Fast Marching Method: Application of the Eikonal equation in path planning problems

Luis Moreno / Santiago Garrido

Dpto. Ing. de Sistemas y Automática
Universidad Carlos III
Madrid

April 10, 2016

Table of contents

Classical Classification of the path planning algorithms

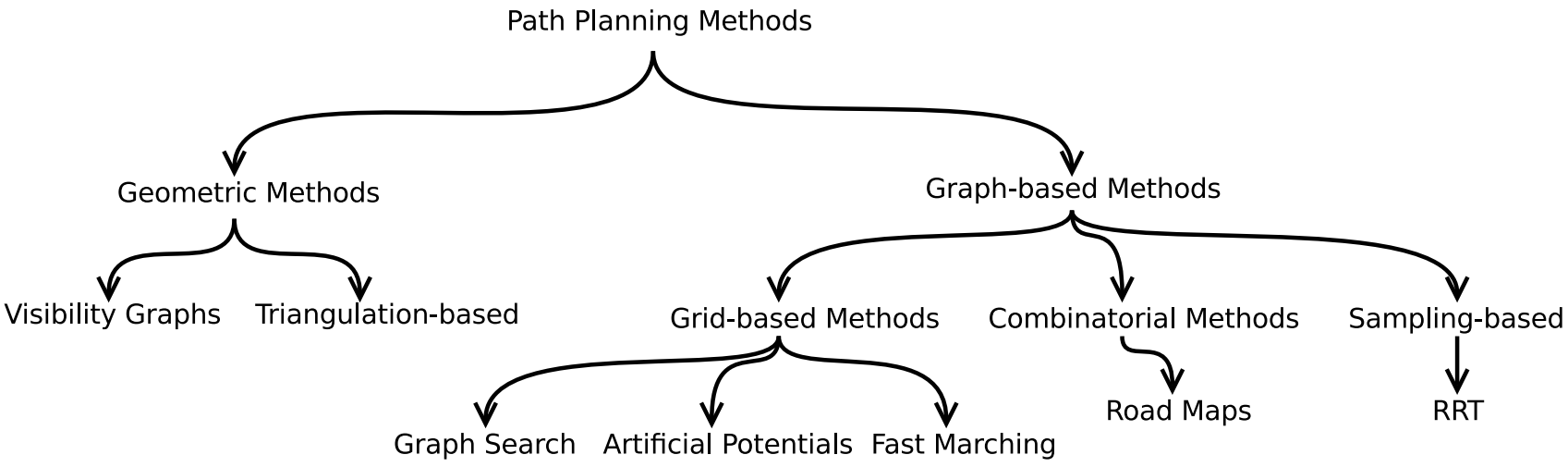


Figure: Classification of the current path planning approaches.

Classic classification but a little confusing

Path planning algorithms

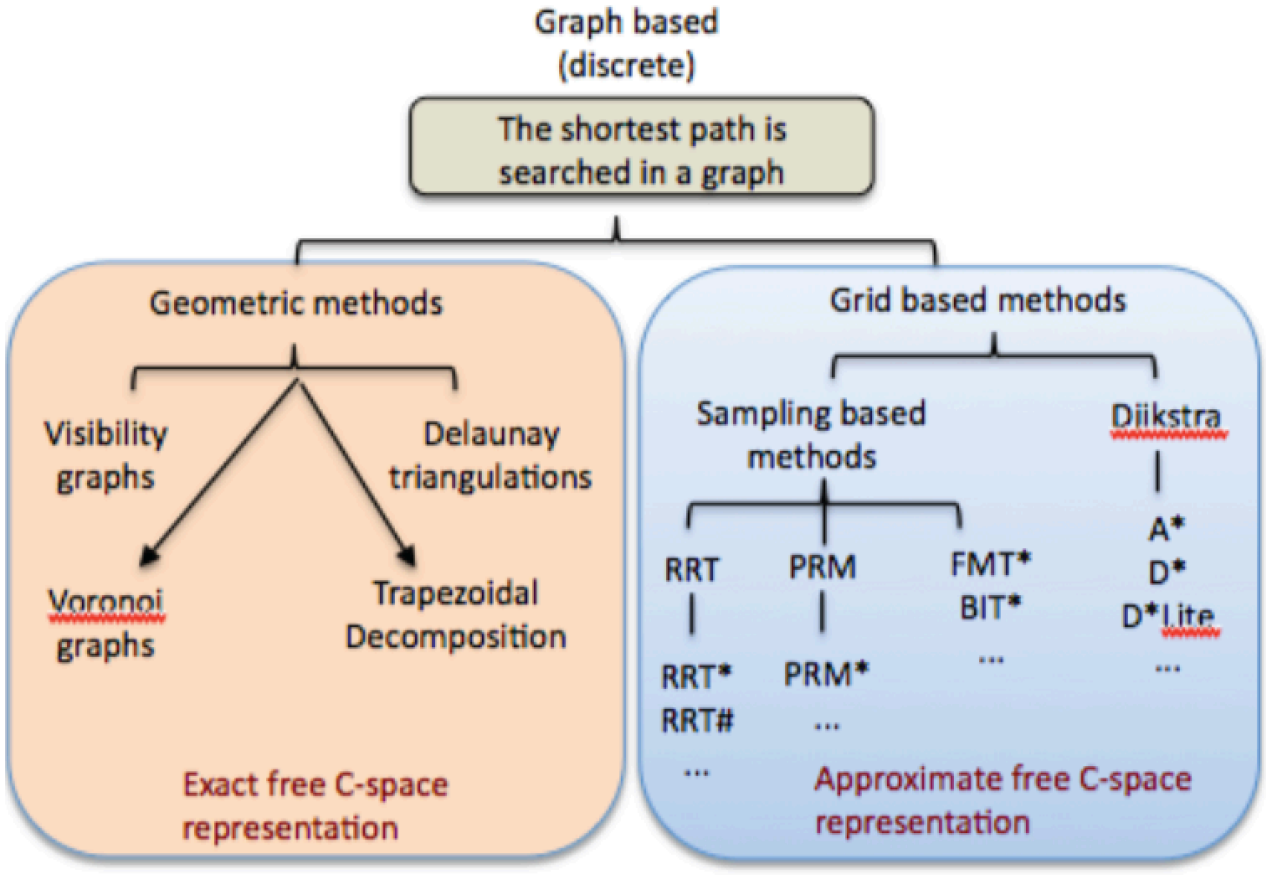
From another point of view

According the way of searching to obtain the path, path planning algorithms can be classified as :

- Graph based methods, the path is obtained after a search in a graph.
- Surface based methods, the path is obtained using a gradient optimization method.

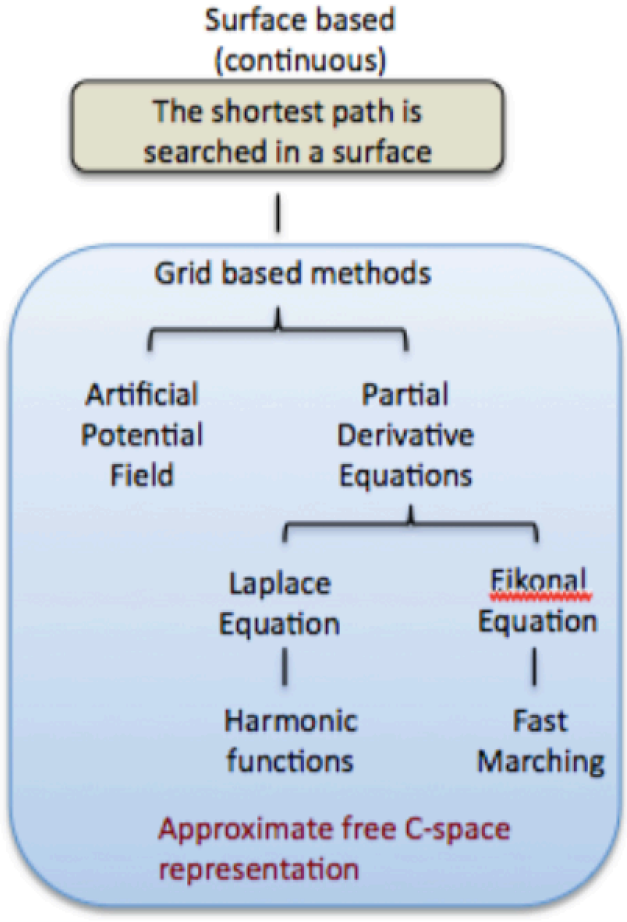
Graph based methods

Scheme



Surface based methods

Scheme



Graph based methods

Deterministic

Those methods operates in two steps:

- In the first step an adjacency graph of the free areas of the environment is built and
- In a second step the connectivity graph is searched to obtain the minimum cost path to go from one node to other in the graph.
- Different methods can be included in this category:
  - Visibility graphs.
  - Delaunay triangulation.
  - Decomposition (trapezoidal) methods,
  - Voronoi graph methods,
  - Dijkstra, A*, D*, Incremental A*, and D* Lite.
  - ...

