Motion planning using Fast Marching Squared method

S. Garrido, L. Moreno and Javier V. Gómez

Abstract Robotic motion planning have been, and still is, a very intense research field. Many problems have been already solved and even real-time, optimal motion planning algorithms have been proposed and successfully tested in real-world scenarios. However, other problems are not satisfactory solved yet and also new motion planning subproblems are appearing. In this chapter we detail our proposed solution for two of these problems with the same underlying method: non-holonomic planning and outdoor motion planning. The first is characterized by the fact that many vehicles cannot move in any direction at any time (car-like robots). Therefore, kinematic constrains need to be taken into account when planning a new path. Outoor motion planning focuses on the problem that has to be faced when a robot is going to work in scenarios with non-flat ground, with different floor types (grass, sand, etc). In this case the path computed should take into account the capabilities of the robot to properly model the environment. In order to solve these problems we are using the Fast Marching Square method, which has proved to be robust and efficient in the recent past when applied to other robot motion planning subproblems.

1 Introduction

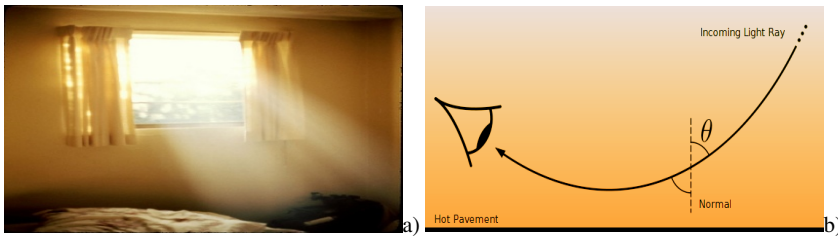
In nature, there are many fields that can be used as attractive fields. For example, the electromagnetic fields. We can guide us following the gradient of the field produced by an antenna and reach its position. In nature, there are also many fields that can be used as repulsive fields, for example the field produced by a system of particles with the same electrical charge of the charge placed in the position of interest.

The problem with the potential fields is how to joint the two fields. Historically, different researchers have tried with different mathematical operations, but in this

S. Garrido, L. Moreno and J.V. Gómez
Carlos III University of Madrid, Spain, e-mail: sgarrido, moreno, jvgomez@ing.uc3m.es

way, the total field has local minima that make the field unusable to find a path from the initial point to the goal.

Instead of using mathematical operations to joint the two fields, the Fast Marching Squared (FM^2) method propose to joint the fields as Nature does. The bees and other insects that use the light to guide themselves can go out from a semidarkness room to the sunny exterior guiding themselves to the lighter zones as in Fig. 1a). In the same way, if we consider the light ray trough a system of lenses, it goes by the path that consumes a minimum time.



a) b)

Fig. 1 a) In a semidarkness room a bee goes to the lighter zones to go outside. b) In the pavement mirage the sun rays bend near the hot road and go to the driver's eyes

These considerations lead us to think that the solution is to use the repulsive field as the refraction index of space in which a light wave is propagated. In this way, the intuition about the propagation of the light in a non homogeneous media can guide us. In the typical road mirage, as you drive down the roadway, there appears to be a puddle of water on the road several metres in front of the car. The light rays coming from the sun are twisted in the vicinity of the hot road, due to the different refractive indices of the different layers of air parallels to the road, and make the beam reaches the driver's eyes as in Fig. 1b).

Mathematically, the propagation of the light is given by the Eikonal equation that is equivalent to the Fermat's principle: Light traveling through some substance has a speed which is determined by the substance. The actual path taken by light between any two points, in any combination of substances, is always the path of least time that can be traveled at the required speeds.

The Fermat's principle is especially interesting in our application, because if we have only a source of light waves, each point is connected with the source with a light path that it is parameterised by the time. The set of all the points of the domain with the time as last coordinate in the case of two spatial coordinates, gives us a Lyapunov surface in which the level curves are isochronals and the Fermats paths are orthogonal to them. It is impossible for the method to have local minima, because if there exists a local minimum x, and the time between other point y and x is minimum, as the time between x and the origin is minimum then the minimum time trajectory between y and the origin passes trough x, i.e. the point x is not a local minimum.

In summary, the proposed method consists in the construction of a repulsive field by propagating a wave from the obstacles and walls. This gives us a refraction index or a slowness potential proportional to the inverse of the propagation velocity of the wave in the medium. Using this first potential as refraction index, a wave is propagated from the goal point. This results in the Lyapunov surface of the second potential. Applying gradient descent, maximum slope paths are obtained. Therefore these are the minimum time trajectories. The level sets of this second potential are, by definition isochronals, i.e. its points are at the same time from the origin.

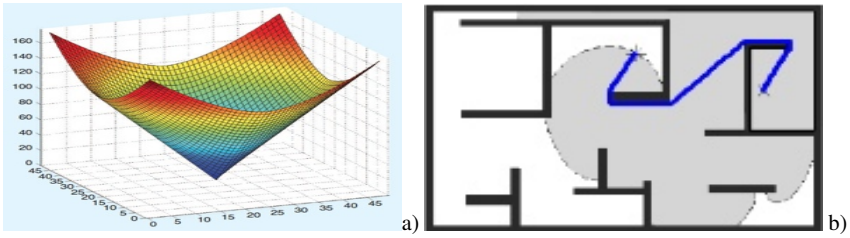


Fig. 2 Lyapunov surface when the propagation wave starts in a point and the refraction index is constant (a) and Fast Marching Path when the refraction index is constant (b)

In Fig. 2 are shown the funnel potential of the light wave propagation with a constant refraction index and a path obtained in that way.

In Fig. 3a) is shown the repulsive potential built by propagating the wave from obstacles and walls (first potential). It is similar to the distance transform, but in this case is continuous, not discrete. In Fig. 3b) is shown the fronts of the propagation of the second wave, and the corresponding path.

As the first potential can be interpreted as a difficulty map, and each point of the previous trajectory has associated a value of grey or difficulty, we can use it as velocity profile with a maximum velocity given by the white color and zero velocity given by the black color, as shown in Fig 3a). In Fig 4b) is shown the second potential (or funnel potential) of Fig.3, in which the third axis represents time.

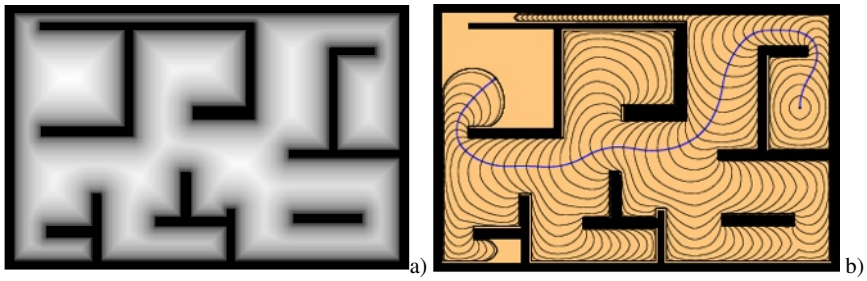


Fig. 3 a) Repulsive field by propagating a wave from the obstacles and walls (first potential) and b) the corresponding wave fronts and its Fast Marching Path when the refraction index is given by a)

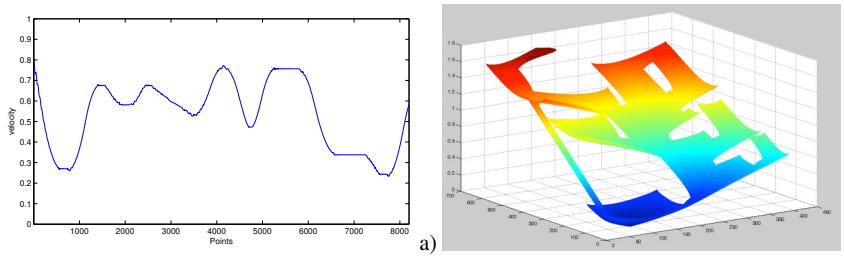


Fig. 4 a) Relative velocity profile of the path obtained in the previous figure, where 1 is the maximum velocity and b) its Lyapunov surface.

In Fig. 5 is shown an example of all the process: map of the environment, first potential and three moments of the wave expansion c) d) and e). In the last is shown the corresponding path.

