 1050-4729, Seoul- Korea, May 2001, IEEE NJ

Minguez, J. & Montano, L. (2004). Nearness Diagram navigation: collision avoidance in troublesome scenarios, *IEEE Trans. Robot. and Automat.*, Vol. 20, Feb. 2004, pp. 45–59, ISSN: 1042-296X

Ogniewicz, R.L. & Kubler, O. (1995). Hierarchic Voronoi skeletons, *Pattern Recognition*, Vol. 28, N° 3, Mar 1995, p 343-359, ISSN: 0031-3203

Osher, S. & Sethian, J.A. (1988). Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations, *Journal of Computational Physics*, Vol. 79, pp. 12–49, 1988.

Quinlan, S. & Khatib, O. (1993). Elastic bands: connecting path planning and control, *Proc. IEEE Int. Conf. on Robotics and Automation,* Vol. 2, pp. 802-807, ISBN: 0-8186-3452-9, Atlanta-USA, May 1993, IEEE NJ

Serra, J. (1982). Image Analysis and Mathematical Morphology, *Academic Press*, 1982

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

Sethian, J.A. (1996b). Theory, algorithms, and applications of level set methods for propagating interfaces, *Acta numerica*, Cambridge Univ. Press, pp. 309–395, 1996.

Sethian, J.A.(1996c). A fast marching level set method for monotonically advancing fronts, *Proc. Nat. Acad Sci. U.S.A.*, Vol. 93, N° 4, pp. 1591–1595, 1996.

Simmons, R. (1996). The curvature-velocity method for local obstacle avoidance, *Proc. IEEE Int. Conf. on Robotics and Automation*, Vol. 4 pp. 3375–3382, Minneapolis, USA,. Apr. 1996

Ulrich, I. & Borenstein, J. (1998). VFH+: reliable obstacle avoidance for fast mobile robots, *Proc. IEEE Int. Conf. on Robotics and Automation*, Vol. 2, pp. 1572–1577, Leuven, Belgium, May 1998.

Ulrich, I. & Borenstein J. (2000). VFH*: local obstacle avoidance with look-ahead verification," in *Proc. IEEE Int. Conf. on Robotics and Automation*, 2000. Vol. 3, pp 2505-2511, San Francisco, USA, Apr. 2000.