Grid based methods

Triangulation methods

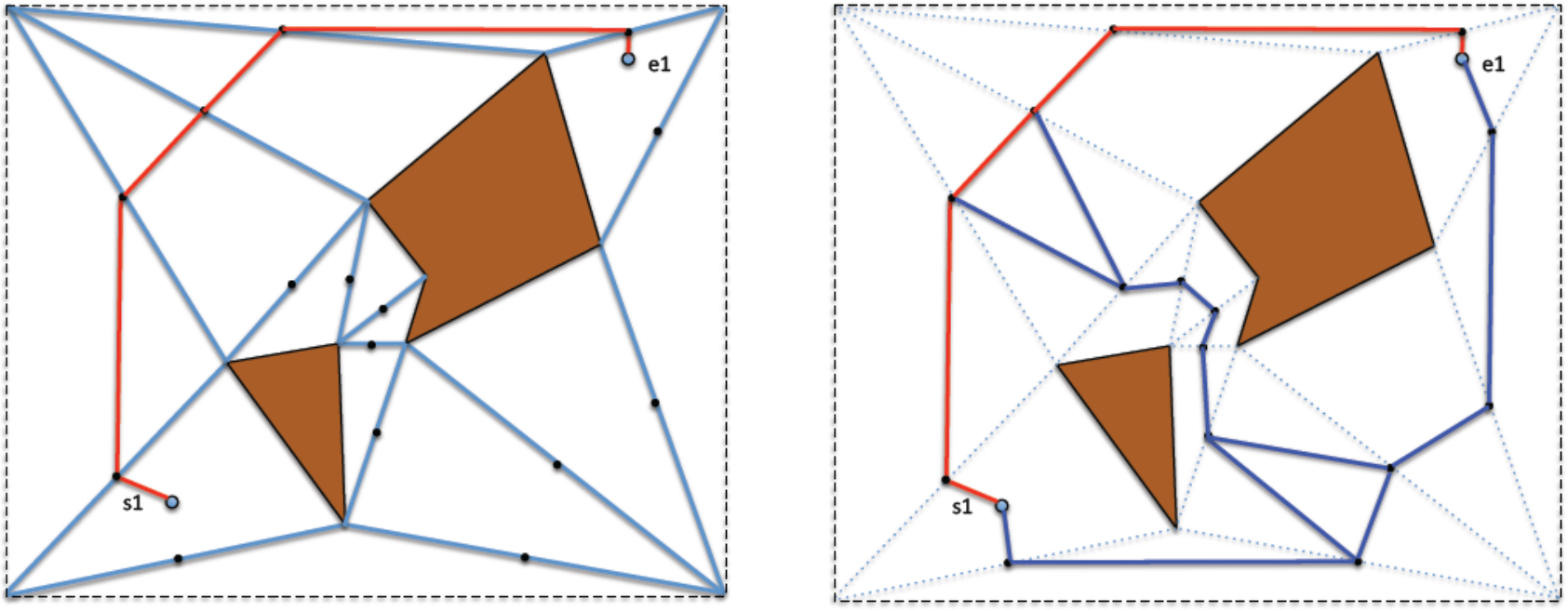
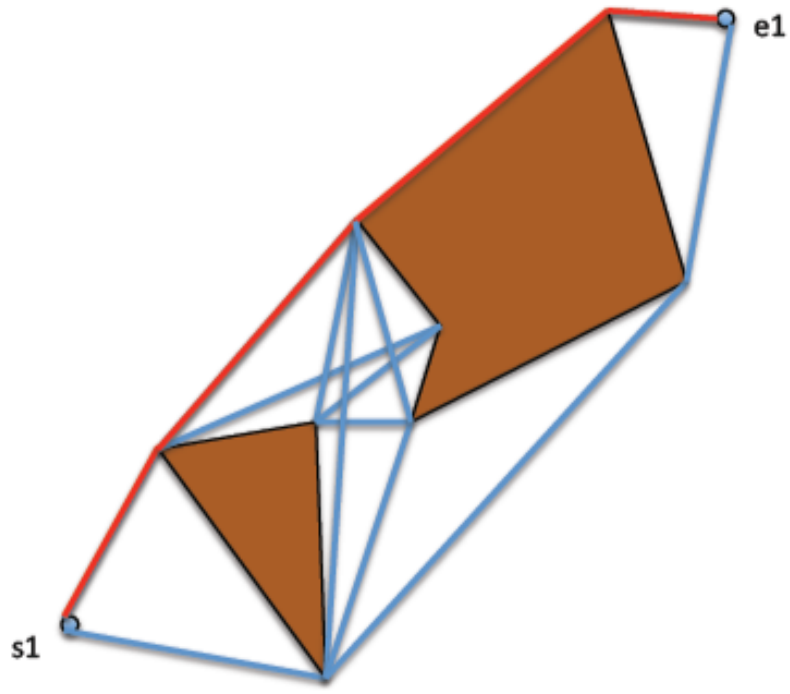