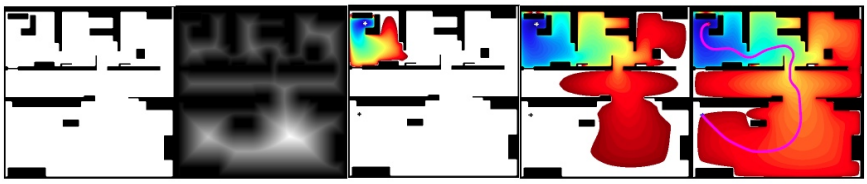
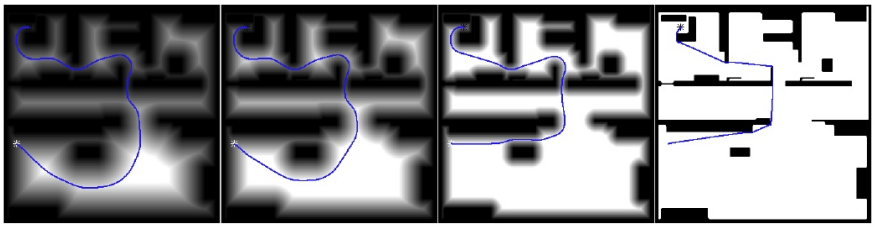


Fig. 5 Example of environment a), first potential b) and tree moments of the wave exponsion c) d) and e). In the last is shown the corresponding path.

It is possible to stop the calculation of the first potential at different distances, or different levels of saturation. In this way, it is possible to have different kinds of shapes of the path: more or less close to the walls and obstacles. In Fig. 6 are shown the paths that correspond to different levels of saturation. In Figs. 7, 8 and 9 are shown the first potential, second potential and the path corresponding to a three different levels of saturation for an environment data taken by the robots laser.

Fortunately, there exist a good method to solve the light propagation (Eikonal) equation numerically. This method is the Fast Marching (FM) method and was developed by Sethian [1]. It is an efficient computational numerical algorithm for tracking and modelling the motion of a physical wave interface (front) without reflexions. This method has been applied to different research fields, including computer graphics, medical imaging, computational fluid dynamics, image processing, computation of trajectories, etc. [2, 3, 4].

The computational efficiency of the method allows the planner to operate at high rate sensor frequencies [7], [14]. For small and medium scale environments, the proposed method avoids the need for a collision avoidance algorithms plus a global motion planner. This enables simplification of the mobile robot or mobile manipu-



(a) Saturation: 0.75 (b) Saturation: 0.5 (c) Saturation: 0.25 (d) Saturation: 0

Fig. 6 First potential and the paths that correspond to different levels of saturation.

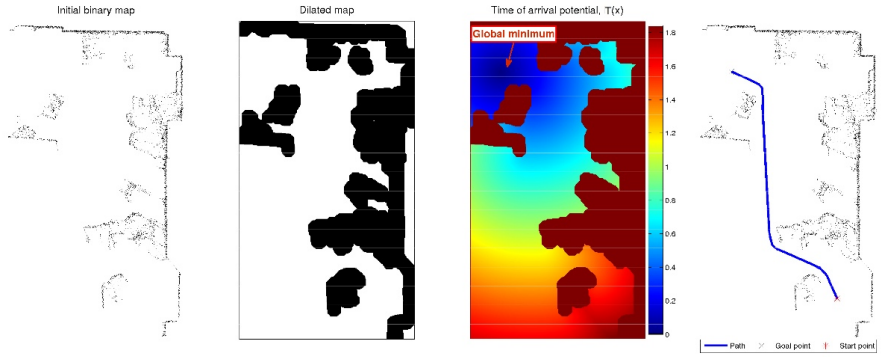


Fig. 7 Laser data of our Lab (original map), First potential, second potential and the path corresponding to a high level of saturation.

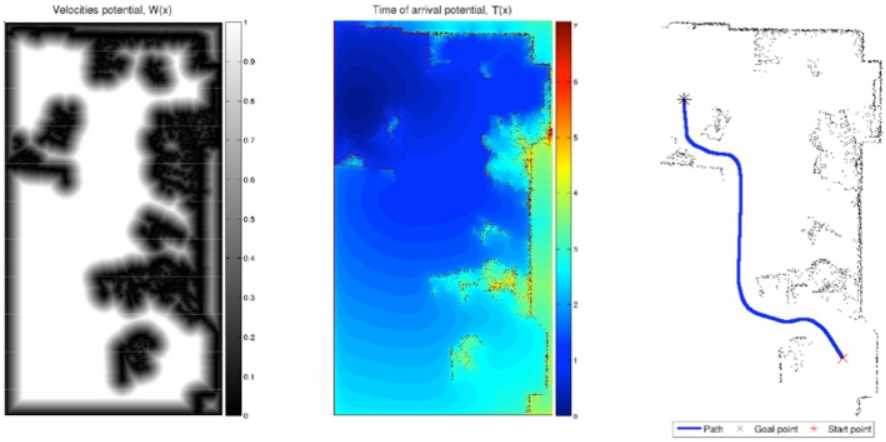


Fig. 8 First potential, second potential and the path corresponding to a medium level of saturation.

lator architectures, while maintaining good time response, smooth and safe planned trajectories with continuous curvature. The trajectory generated by the planner is

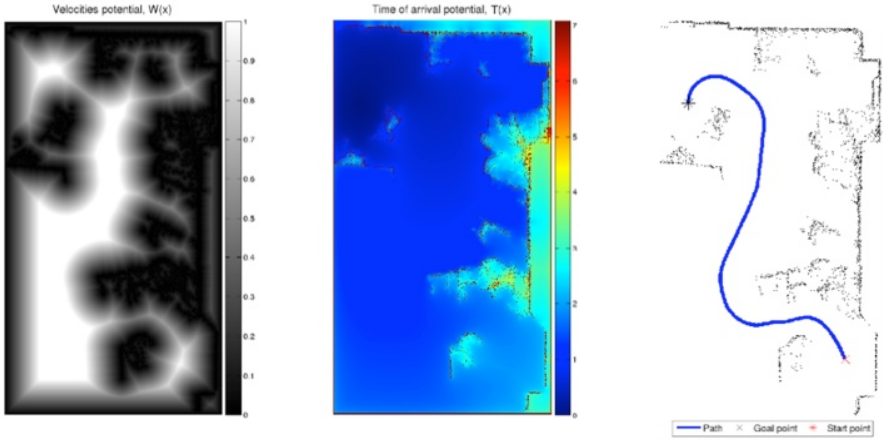


Fig. 9 First potential, second potential and the path corresponding to a low level of saturation.

the fastest possible to achieve the goal position, by implication the best path according to the maximum acceptable velocity at each point in the trajectory (path plus velocity).

2 The Eikonal Equation and the Fast Marching Planning Method

One way to characterise the position of a front in expansion is to compute the time of arrival T, in which the front reaches each point of the underlying mathematical space of the interface. It is evident that for one dimension we can obtain the equation for the arrival function T in an easy way, simply considering the fact that the distance θ is the product of the speed F and the time T.

$$\theta = F \cdot T \tag{1}$$

The spatial derivative of the solution function becomes the gradient

$$1 = F\frac{dT}{d\theta} \tag{2}$$

and therefore the magnitude of the gradient of the arrival function $T(\theta)$ is inversely proportional to the speed.

$$\frac{1}{F} = |\nabla T| \tag{3}$$

For multiple dimensions, the same concept is valid because the gradient is orthogonal to the level sets of the arrival function $T(\theta)$. In this way, we can characterise the movement of the front as the solution of a boundary conditions problem. If speed F depends only on the position, then the equation (3) can be reformulated as the eikonal equation:

$$|\nabla T| F = 1. \tag{4}$$

The Fast Marching Method is a numerical algorithm for solving the Eikonal equation, originally, on a rectangular orthogonal mesh introduced by Sethian in 1996 [1]. The Fast Marching Method is an $O(n)$ algorithm as has been demonstrated by [5], where n is the total number of grid points. The scheme relies on an upwind finite difference approximation to the gradient and a resulting causality relationship that lends itself to a Dijkstra-like programming approach.