Figure: Triangulation, induced graph and shortest path

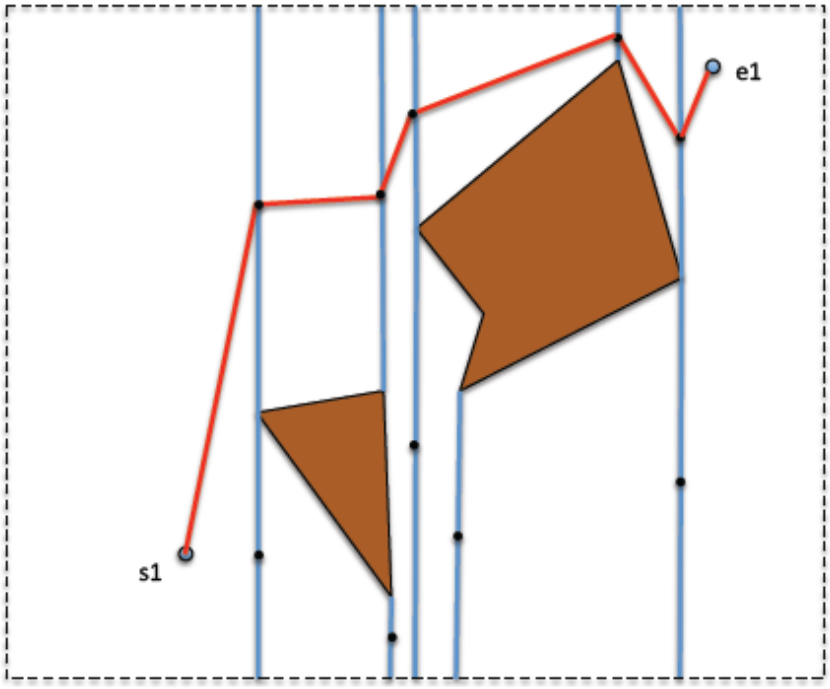
Grid based methods

Visibility graphs



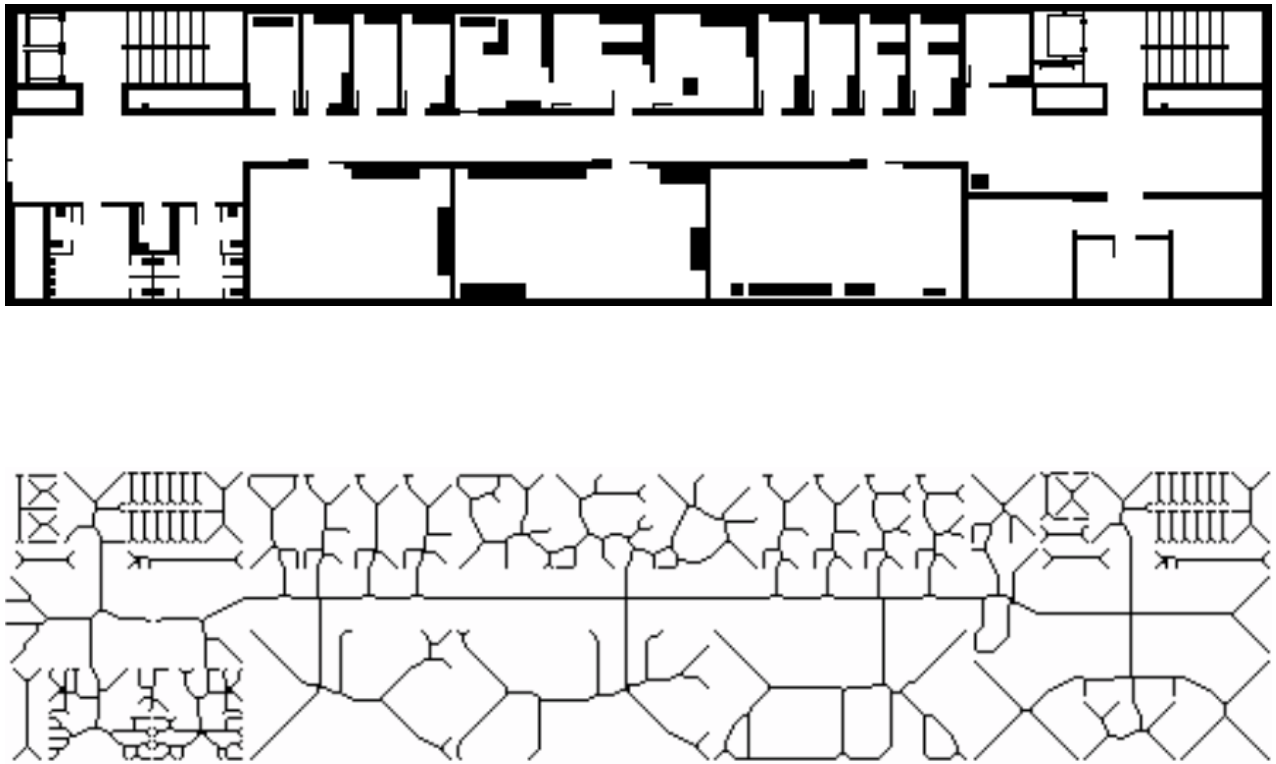
Grid based methods

Decomposition methods



Grid based methods

Voronoi diagrams



Graph based methods

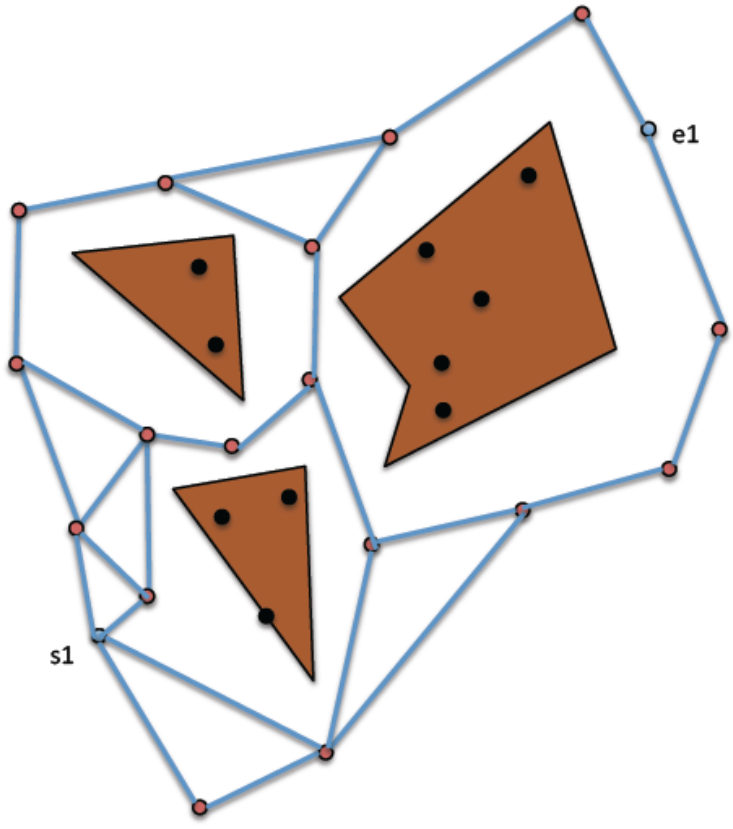
Sampling based

A particular case of the graph based methods are the so called sampling based methods

- In these methods the adjacency graph is built by using some kind of sampling strategy to decide the candidate new points to be included in the adjacency graph.
- Depending on the strategy used to obtain the sampled points there exists different methods:
  - PRM (probabilistic roadmap method) and
  - RRT (Rapid Random Tree).

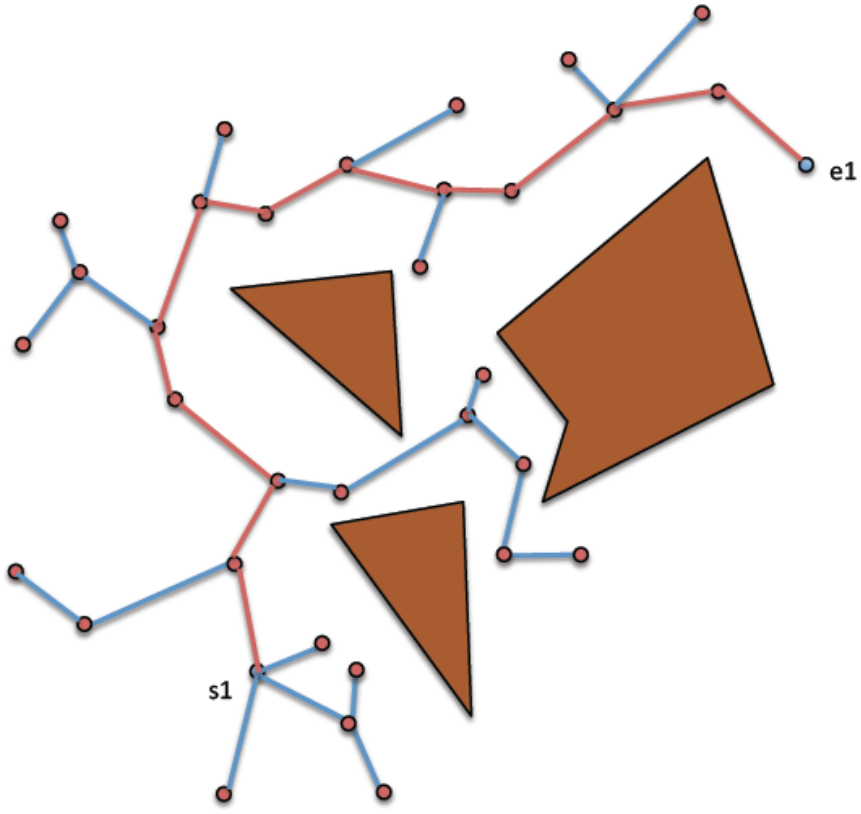
Grid based sampling methods

PRM



Grid based sampling methods

RRT



Surface based path planning methods

Surface based methods also operates in two steps:

- In the first step a cost surface is generated. Typically defined such that the minimum cost point is located in the goal point.
- And in a second step a continuous path is obtained using a gradient descend technique to move from the initial point to the goal one.
- The cost surface is calculated at the points of a grid (which implicitly establish an adjacency scheme) but those methods operates in a continuous mode letting the robot move freely through the surface without being restricted to the edges that connect the grid cells.

Surface based methods

Between this family of methods we can include:

- Artificial potential fields. The robot is treated as an electric charge moving under a potential field in which the obstacles have the same charge, so they repeal the robot from them. The goal point has the opposite charge, attracting the robot towards it. The main drawback of this approach is that it is prone to local minima and oscillations.
- Harmonic functions. It can be considered an evolution of potential methods, but based on the solution of the Laplace equation. This equation holds for the steady temperature in an isotropic medium or electrostatic potentials at points of space. The main drawback of this approach is that it is prone to quantization effects and it is slow.
- The Fast Marching Method . In this case, the cost for each node is related with the time a propagating wave takes to reach that node.