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 (there are no reflections, interferences or diffractions). 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, in addition to a whole set of numerical tricks, gives it its tremendous speed.

Since its introduction, the Fast Marching Method approach has been successfully applied to a wide array of problems that arise in geometry, mechanics, computer vision, and manufacturing processes, see [7] for details. Numerous advances have been made to the original technique, including the adaptive narrow band methodology [8] and the Fast Marching Method for solving the static eikonal equation ([6],[7]). For further details and summaries of level set and fast marching techniques for numerical purposes, see [7].

2.1 Properties

The proposed FM^2 algorithm [11],[12],[13],[14],[15], [16] has the following key properties:

- *Fast response*. The planner needs to be fast enough to be used reactively in case unexpected obstacles make it necessary to plan a new trajectory. To obtain this fast response, a fast planning algorithm and fast and simple treatment of the sensor information is necessary. This requires a low complexity order algorithm for a real time response to unexpected situations.
- *Smooth trajectories*. The planner must be able to provide a smooth motion plan which can be executed by the robot motion controller. In other words, the plan does not need to be refined, avoiding the need for a local refinement of the trajectory. The solution of the eikonal equation used in the proposed method is given by the solution of the wave equation:

$$\phi = \phi_0 e^{ik_0(\eta x - c_0 t)}$$

As this solution is an exponential, if the potential $\eta(x)$ is $\mathscr{C}^\infty$ then the potential ϕ is also $\mathscr{C}^\infty$ and therefore the trajectories calculated by the gradient method over this potential would be of the same class. At least from a theoretical point of view, because the equation is solved numerically and the result is an approximation of that trajectory.

This smoothness property can be observed in Fig. 3, where trajectory is clearly good, safe and smooth. One advantage of the method is that it not only generates the optimum path, but also the velocity of the robot at each point of the path. The velocity reaches its highest values in the light areas and minimum values in the greyer zones. The FM^2 Method simultaneously provides the path and maximum allowable velocity for a mobile robot between the current location and the goal.

- *Reliable trajectories*. The proposed planner provides 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 collision avoidance controllers and the global planners, when local traps or blocked trajectories exist in the environment. This is due to the refraction index, which causes higher velocities far from obstacles.
- *Completeness.* As the method consists of the propagation of a wave, if there is a path from the initial position to the objective, the method is capable of finding it.

2.2 Algorithm Implementation on an orthogonal mesh

The Fast Marching Method applies to phenomena that can be described as a wave front propagating normal to itself with a speed function $F = F(i, j)$. The main idea is to methodically construct the solution using only upwind values (the so called entropy condition). Let T(i,j) be the solution surface $T(i, j)$ at which the curve crosses the point (i, j), then it satisfies $|\nabla T|F = 1$, the Eikonal equation.

In order to understand how fast marching works, imagine an imprudent visitor that leaves unextinguished fire at some location in a natural reserve. The flame quickly becomes a forest fire, which expands outwards. Fire consumes the reached trees so the fire always propagates forward. We can record the fire front position at different points in time. It appears that the fire traverses the route having the smallest propagation time (and hence, the shortest length if the velocity is constant). In optics and acoustics this fact is known as Fermats principle or, in a more general form, the least action principle. In plain language, Fermats principle states that light traveling between two points always chooses the quickest path. Snells law of refraction follows directly from this principle.

It is necessary to know that the propagation happens from smaller to bigger values of T. The algorithm classifies the points of the mesh into three sets: black, red and green, because our interface propagates like of a forest fire. Black points are points where the arrival time has been computed and is not going to change in the

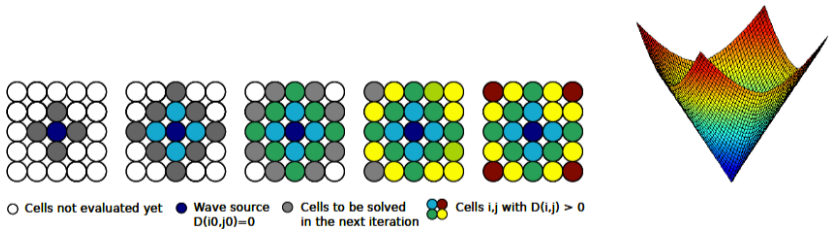
future. Green points are points that haven't been processed yet, for which the arrival time have not been computed up to now (corresponding with live trees). Red points are those belonging to the propagating wave front, which can be considered as an interface between the black and the green regions of the triangular mesh. In our forest fire example, red points correspond with trees that are currently in flames. Initially, only the source x_0 is marked as black and all points adjacent to it, are marked as red. The remaining points are marked as green. At each iteration, the red with the smallest value of T(x) is put into the black set. This T(x) value is calculated using the black points in triangles sharing it. The updated adjacent points are tagged as red. The process continues until all points become black or the goal is reached.

This equation is applied on grid points. Grid points are classified in three different types: alive, trial and far.

- Alive Points (black points) are points where values of T are known.
- Trial Points (red points) are points around the curve (alive points), where the propagation must be computed. The set of trial points is called narrow band. To compute propagation, points in the narrow band are updated to alive points, while the narrow band advances.
- Far Away Points (green points) are points where the propagation was not computed yet. During the propagation far away points are converted to trial points.

Fig. 10a) explain this idea: in the first subfigure the black point (alive) represents the initial curve; in 2nd subfigure the value of T is computed in the neighbourhood of black point; this neighbourhood is converted from far away (green) to trial points (red); in 3rd subfigure the trial point with smallest value of T is chosen (point A); in 4th subfigure the values of T are computed in the neighbours of point A, converting them from far away to trial. In fifth subfigure the trial point with smallest value of T is chosen (for example, "D"); in the last subfigure the neighbours of D are converted from far away to trial. And so on.

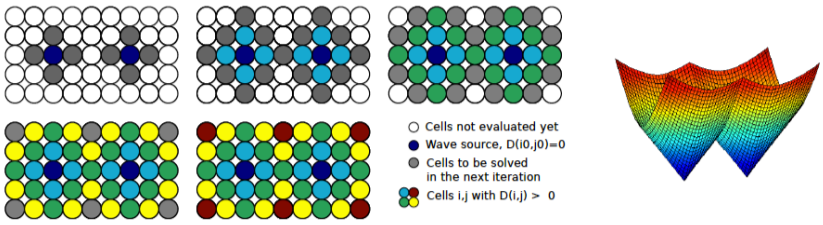
Fig. 11 represents the scheme of Fast Marching propagation with two initial points.



(a) Iterations of FM with one wave in 2D.

(b) Time of arrival potential $D(\mathbf{x})$ (third axis)

Fig. 10 Scheme of Fast Marching propagation with an initial point. Different colores (blue to red) represent different arrival times in increasing order.



(a) Iterations of FM with two propagating waves (in 2D). (b) Time of arrival potential $D(\mathbf{x})$ (third axis)

Fig. 11 Scheme of Fast Marching propagation with two initial points. Different colores (blue to red) represent different arrival times in increasing order.

2.3 Algorithm Implementation on an triangular mesh

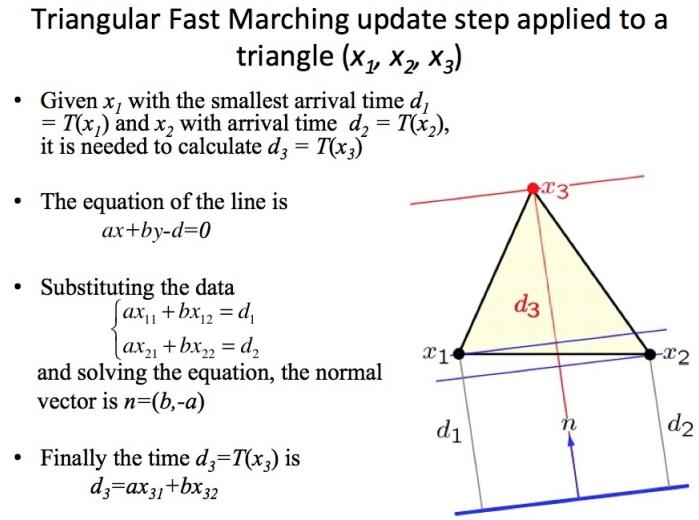


Fig. 12 Scheme of Fast Marching update step.

The numerical basis of the fast marching method and its foremost difference with Dijkstra's algorithm resides in the update procedure. While in Dijkstra's algorithm the path is restricted to the graph edges, and a graph vertex was updated each time from an adjacent vertex, in fast marching, because the path can pass through the triangular faces of the mesh, a vertex has to be updated from a triangle, requiring two supporting vertices. We assume that the update step is applied to a triangle (x_1, x_2, x_3), where x_1 is the red point with the smallest arrival time $T_1 = T(x_1)$, x_2 is a point for which some arrival time approximation $T_2 = T(x_2)$ is available, and x_3

is the red or green point, whose arrival time approximation $T_3 = T(x_3)$ is that the triangle lies in the plane with $x_3 = 0$. In essence, given that the front reaches x_1 at time T_1 and x_2 at time T_2, the update step has to estimate the time when the front arrives to x_3, as shown in Fig.12.

2.4 Results of FM^2 method

To illustrate the potential of the proposed method, four working conditions are evaluated (sensor-based operation, map-based operation, combined operations, behaviour in cluttered environment and the computational cost is shown.

2.4.1 Sensor-based planning

In the first test, the method proposed is applied directly to the data obtained from a laser scan around the robot, where the method obtains a good trade off between trajectory distance, distances to obstacles and smooth overall trajectory as shown in Fig. 13 and 14. These images correspond to a corner of a corridor of our University.

2.4.2 Map-based planning

In the second test, in order to show global plan capabilities, the method is applied to the whole plant of the building where the Robotics laboratory is located. The laboratory floor is around 2000 square meters (medium size). The results are shown in Fig. 15 and 16.

2.4.3 Combined planning

The third test shows the combination of the global and local properties of the method. In this case a simple trajectory motion is determined from an initial position to the goal position. During the motion, the robot observes the environment with its laser scan, places it on the map and plans a new trajectory. Local observations (obstacles located in the middle of the corridor) result in slightly modified trajectories to avoid the obstacles detected (Fig. 17). In the last image in Fig. 17 the detected obstacles blocked the corridor and the sensor based global planner finds a completely different trajectory. It is worth noting that in this case, the fact that the global planning capability takes action, allows automatic replanning of the trajectory. This replanning is not possible with some other methods due to the separation of the two planners.

This technique shows the advantage of a method which is not only local, but also global, that combines sensor based local planning capabilities with global plan-

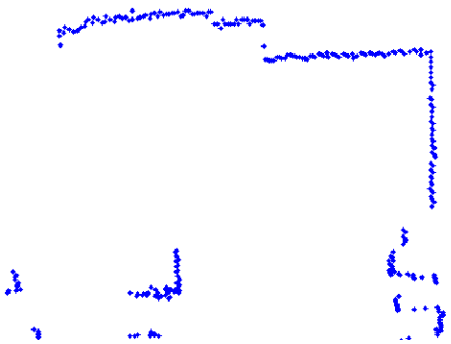


Fig. 13 Laser scan data corresponding to a corner of a corridor of our University.

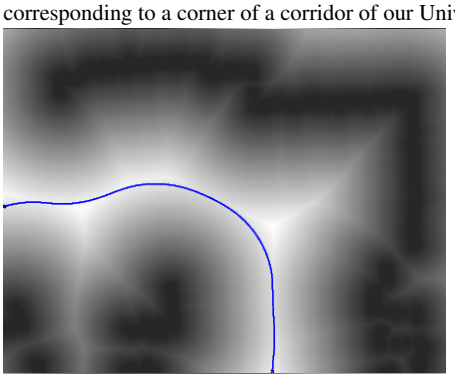


Fig. 14 Repulsive potential of the scanned data and trajectory obtained with the FM^2 Method.

Fig. 15 Slowness Potential of FM^2 1st step for the UC3M Robotics Lab floor.

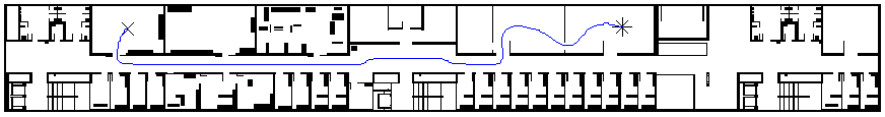


Fig. 16 Motion trajectory obtained with the FM^2 Method for the UC3M Robotics Lab floor.

ning capabilities to react quickly to the obstacles while maintaining reliability in the planned trajectory. The method always finds the solution, if one exists.

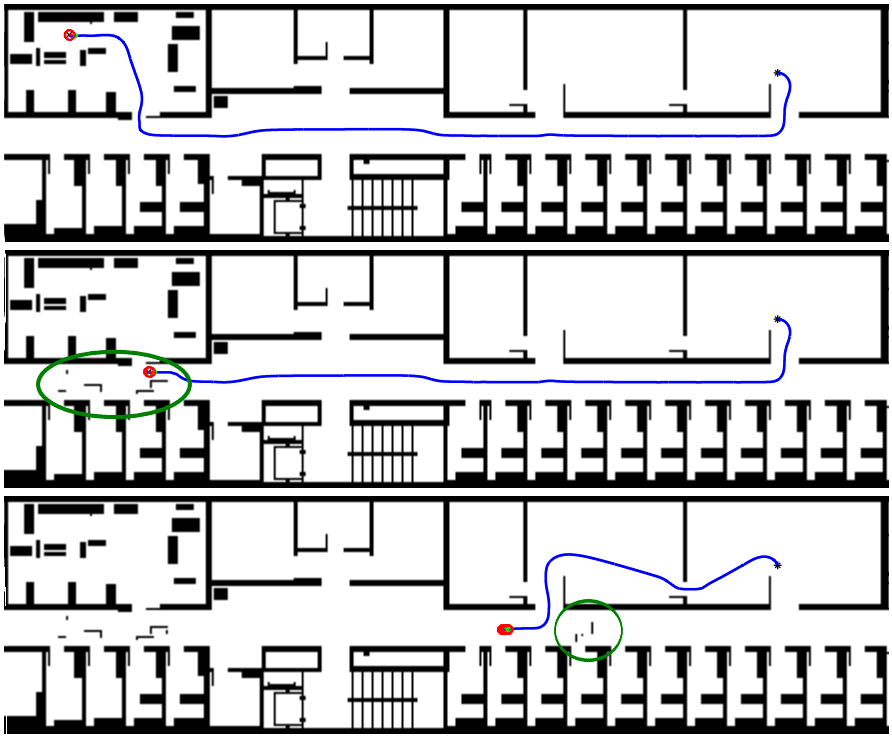


Fig. 17 Dynamical evolution of the path when the robot reads information about the new obstacles (marked with green ellipsoids) absent in the previous map and the robot cannot pass through the corridor.

3 Application of the FM^2 to car-like robots

An important kind of robots are the nonholonomic robots, that can't move freely in any desired direction, but they have to accomplish a set of constraints. A typical case are the car-like robots.

In this section, we are going to describe how to apply the FM^2 method to car-like robots. An interesting feature that has not been sufficiently highlighted in the previous sections is that by using the gradient over the second potential, it is possible to calculate a vector field whose field lines are the paths that go from each point to the target, away from obstacles and walls.

In order to apply the proposed method, it is considered a 3D C-Space of the environment, with the two dimensions of the robot's position and the vehicle's orientation as the third dimension. Computing a trajectory along the C-Space built taking into account the vehicle's dimensions, it is possible to guarantee the absence of collisions. This means we operate over the configuration space instead of the bi-dimensional environment map (see Fig. 19, in which the third dimension is the orientation of the robot, with 21 possible values. These orientations are repeated

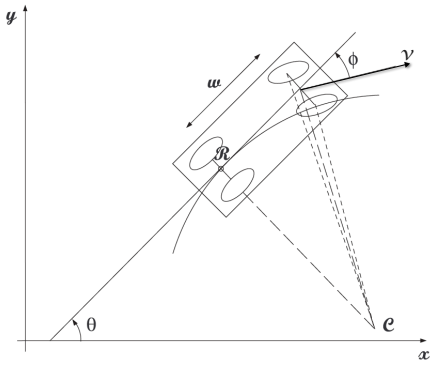


Fig. 18 A car-like robot.