Surface based methods

Artificial potentials

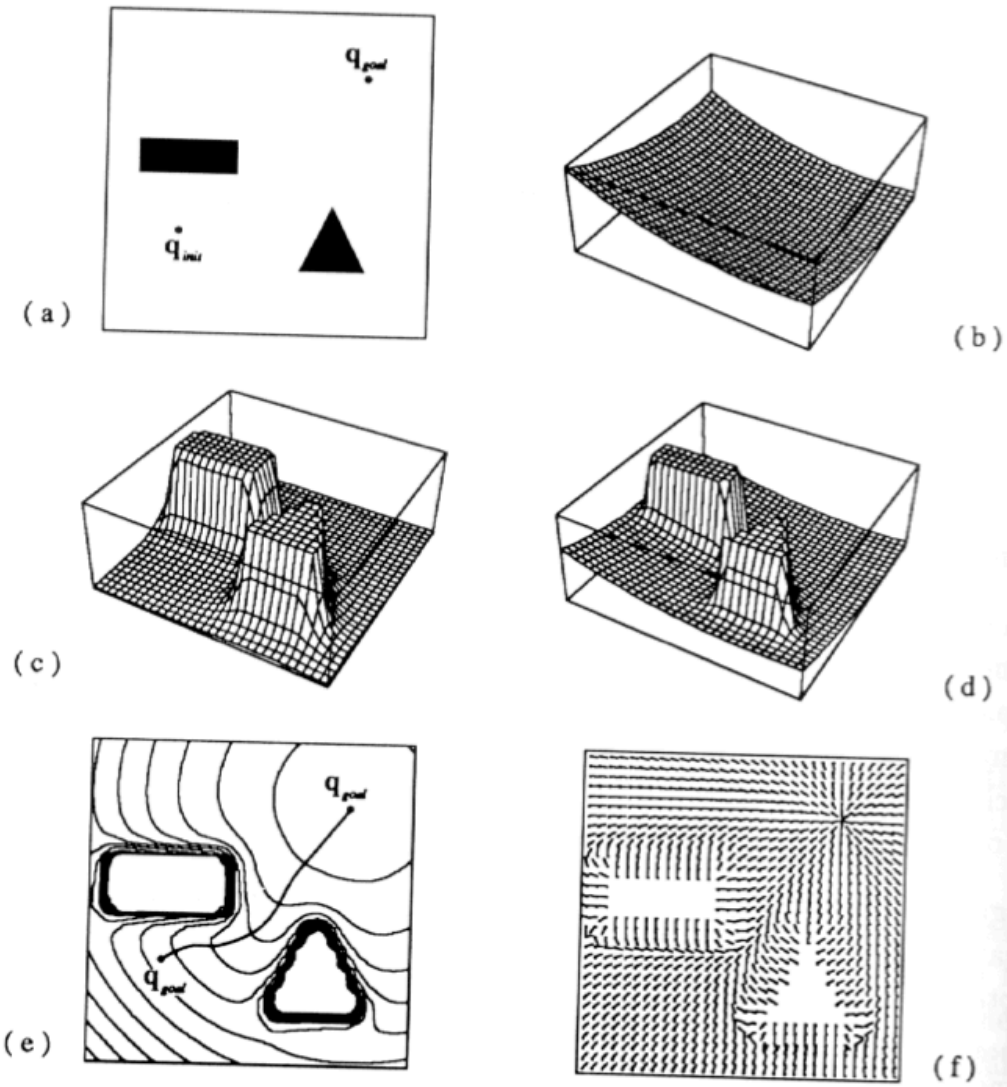


Figure: Latombe book

Surface based methods

Harmonic functions

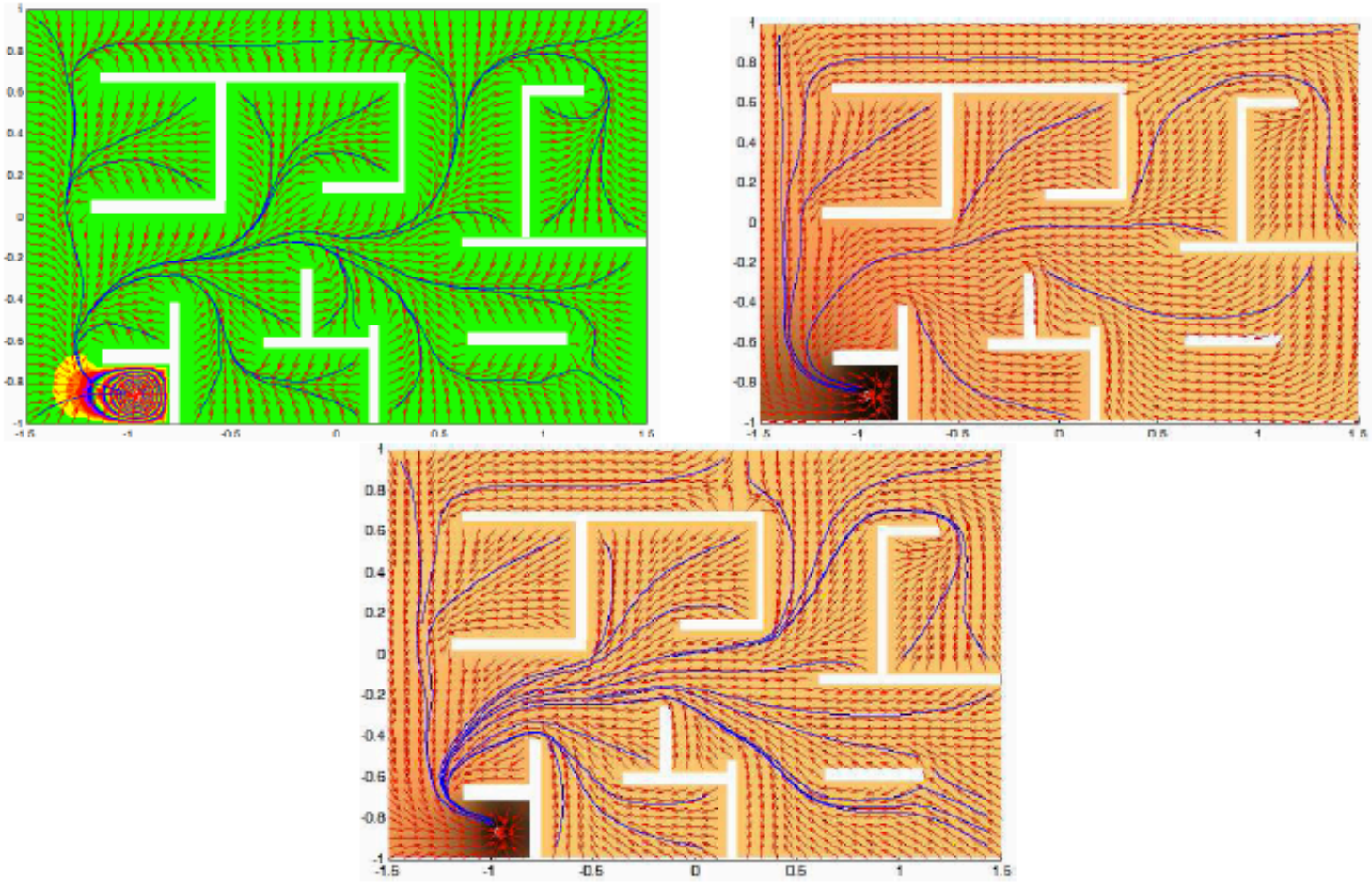


Fig. 1. Robot's trajectories obtained to solved Laplace's equation with the boundary conditions of Dirichlet, Neumann and Robin, respectively.

Surface based methods

Harmonic functions: boundary conditions effect

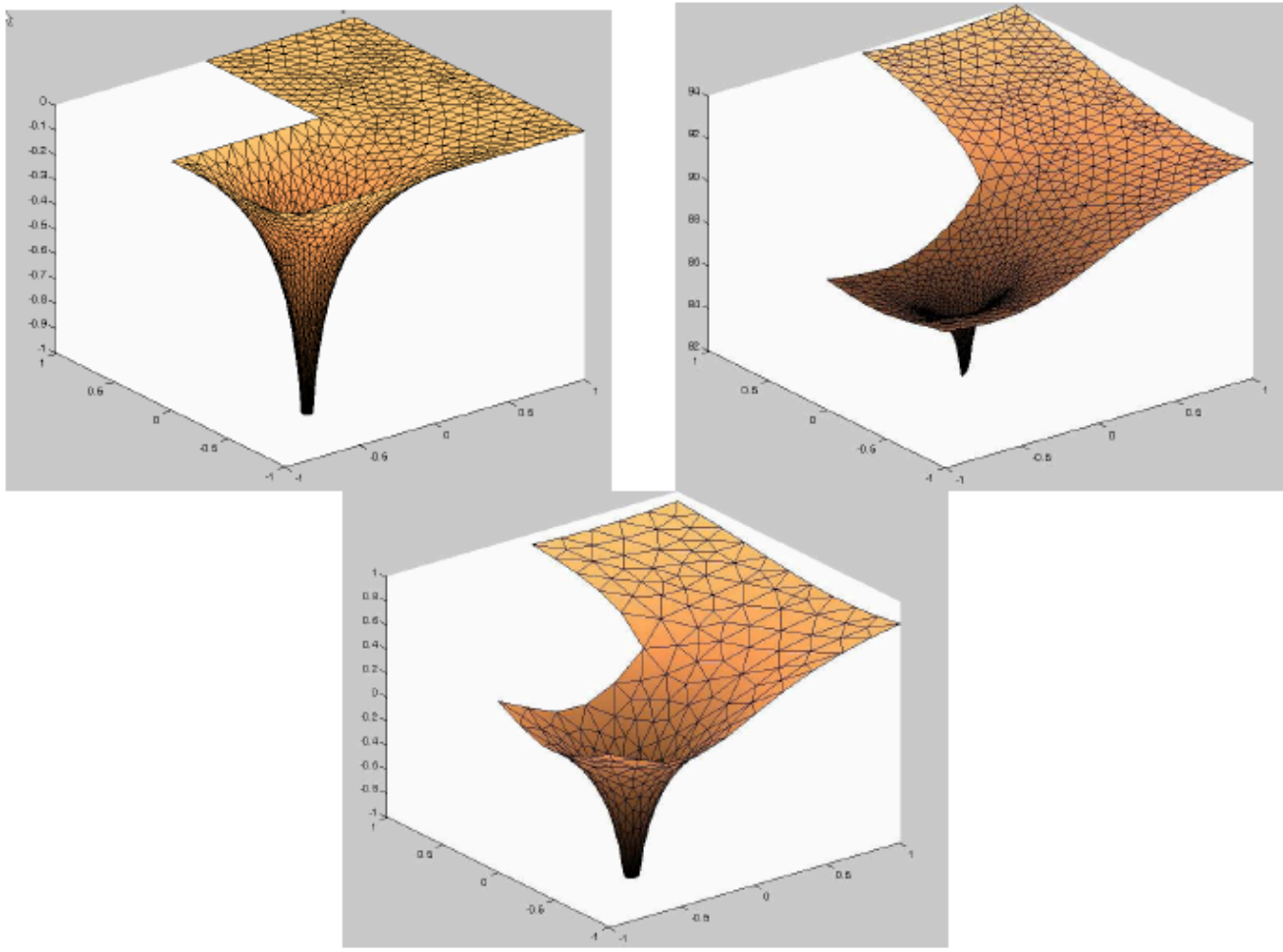


Fig. 2. Solutions of Laplace's equation with the boundary conditions of Dirichlet, Neumann and Robin, respectively.

Eikonal equation in path planning

Eikonal equation

- The eikonal equation appears in problems of wave propagation. It is derivable from Maxwell's equations of electromagnetics, and provides a link between physical (wave) optics and geometric (ray) optics.
- In Geometrical Optics, Fermat's least time principle for light propagation in a medium with space varying refractive index $\eta(x)$ is equivalent to the eikonal equation .
- This equation is also known as Fundamental Equation of the Geometrical Optics.
- The eikonal (from the Greek eikon, which means image). Constant values of the eikonal represent surfaces of constant phase, or wavefronts.
- The normals to these surfaces are rays (the paths of energy flux).

Eikonal equation

- A way to characterize the position of a front in expansion is to compute the time T, in which the front reaches each point of the space.
  - For one dimension the equation for the arrival function T can be obtained easily from (x distance and the speed F of propagation) $T : x = FT$.
  - The spatial derivative of the solution function is: $1 = F\frac{dT}{dx}$ and therefore the magnitude of the gradient of the arrival function $T(x)$ is inversely proportional to the speed, $\frac{1}{F} = |\nabla T|$.
  - For multiple dimensions, the same concept is valid because the gradient is orthogonal to the level sets of the arrival function $T(x)$.
  - The speed F depends only on the position, then the equation $\frac{1}{F} = |\nabla T|$ or the Eikonal equation:

$$|\nabla T|F = 1.$$

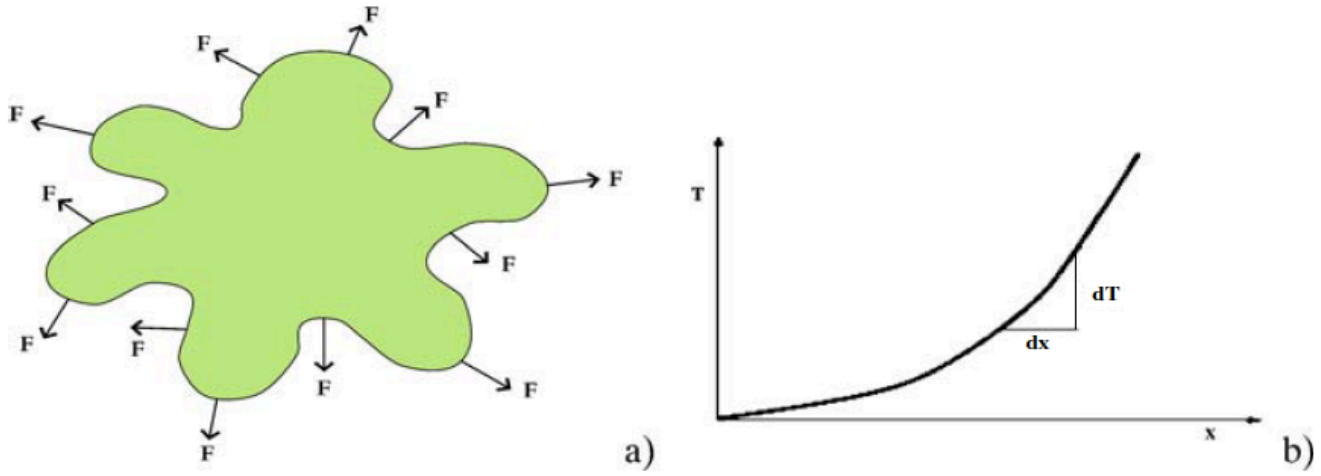
Fast Marching Method (FMM)

- The Fast Marching Method (FMM) is a particular case of Level Set Methods, initially developed by Osher and Sethian (Osher 1988).
- It is an efficient computational numerical algorithm for tracking and modeling the motion of a physical wave interface (front), denoted Γ.
- Ithas been applied to different research fields including computer graphics, medical imaging, computational fluid dynamics, image processing, computation of trajectories, etc.