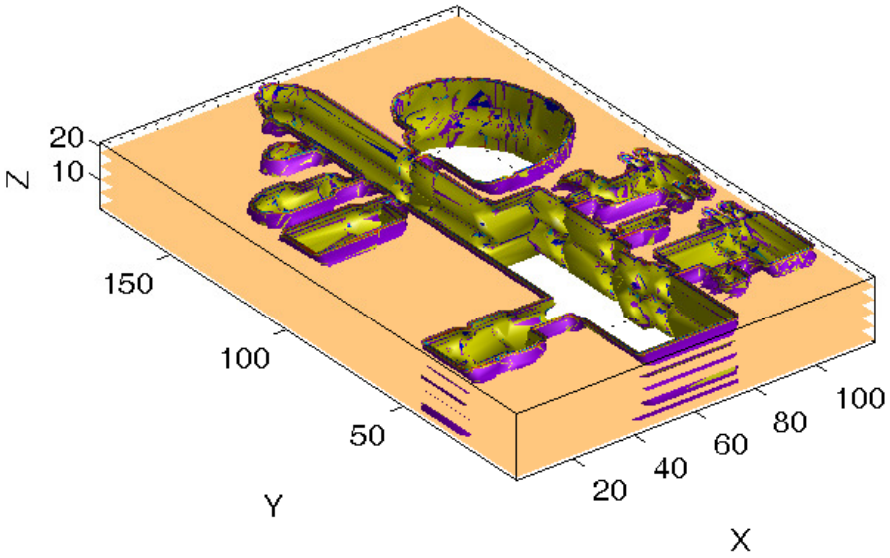


Fig. 19 Three dimensional C-space of the car-like robot, where the third dimension is the orientation. These orientations are repeated above and under the principal interval in order to permit manoeuvres

above and under the principal interval in order to permit manoeuvres). The C-space has been built iteratively placing the vehicle in every position and with every possible angle. This is a slow task, but it can be done offline and once per map.

After that, the slowness potential (distance transform) is calculated using the Fast Marching method for this resultant space. The wave is propagated from the walls of the previously calculated C-space.

Based on this slowness map, the Fast Marching Method creates the second potential $T(\mathbf{x})$ that represents arrival time of the wavefront, and in this way the method

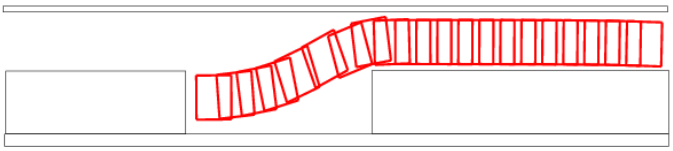


Fig. 20 Parking maneuver using FM^2-NH.

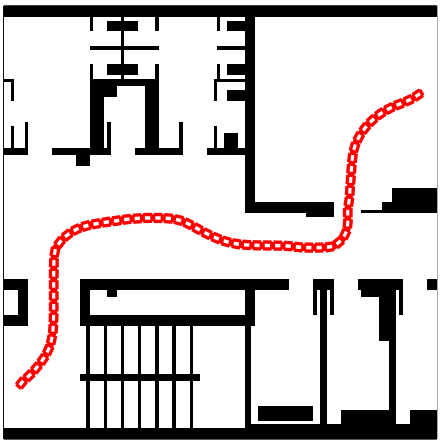


Fig. 21 FM^2-NH applied to the car-like robot in the University corridor.

gives the arrival time as the fourth axis. The origin of the wave is the goal point, which continues propagating until it reaches the current position of the robot.

Using this Funnel shaped second Potential the associated OXY vector field is calculated. This vector field has as field lines the different line paths from the different points of the C-space and all of them finish in the goal point. This lines, also, go away from the obstacles. This vector field is going to be used to move the car-like robot.

Car-like robots have a limited steering angle causing them to move along paths of bounded curvature. This can be expressed as a constraint on the curvature radius. This constraint can be directly included in the algorithm using the vector field, in form of limits during the path calculation. Fig. 20 shows the result to apply the algorithm to a parking manoeuvre.

Finally, starting from the initial position and orientation, the path is constructed step by step, according to the following order:

- The front wheels are aligned with the vector field in the midpoint of the front axis.

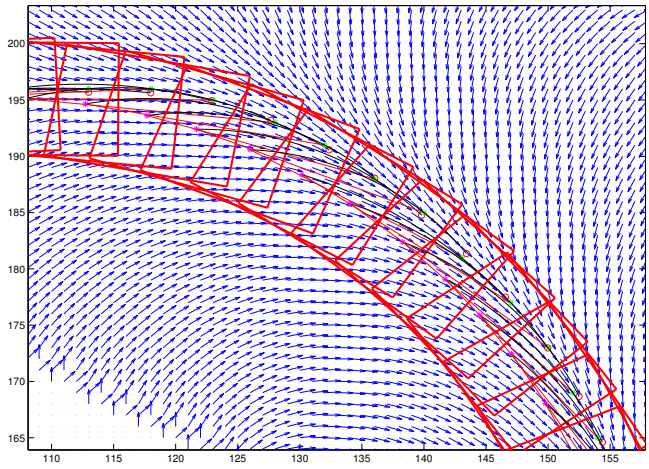


Fig. 22 Movement of the vehicle on the vector field.

- The perpendicular lines to the front and rear wheels are considered and their intersection is taken as center of the step movement.
- With the previously calculated center, the vehicle is moved a circumference arc of length proportional to the vector modulus correspondent to that point.

The previous process is repeated from the new point until the destination point is reached. The final point and orientation is always reached because the funnel potential end at this point and orientation.

Consider the car-like robot shown in Fig. 18. In this figure (x,y) is the position of the center of the rear axis, θ is the car orientation respect the OX axis. It is necessary to take into account the non-holonomic constraint

$$\dot{y}\,cos\theta - \dot{x}\,sin\theta = 0$$

and the car-like movement can be modelled, assuming the distance between the front and rear axes as 1, as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\phi} \end{pmatrix} = \begin{pmatrix} v\,\cos\phi\,\cos\theta \\ v\,\cos\phi\,\sin\theta \\ v\,\sin\phi \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} v_1 + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} v_2 \tag{5}$$

where ϕ is the front wheels orientation, v is the car velocity and v_1, v_2 are the two control inputs: acceleration of the car and angular velocity of the front wheels.

An interesting remark is that in this equation everything is done by the vector field except the control inputs v_1, v_2: the acceleration of the car and the angular velocity of the front wheels. These control inputs can be deduced and in this way the method not only give the trajectory but also the control inputs to follow that trajectory.

The result of an example of the nonholonomic version of the FM^2 method can be observed in Fig. 21, where a corridor of the university is shown. The corresponding C-space is represented in Fig. 19. The top and the bottom are connected because the angle wraps around 2π. The trajectory obtained is smooth and safe.

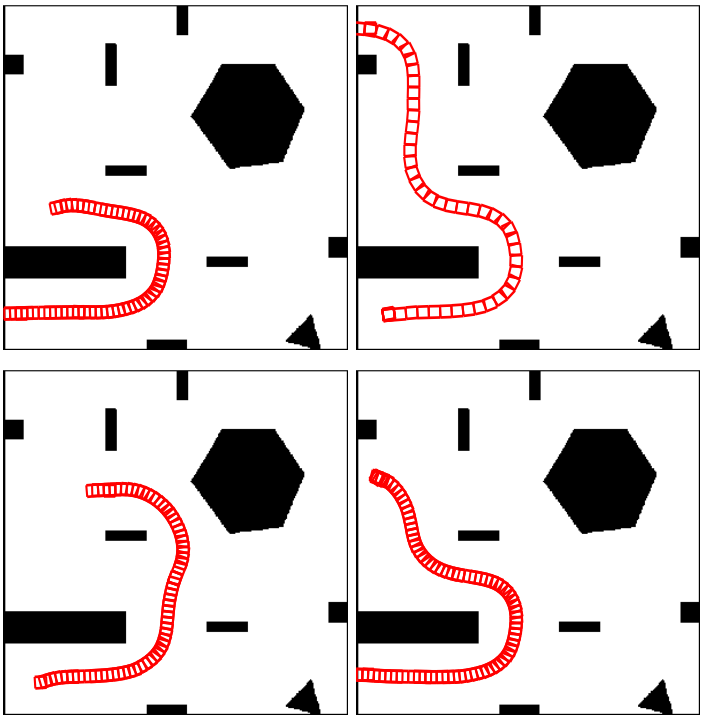


Fig. 23 Different motion trajectories obtained with the proposed method (non holonomic).

To illustrate the capability of the proposed method, different situations are shown (see Fig. 23 and 24). In the case of the Fig. 23 a simple trajectory is determined from an initial position and orientation to the goal position and orientation. Local observations (obstacles located in the scene) originate slightly modified trajectories to avoid the detected obstacles. We can conclude that the four situations have good trajectories (safe and smooth) between the initial and the final point. In the enlarged image we can see the velocity field (see the Fig. 22) calculated for the movement of the car-like robot, where the vectors have been normalised for a better visualisation. This technique shows the advantage of a method which is not only local, but also

global, which combines sensor based local planning capabilities with global planning capabilities to react to the obstacles very quickly while maintaining reliability in the planned trajectory. The proposed method is highly efficient from a computational point of view because the Fast Marching can be implemented with complexity O(n), where n is the number of cells in the environment map.

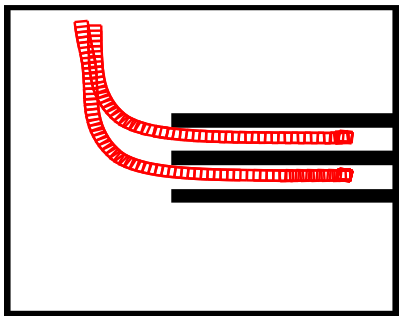


Fig. 24 Example trajectory obtained with FM^2-NH

3.1 Comparison with existing methods

The common limitation of all the reactive navigation methods 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 trap situations. This approach has been adopted by Ulrich [17] which uses a look-ahead verification to analyse the consequences of a given motion a few steps in advance to avoid trap situations. Other authors exploit the information about global environment connectivity to avoid trap situations (Minguez [18]). Those solutions still maintain the classical two level approach, and require additional complexity at obstacle avoidance level to improve the reliability at this level.

The proposed method is consistent at local and global scale because it guarantees a motion path (if it exists), and does not require global replanning supervision to restart a planning when a local trap is detected or a path is blocked. Furthermore, the path calculated has good safety and smoothness characteristics.

Most of the other methods give paths that are not smooth, even though they only provide a few loose points linked by segments of straight lines. The only methods that give comparable results are based on harmonic functions (the solutions of the equation of Laplace) but they have the problem of slowness.

4 How to deal with difficulty and uncertainty in an Outdoor Environment to Plan Trajectories using the Fast Marching method. Algorithm Implementation on a triangular mesh

This section applies the FM^2 to the problem of finding trajectories for an outdoor robot. The objective is to apply Fast Marching to a 3D triangular mesh that represents the surface terrain to find a trajectory between two points. The proposed method uses a triangular mesh because this kind of grid adapts better to 3D surfaces. The advantages of this approach are that, in the first step of the method, the algorithm calculates a weight matrix W that can represents difficulty, refraction index (inverse of speed) or uncertainty based on the information extracted from the 3D surface characteristics and the sensor data of the robot. In the experiments carried out in this work these features are the spherical variance, the gradient of the surface, the height, and also the incertitude in the map because some portions of the map can't be measured directly by the robot. This difficulty matrix is used to define the speed of propagation of the Fast Marching wave in order to find the best path depending on the task requirements, e.g., the trajectory with the fastest path, the least energy consumption, the most plain terrain, the safest path or the known terrain. The method also gives the robot's maximum admissible speed in each point. This depends on difficulty matrix. The results presented in this paper show that it is possible to model the path characteristics as desired, by varying this difficulty matrix W.

4.1 Matrix W : the difficulty map

The proposed method is based on the FM method, changing the speed of propagation of the wave using a potential generated from the 3D environment characteristics and the robot limitations. This way, the method changes the time when the front reaches each point and when the generated trajectory is calculated. This trajectory is not going to be the simple geodesic, but it is going to be modified according to the robot and task needs. To be able to modify this speed, the proposed method creates a weight matrix W, which is currently built based on the main characteristics of the 3D surface: the *spherical variance*, the *saturated gradient*, the *height* and the uncertainty. Some other characteristics can be added to the method and it will build a different potential surface.

4.1.1 Spherical variance

The spherical variance is a measure the roughness of a surface. It can determine if a zone is crossable or not. In [19], it is presented a method to calculate the roughness

degree. This method is based on the normal vector dispersion in each point of the surface:

- In a uniform terrain (low roughness), the normal vectors in a surface will be approximately parallel and, for this reason, they will present a low dispersion.
- On the other hand, in an uneven terrain (high roughness) the normal vectors will present great dispersion due great to changes in their orientation.
  The method to calculate the spherical variance is:

  1. Given a set of n normal vectors to a surface, defined by their three components $\{(x_i, y_i, z_i)\}$, the module of the sum vector $\mathbf{R}$ is calculated by:

$$R = \sqrt{\left(\sum_{i=0}^{n} x_i\right)^2 + \left(\sum_{i=0}^{n} y_i\right)^2 + \left(\sum_{i=0}^{n} z_i\right)^2} \tag{6}$$

  2. Next, the mean value is normalised by dividing the module R between the number of data n, so the value of the result is within $[0,1]$. In this way, we have

$$\frac{R}{n} \in [0,1] \tag{7}$$

  3. Finally, the spherical variance S_v is defined as the complementary of the previous result.

$$S_v = 1 - \frac{R}{n} \tag{8}$$

When $S_v = 1$, there exists a maximum dispersion that can be considered as the maximum roughness degree, and when $S_v = 0$, a full alignment exists and the terrain will be completely flat.

4.1.2 Saturated Gradient

The gradient of a surface is a vectorial field. In each point, the gradient point in the direction of the greatest rate of increase of the scalar field in that point, and whose module is the greatest rate of change in that point.

The gradient of $f(x,y)$ is defined to be the vector field whose components are the partial derivatives of f. That is:

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, ..., \frac{\partial f}{\partial x_n}\right) \tag{9}$$

In order avoid having path slopes greater than the robot can perform the gradient is saturated with that limit. That means that, if the gradient value exceeds that limit, the point will not be included in the list of accessible points determined by the robot limitations.

4.1.3 Construction of Matrix W

By using this matrix W the algorithm modifies the path that the robot is going to follow across the 3D surface. The way the matrix modifies the path is by giving a viscosity value for each point on the surface. It means that the propagation speed of the front end of the FM wave is modified. Hence, the time when the wave reaches each point will depend on that difficulty. It is possible to add as many characteristics as we need to get different paths. These characteristics will modify the viscosity at each point.

The saturated gradient, the spherical variance, and the height are three matrices G, Sv, and H with the same size as the vertex matrix (the 3D mesh). The value of each vertex of the 3D grid will be determined by the calculated gradient, spherical variance, and the height of each point.

The matrix W is a weighted average of each surface characteristic we are interested in, and in each case it gives more importance to the more important factors depending on the task requirements.

The values of the component matrices vary from 0 to 1, so the values of matrix W are also within this range. The components of matrix W with a value of 0 (less difficult) will be points in the *vertex* matrix with maximum speed. Hence, these are points which the robot can cross without any problem and at its maximum speed. The elements of W with a value of 1 will be points with a minimum speed, and in that case, the robot will not be able to pass across them.

$$W = a_1 \cdot G + a_2 \cdot Sv + a_3 \cdot H \tag{10}$$

where:

$$\sum_i a_i = 1 \tag{11}$$

After the difficulty matrix W is generated, the method runs the FM algorithm over the modified mesh (3D mesh + matrix W) to calculate the best trajectory. With the FM method the path found will be the less time path in the W metrics. If W is constant, this path will be the shortest because all the points in the surface will have the same 'speed' for the front propagation, i.e. the path is the geodesic. With a non constant matrix W, the proposed method changes that 'speed', since this matrix gives information about the difficulty to pass through each point of the surface. The trajectory will be modified depending on the surface conditions and characteristics and according to the robot limitations. Since the method modifies the 'speed' of the Fast Marching wave, and in each point W gives the difficulty that can be interpreted as maximum speed, it gives not only the best trajectory, but also the speed to control the robot.