Fast Marching Method

Idea

- Intuitively, Fast Marching Method gives the propagation of a front wave in an inhomogeneous media.
- Imagine that the curve or surface moves in its normal direction with a known speed F . The objective would be to follow the movement of the interface while it evolves.



Wavefront propagating with velocity F (a), and arrival function T(x), for an unidimensional wavefront (b).

Eikonal equation

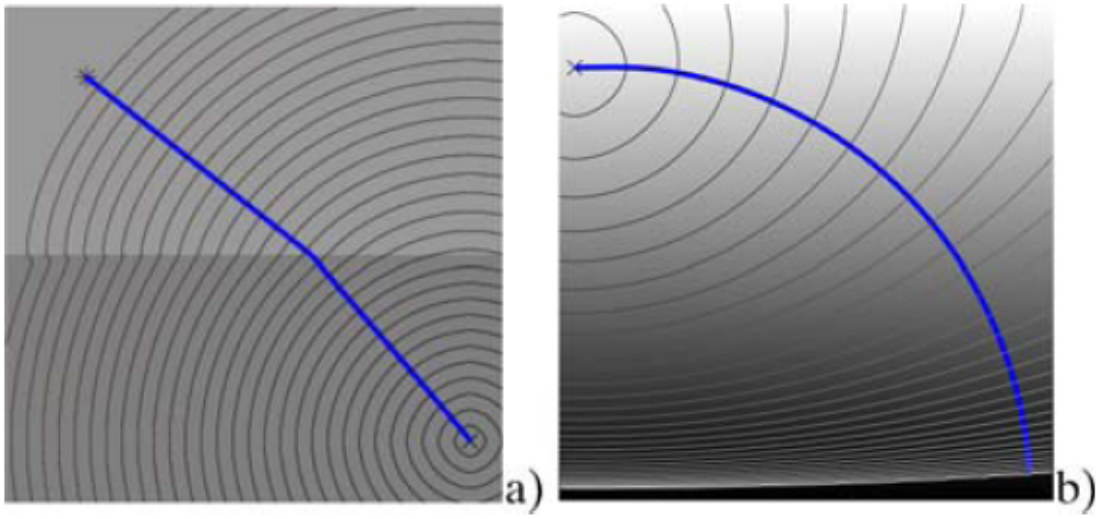


Figure: Propagation of a wave and the corresponding minimum time path when there are two media of different slowness (diffraction) index. (a), the same with an vertical gradient (b).

Fast Marching Method

Interface propagation

- The wavefront is called the interface and can be a flat curve in 2D, or a surface in 3D (can be generalized to n dimensions). The method calculates the time T that a wave needs to reach every point of the space.
- We assume that the front Γ evolves by motion in the normal direction of the front.
- The speed, denoted F, does not have to be the same everywhere, but it is always non-negative. At a given point, the motion of the front is described by the equation known as the Eikonal equation:

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

  where x is the position, $F(x)$ the expansion speed of the wave at that position, and $T(x)$ the time that the wave interface requires to reach x.
- The magnitude of the gradient of the arrival function $T(x)$ is inversely proportional to the velocity:

$$\frac{1}{F(x)} = |\nabla T(x)|$$

Fast Marching Method

Interface propagation

- The $T(x)$ function originated by a wave that grows from one single point presents only a global minima at the source and no local minima.
- As $F(x) \geq 0 \ \forall x$ the wave only grows (expansion), and hence, points farther from the source have greater T. A local minima would imply that a point has a T value lesser than a neighbourg point which is nearer to the source. This is impossible as this neighbourg must have been reached by the wave sooner.

Fast Marching Method

Interface propagation solution

- Due to the front can only expand ($F(x) \geq 0 \quad \forall x$), the arrival time $T(x)$ is single valued. Sethian proposed a discrete solution for the Eikonal equation.
- In 2D the space is discretized using a grid map, denoting by i, j the row i and column j of the grid map that corresponds to a point $p(x_i, y_j)$ in the real world. The discretization of the gradient $\nabla T(x)$ drives to the following equation:

$$max(D_{ij}^{-x}T, 0)^2 + min(D_{ij}^{+x}T, 0)^2 + max(D_{ij}^{-y}T, 0)^2 + min(D_{ij}^{+y}T, 0)^2 = \frac{1}{F_{ij}^2} \tag{1}$$

or to the one proposed by Sethian, simpler but less accurate:

$$max(D_{ij}^{-x}T, -D_{ij}^{+x}, 0)^2 + max(D_{ij}^{-y}T, -D_{ij}^{+y}, 0)^2 = \frac{1}{F_{ij}^2} \tag{2}$$

where:

$$\begin{aligned} D_{ij}^{-x} &= \frac{T_{i,j} - T_{i-1,j}}{\triangle x} \\ D_{ij}^{+x} &= \frac{T_{i+1,j} - T_{i,j}}{\triangle x} \\ D_{ij}^{-y} &= \frac{T_{i,j} - T_{i,j-1}}{\triangle y} \\ D_{ij}^{+1} &= \frac{T_{i,j+1} - T_{i,j}}{\triangle y} \end{aligned} \tag{3}$$

Fast Marching Method

Interface propagation

- and $\triangle x$ and $\triangle y$ are the grid spacing in the x and y directions.
- Substituting Eq. 3 in Eq. 2 and letting

$$\begin{aligned} T &= T_{i,j} \\ T_1 &= min(T_{i-1,j}, T_{i+1,j}) \\ T_2 &= min(T_{i,j-1}, T_{i,j+1}) \end{aligned} \tag{4}$$

- we can rewrite the Eikonal Equation, for a discrete 2D space as:

$$max\left(\frac{T - T_1}{\triangle x}, 0\right)^2 + max\left(\frac{T - T_2}{\triangle y}, 0\right)^2 = \frac{1}{F_{i,j}^2} \tag{5}$$

Fast Marching Method

Interface propagation example

Example of a front propagation:

Video FM.mp4

FM Algorithm

The algorithm has three stages: initialization, main loop, and finalization.

- Initialization : The algorithm starts by setting $T = 0$ in the cell or cells that originate the wave. These cells are labeled as frozen. Afterwards it labels all their Manhattan neighbors as narrow band, computing T (Eq. 5) for each of them.
- Main loop: In each iteration the algorithm will solve the Eikonal Equation (Eq. 5) for the Manhattan neighbors (that are not yet frozen) of the narrow band cell with the lesser T value. This cell is then labeled as frozen. The narrow band maintains an ordered list of its cells. Cells are ordered by increasing T value (first cells have lesser T values).
- Finalization: When all the cells are *frozen* (the *narrow band* is empty) the algorithm finishes.

Example 1

Origin: one point

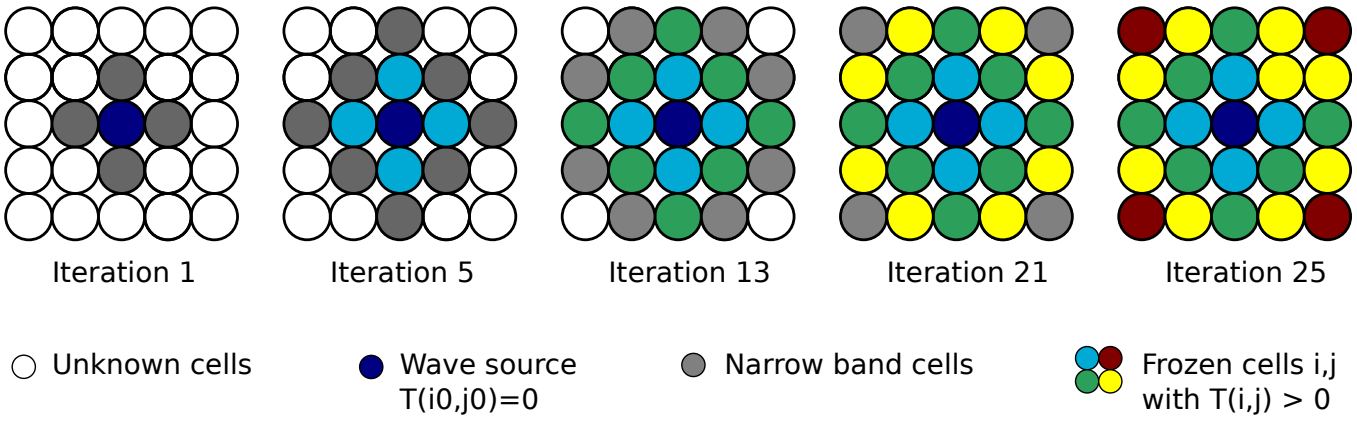


Figure: Iterations of the FMM in a 5x5 grid map with one wave source.