4.1.4 Test on data taken in advance

As previously stated, in the proposed method we need terrain data that can be an elevation map, global or local laser data or a mixture of all. In relation to the outdoor environment reconstruction, a triangle-based 3D surface is chosen.

The method works in 3D, in order to create a triangular mesh, the algorithm reads the data from the bitmap file to create the three matrices X, Y, and Z and then, it builds a 3D mesh based on X, Y and Z coordinates. The fist step of the algorithm is to generate a Delaunay triangulation in 3D.

After the mesh is created, the algorithm extracts the vertices and the faces of the triangles. Using these values, the algorithm is able to model the 3D triangular surface.

Several paths over the surface already presented will be obtained between the same initial and final points. Those paths are obtained by varying the values of the weight factors a_i of matrix W.

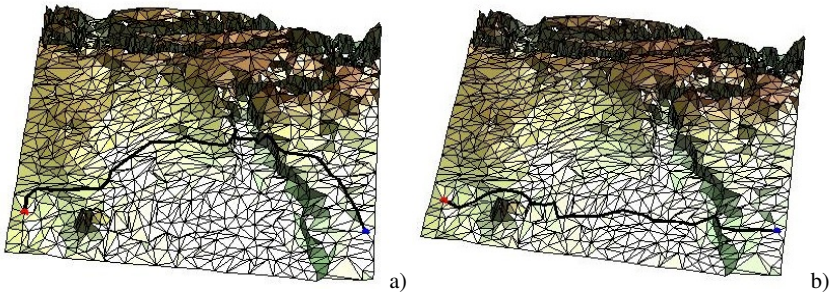


Fig. 25 Path calculated when a) $W = A$ and when b) $W = G$ in a mars map.

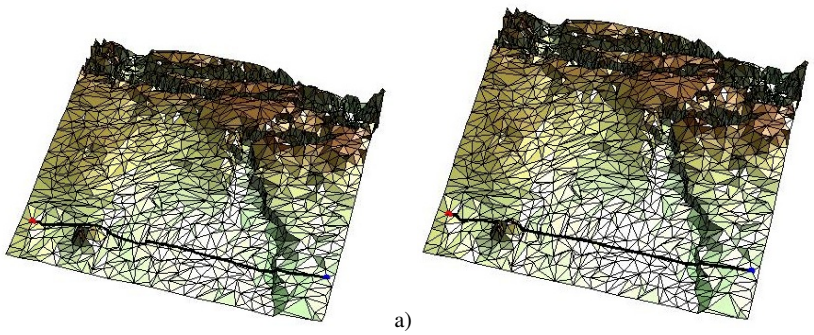
In the case that $W = A$, this implies that the difficulty of the path will be determined by the height of every point of the mesh. In Figure 25a), the path obtained when the height is penalised, without considering the roughness of the surface or its inclination, is presented. As can be observed, the calculated path will try to reach the final point passing trough the deepest part of the map.

On the other hand, if we decide to calculate the path penalising just the inclination of the surface, then the difficulty matrix is defined as $W = G$. In this case, as shown in Figure 25b), the path will follow the parts with smallest slope.

The general idea proposed in this section is the possibility of combining the different matrices in order to obtain a path that considers the height A, the roughness Sv, and the inclination G of the surface, among others. In the previous figures, it can be observed that, for the selected initial and final points, the height matrix favour that the path goes all the way trying to avoid the highest parts of it. On the other hand, the gradient matrix G favours the path with smallest slope. Therefore, we can select the values of each weight factor a_i in order to consider the limitations or features of the robot used.

The final step is to propagate the wave using as refraction index the difficulty matrix W from the goal point until it gets the present position of the robot and in this funnel shaped potential, the trajectory is calculated by using the gradient method.

Figure 26a) shows a view of the path obtained when $W = Sv$. As can be observed the result is an intermediate path. Fig. 26b) shows a view of the path obtained when $W = 0.20 * A + 0.40 * Sv + 0.40 * G$.



a) b)

Fig. 26 Path calculated using a) $W = Sv$ and when b) $W = 0.20 * A + 0.40 * Sv + 0.40 * G$ in a Mars map.

Moreover, the values of the weight factors a_i can be changed if the robot to be used is different or modified. It is also important to note that the trajectories calculated are a tentative path for the robot. The path can be modified online by modelling the environment with the robot sensors and recalculating the trajectory in a local area.

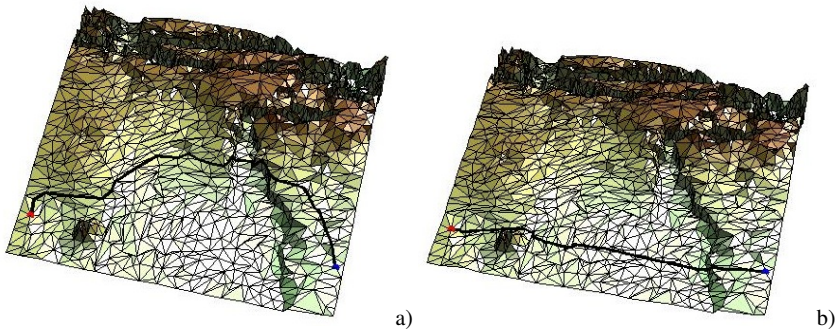
4.1.5 Introduction of the uncertainty in the slowness matrix W

When there is a certain uncertainty the robot has to modify the trajectory or the velocity. For example, in the case of robot in Mars, if the robot doesn't have enough information of part of the trajectory, because it hasn't visual data of that part, could be better to change the trajectory to zones the robot can visualise.

How can we introduce that uncertainty in the map in order to change the trajectory? Fortunately, the viscosity matrix W can also be understood as an uncertainty matrix, the grey degree can be understood as a measurement of the uncertainty.

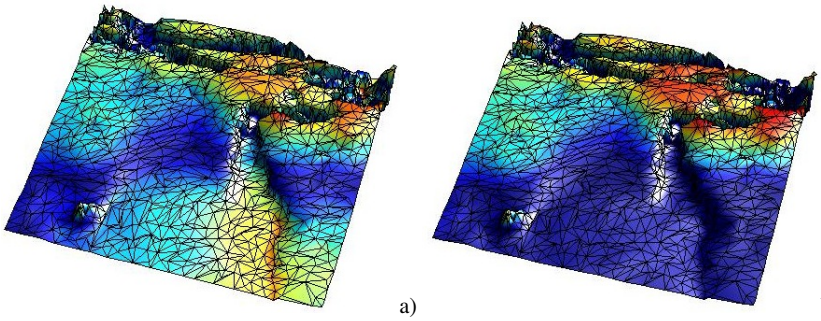
For example, suppose that the robot has no data of the points lower than its altitude, in that case the shadow points are represented in the matrix W with values next to zero (velocity of the media). In Fig. 27 is shown the difference between the robot trajectories without and with uncertain data of the points lower than the robot's altitude. As can be seen in the figure on the right, the trajectory is modified to not to go through the lower areas. In Fig. 28 is shown the difference between the difficulty-uncertainty W matrices when the robot has and hasn't data of the points lower than

its altitude. As can be seen in the right figure the lower parts have a bluish colour due to a bigger uncertainty and lower values in the W matrix that correspond to lower media velocity. In Fig. 29 is shown the difference between the wave expansion D matrices when the robot has and hasn't data of the points lower than its altitude. As can be seen, in the figure on the right, the expansion of the wave is more directed to the zone with less incertitude.



a) b)

Fig. 27 Difference between the robot trajectories a) without and b) with uncertain data of the points lower than its altitude.