Example 2

Source: two points simultaneously

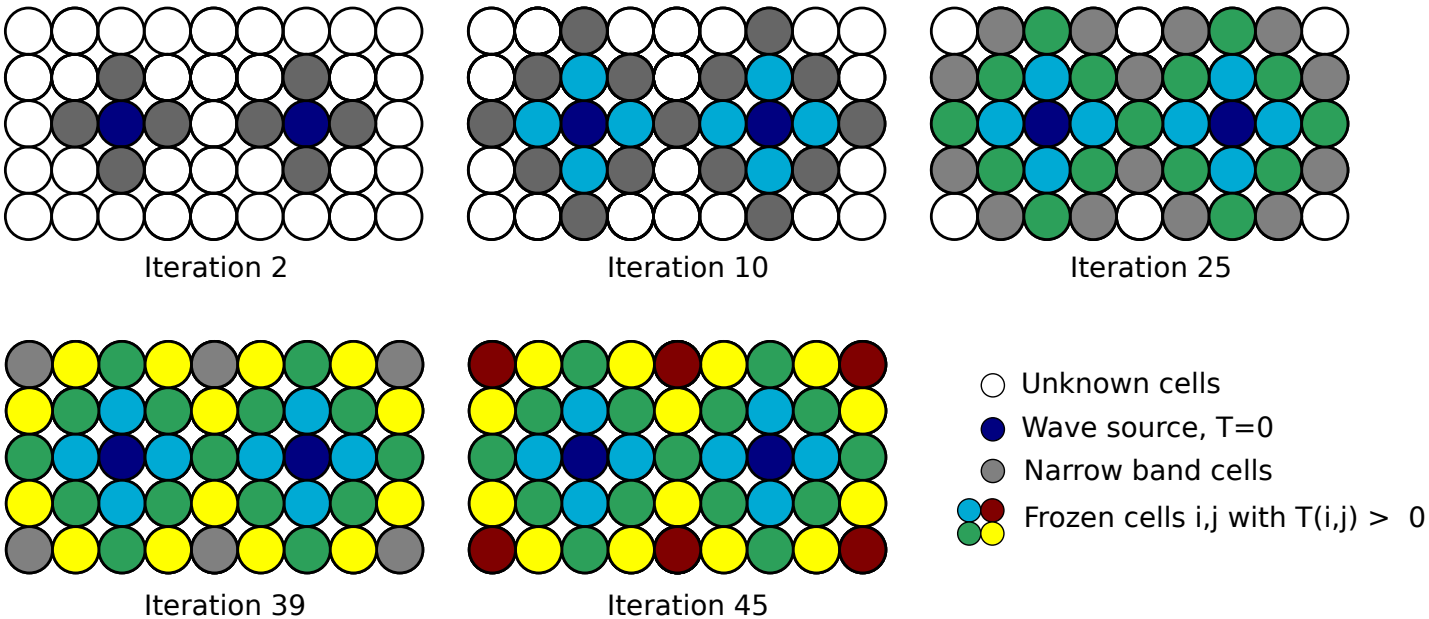


Figure: Iterations of the FMM in a 5x9 grid map with two wave sources.

Algorithm

Initialization

- Inicialización:

G: rejilla de tamaño m x n.
g_s: celdas donde se origina la onda.
NB: narrow band.
V:velocidad de la celda.

g: celda.
g_n: vecino von Neumann.
T:tiempo de viaje.
E:etiqueta de la celda: Open | Narrow | Frozen.

```
for each g_s ∈ G
do {
    g_s.T ← 0
    g_s.E ← Frozen
    for each g_n de g_s
    do {
        if g_n.E != Frozen | Celda ocupada | g_n.V = 0
        then {
            p ← resolver Eikonal
            if g_n.E = Narrow
            then {
                if p < g_n.T
                    g_n.T ← p
            }
            if g_n.E = Open
            then {
                g_n.E ← Narrow
                g_n.T ← p
                añadir g_n a NB
            }
        }
    }
}
```

Algorithm

Main loop

- **Iteraciones**

```
while NB > 0
do {
    g ← Celda de menor TT en NB
    g.E ← Frozen
    for each g_n de g
    do {
        if g_n.E != Frozen | Celda ocupada | g_n.V = 0
        then {
            p ← resolver Eikonal
            if g_n.E = Narrow
            then {
                if p < g_n.T
                    g_n.T ← p
            }
            if g_n.E = Open
            then {
                g_n.E ← Narrow
                g_n.T ← p
                añadir g_n a NB
            }
        }
    }
}
```

Application to path planning

- Consider a binary grid map, in which obstacles are valued as 0, and free space as 1. These values can be taken as the wave expansion speed F over the grid map (at obstacles, speed is 0 and on free space, wave expansion speed is constant and equals to 1).
- To compute the path between two points p_0 and p_1 we expand a wave from p_1 until it reaches p_0.
- Due to the wave expansion properties, the path followed by the wave interface will be always the shortest trajectory in time and due to speed is constant, this trajectory is also the shortest solution in distance.
- The wave is originated from the target point, hence, the computed $T(x)$ field will have only one minima at the target point. Hence, following the maximum gradient direction from the initial point we will reach the target point, obtaining the trajectory.

Example

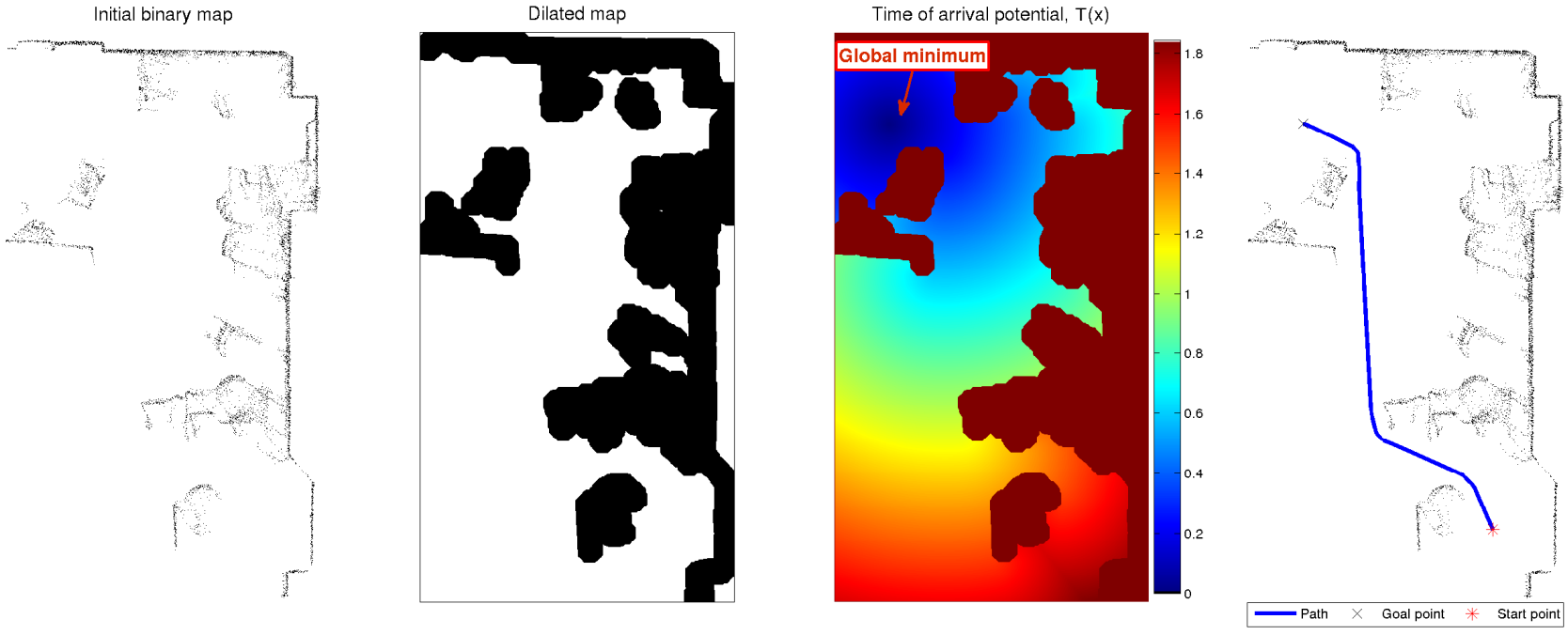


Figure: Steps of the FMM applied to path planning.

- Binary map (3D laser range sensor projected on 2D). The obstacles (labeled as 0, black points) are dilated by the maximum radius of the robot. Then, the FMM is applied using the goal point as a wave source. Once the interface Γ has reached the start point the algorithm stops expanding.

Path determination

Maximum gradient

- The resulting grid map stores at any pixel the time T required to reach that pixel.
- The isocurves are the points that have been passed through at the same instant of time (the front wave) and the maximum gradient direction at any point is the normal direction to the isocurve.
- To obtain the path between the initial and the goal points we follow the maximum gradient direction starting at the initial point until goal.

Problems

- The trajectories generated are optimal according to the minimal Euclidean distance criterion.
- Highly unsafe, the path is too close to the obstacles, this can be alleviated by growing the obstacles in a pre-specified distance (at least the radius of he robot) but this solution is inefficient (particularly in doors or narrow crossings).
- The trajectories generated are not smooth.
- These facts turn FMM into an unreliable path planner for most robotic applications.
- But ... these problems can be solved while maintaining its advantages.

How to avoid these problems

- Basic idea:
  - Play with the refraction index
  - We determine a refraction index that is the maximum allowable speed by the robot at that point of the map.
  - This works as a kind of viscosity, the robot moves faster when is far from the obstacles and slower when is close to them.
- This two step approach gives us some substantial advantages:
  - The robot path is safer, due to it separates from obstacles decreasing the collision risk.
  - The robot path is faster (not necessarily shorter in distance, but shorter in time).
  - We obtain not only the path plan but also the motion plan, that means we have the points of the path and the speed at that points.

Fast Marching Square (FM^2)

Method idea

- Starting with the evidence grid map in which obstacles are labeled as 0 and free space as 1.
- Applying the FMM to this map being the obstacles surface the generator of a wave source (hence, several waves are expanded at the same time).
- The map resulting represents a kind of scalar (potential) field. As pixels get far from the obstacles, the computed T value is greater. This map can be seen as a velocities map or a refraction map.
- If we consider the T value as a measure proportional to the maximum allowed speed of the robot at each point, we can appreciate that speeds are lower when the pixel is close to the obstacles, and greater far from them.

Fast Marching Square

Basic

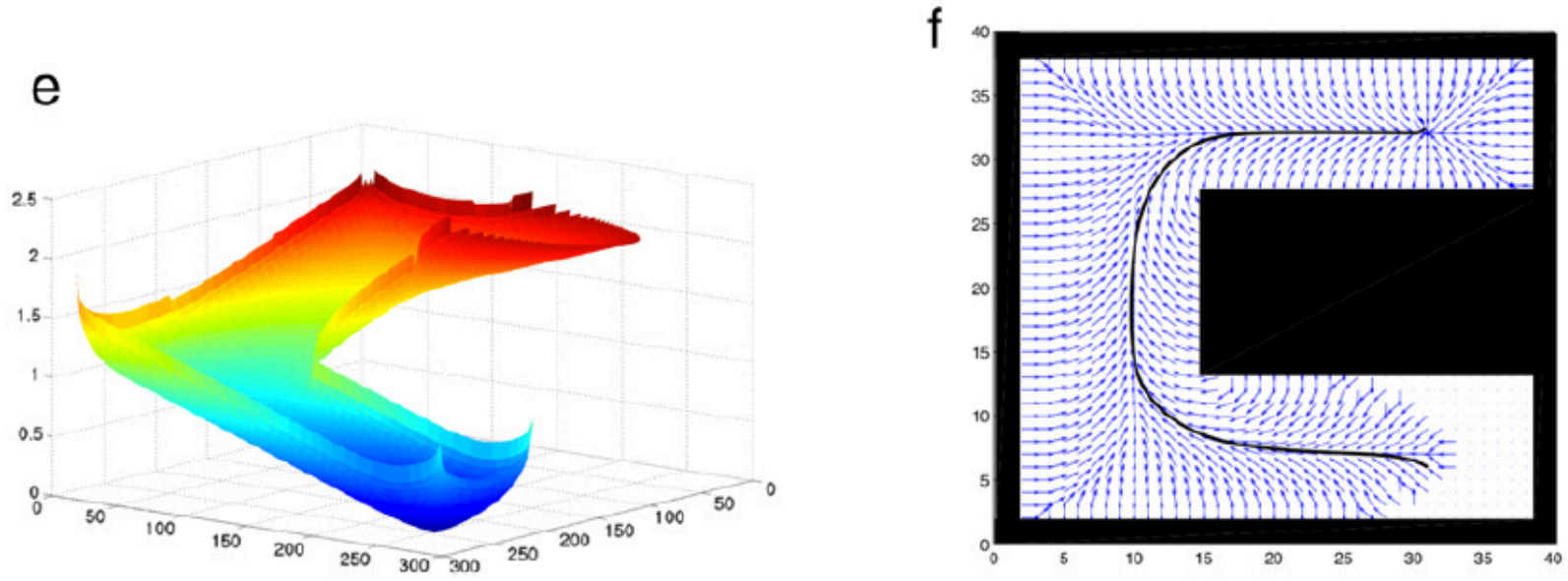


Figure: Time arrival map to each cell (c), maximum gradient directions map and the shortest time path following the maximum gradient at each point(d).

Fast Marching Square FM^2

Path

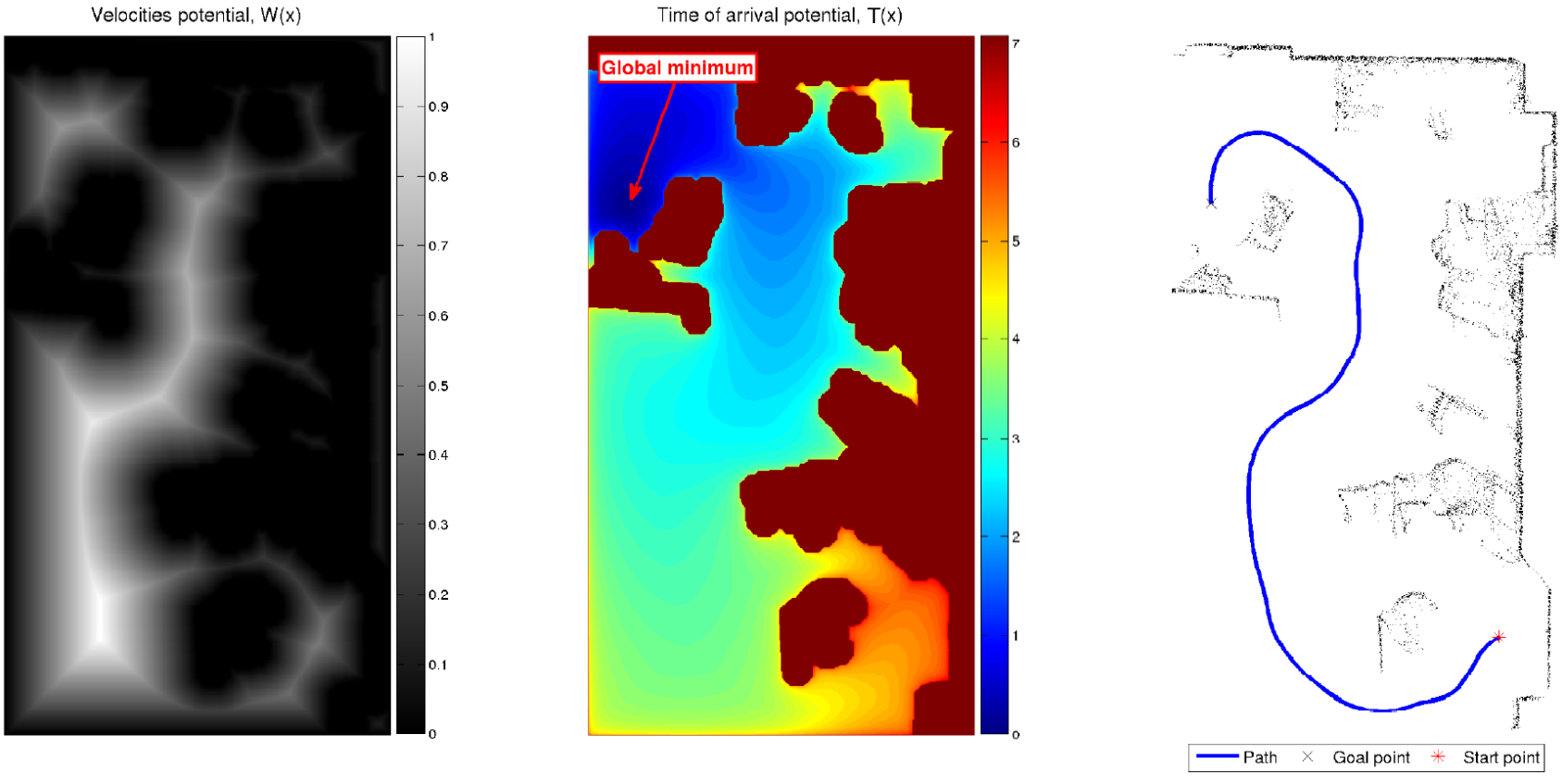


Figure: Steps of the FM^2 applied to path planning.

Fast Marching Square FM^2

Velocity profile

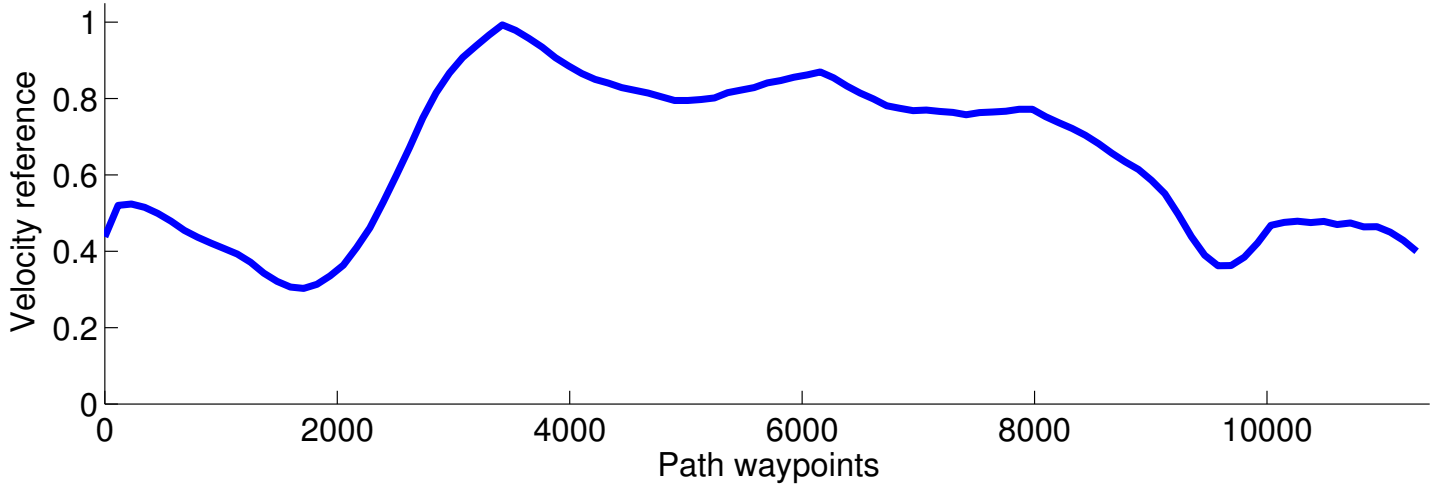


Figure: Velocity profile of the path shown in previous figure.