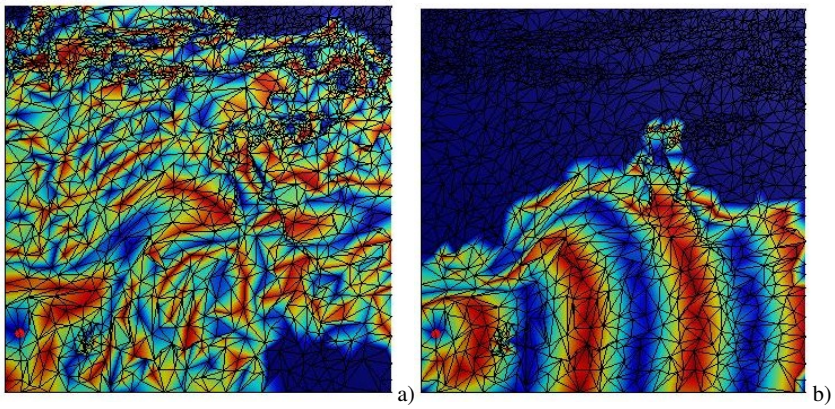


a) b)

Fig. 28 Difference between the difficulty-uncertainty W matrices when the robot a) has and b) hasn't data of the points lower than its altitude.

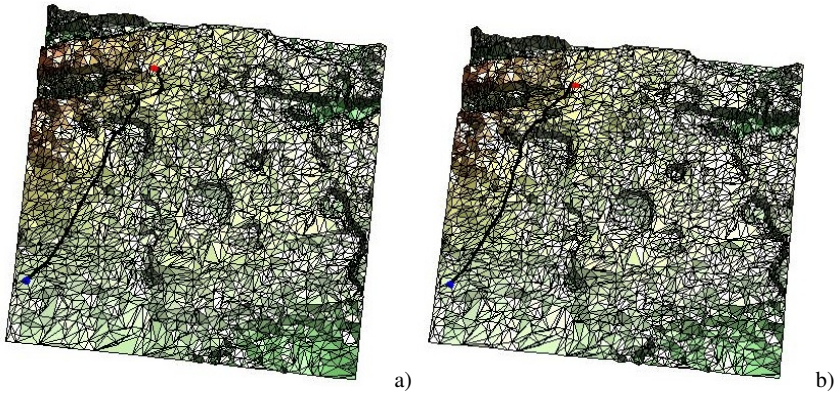
In Fig. 30 is shown the difference in the paths when the gradient is not saturated and when it is.

In Fig. 31 are shown the saturated gradient and the spheric variance corresponding to the previous figures.



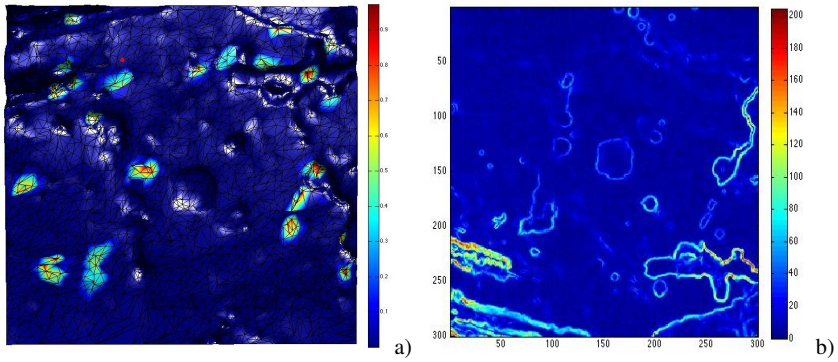
a) b)

Fig. 29 Difference between the wave expansion D matrices when the robot has and hasn't data of the points lower than its altitude.



a) b)

Fig. 30 Difference between the difference in the paths when the gradient a) is not saturated and b) when it is.



a) b)

Fig. 31 a) Spheric variance and b) the saturated gradient corresponding to the previous figures.

5 Conclusions and Future Work

As shown along the paper, the FM and FM^2 methods are very powerful when applied to robot motion planning. Many different problems can be faced with the same underlying method in addition to minor modifications.

When applied to 2D or 3D environments, the FM^2 method is able to provide efficient solutions in a very short period of time, reaching even real-time applications. However, as it is based on grid maps, it suffers from the curse of dimensionality. The number of cells in an environment representation increases polinomically with the dimensions.

Therefore, future work focuses on applying different heuristics, and include previous experience in the planner in order to boost the plannification process.

References

1. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. Proc. National Academy of Sciences, 93, 1591-1595 (1996)
2. Jbabdi, S., Bellec, P., Toro, R., Daunizeau, J., Plgrini-Issac, M., and Benali, H.: Accurate anisotropic fast marching for diffusion-based geodesic tractography. Int. J. Biomed. Imaging., vol. 2008, p. 12, 2008.
3. Li, H., Xue, Z., Cui, K., and Wong, S.T.C.: Diffusion tensor-based fast marching for modeling human brain connectivity network. Comp. Med. Imag. Graph., vol. 35, no. 3, pp. 167?178, 2011.
4. Yang, K., Li, M., Liu, Y., and Jiang, C.: Multi-points fast marching: A novel method for road extraction. Proc. 18th Int. Conf. Geoinformatics: GIScience in Change, Geoinformatics, June 2010, pp. 1?5.
5. Yatziv, L., Bartesaghi, A., Sapiro, G. (2005). A Fast O(n) Implementation of the Fast Marching Algorithm. Journal of Computational Physics 212. pp. 393-399.
6. Sethian, J. A. (1996). Theory, Algorithms, and Aplications of Level Set Methods for Propagating Interfaces. Acta Numerica: pp. 309-395.
7. Sethian, J. A. 1996. "Level Set Methods". Cambridge University Press.
8. Adalsteinsson, D., and Sethian, J. A. (1995). A Fast Level Set Method for Propagating Interfaces. J. Comput. Phys. 118(2): pp. 269-277.
9. Alton, K. R., Mitchel, I. M., (2008). Fast Marching Methods for Stationary Hamilton-Jacob Equations with Axis-aligned Anisotropy. SIAM J. Numerical Analysis 47(1),pp. 363–385.
10. Petres, C., Pailhas, Y., Evans, J., Petillot, Y. and Lane, D.(2005). Underwater Path Planing Using Fast Marching Algorithms. IEEE Oceans 2005 Europe Conferences, vol. 2, pp. 814-819.
11. Garrido, S., Moreno, L., Blanco, D. (2006). Voronoi Diagram and Fast Marching applied to path planning. Robotics and Automation. ICRA 2006. Proceedings 2006 IEEE International Conference on, pp. 3049–3054.
12. Garrido, S., Moreno, L., Blanco, D. (2007) Sensor-based global planning for mobile robot navigation. Robotica 25 , pp. 189-199.
13. Garrido, S., Moreno, L., Blanco, D. (2008) Exploration of 2D and 3D environments using Voronoi transform and Fast Marching method. Journal of Intelligent and Robotic Systems 55(1), pp. 55–80.
14. Garrido, S., Moreno, L., Abderrahim, M., Blanco, D. (2009) FM2: A real-time sensor-based feedback controller for mobile robots. International Journal of Robotics and Automation 24(1), pp. 3169–3192.

15. Valero-Gomez, A., Gomez, J., Garrido, S. Moreno, L. (2013) The Path to Efficiency: Fast Marching Method for Safer, More Efficient Mobile Robot Trajectories. Robotics and Automation Magazine, IEEE, 20(4). pp.111–120.
16. Gomez, J. V., Vale A., Valente F., Ferreira J., Garrido S., and Moreno L. (2013). Fast Marching in Motion Planning for Rhombic Like Vehicles Operating in ITER. 2013 IEEE International Conference on Robotics and Automation : pp. 5533-5538.
17. Ulrich, I. and Borenstein, J. (2000). Vfh*: local obstacle avoidance with lookahead verification, Proc. IEEE Int. Conf. on Robotics and Automation, pp. 2505-2511.
18. Minguez, J., Montano, L. (2001). Global nearness diagram navigation. Proc. IEEE Int. Conf. on Robotics and Automation (Seoul, Korea), pp. 33-39.
19. Castejon, C., Boada, B., Blanco, D., Moreno, L. (2005). Traversable region modeling for outdoor navigation, Journal of Intelligent and Robotic Systems 43 (2-4), pp. 175–216.