Fast Marching Square FM^2

Comments

- The FM^2 method is analogous to the Geometric Optics where light rays (trajectory in FM^2) travel in curved trajectories in media with changing refraction index (velocities map).
- Therefore, time optimality is justified by the principle of Fermat: *Light travels the path which takes least time*.
- As a final remark, the FM^2 does not require to dilate the obstacles, since it is going to compute a very safe path for the robot.
- However, the binary map could be dilated before computing the velocities map.
- But, .. is not completely logical

Saturated Fast Marching Square Method

- In most scenarios the trajectories provided by FM^2 are neither logical nor optimal, even though the quality of the trajectory in terms of smoothness and safety is always good.
- The FM^2 computed trajectory tries to keep the trajectory as far as possible from obstacles.
- This computed trajectory is similar to the path computed with the Voronoi diagram. But there are environments in which there is no need to follow a trajectory so far away from obstacles, as distance may be safe enough to navigate.
- To solve this a saturated variation of the velocities map can be used. The scaling of the map is made according to two configuration parameters:
  - Maximum allowed speed, which is the maximum control speed the robot may receive.
  - Safe distance, which is the distance from the closest obstacle at which the maximum speed can be reached.

Saturated Fast Marching Square Method

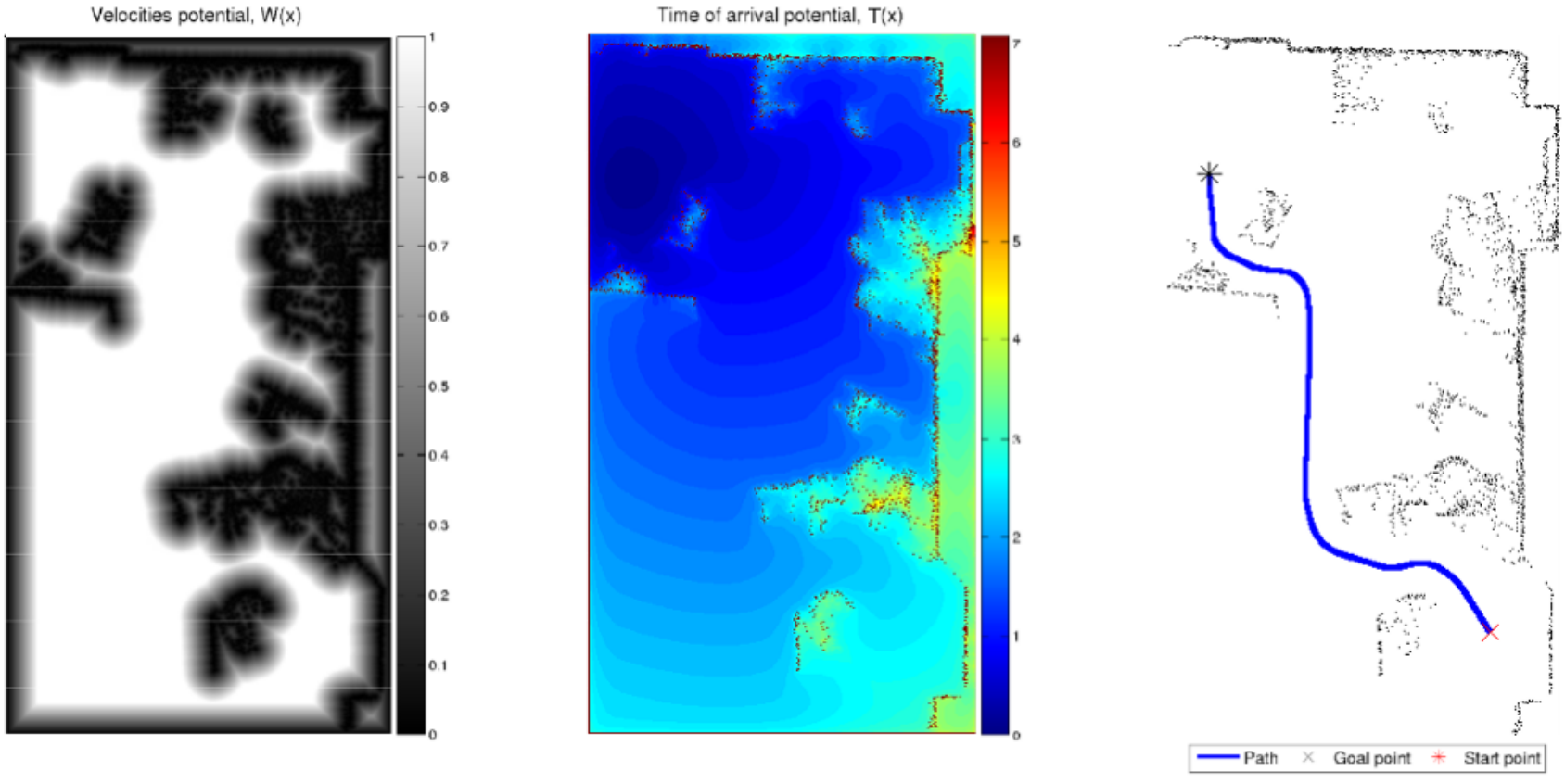


Figure: Results of the saturated version of FM^2: Saturation level: 0.3.

Saturated Fast Marching Square Method

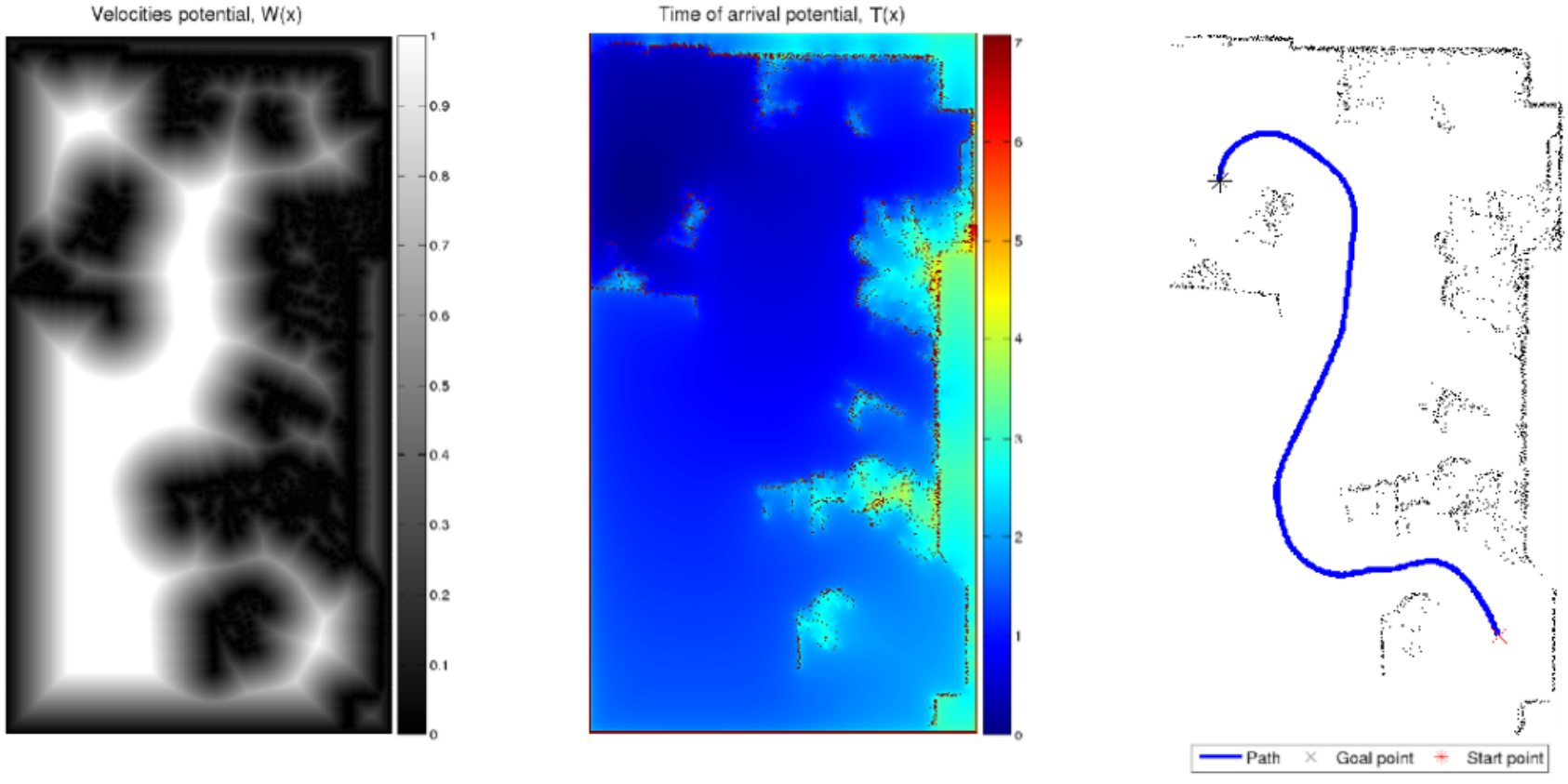


Figure: Results of the saturated version of FM^2: Saturation level: 0.6.

Saturated Fast Marching Square Method

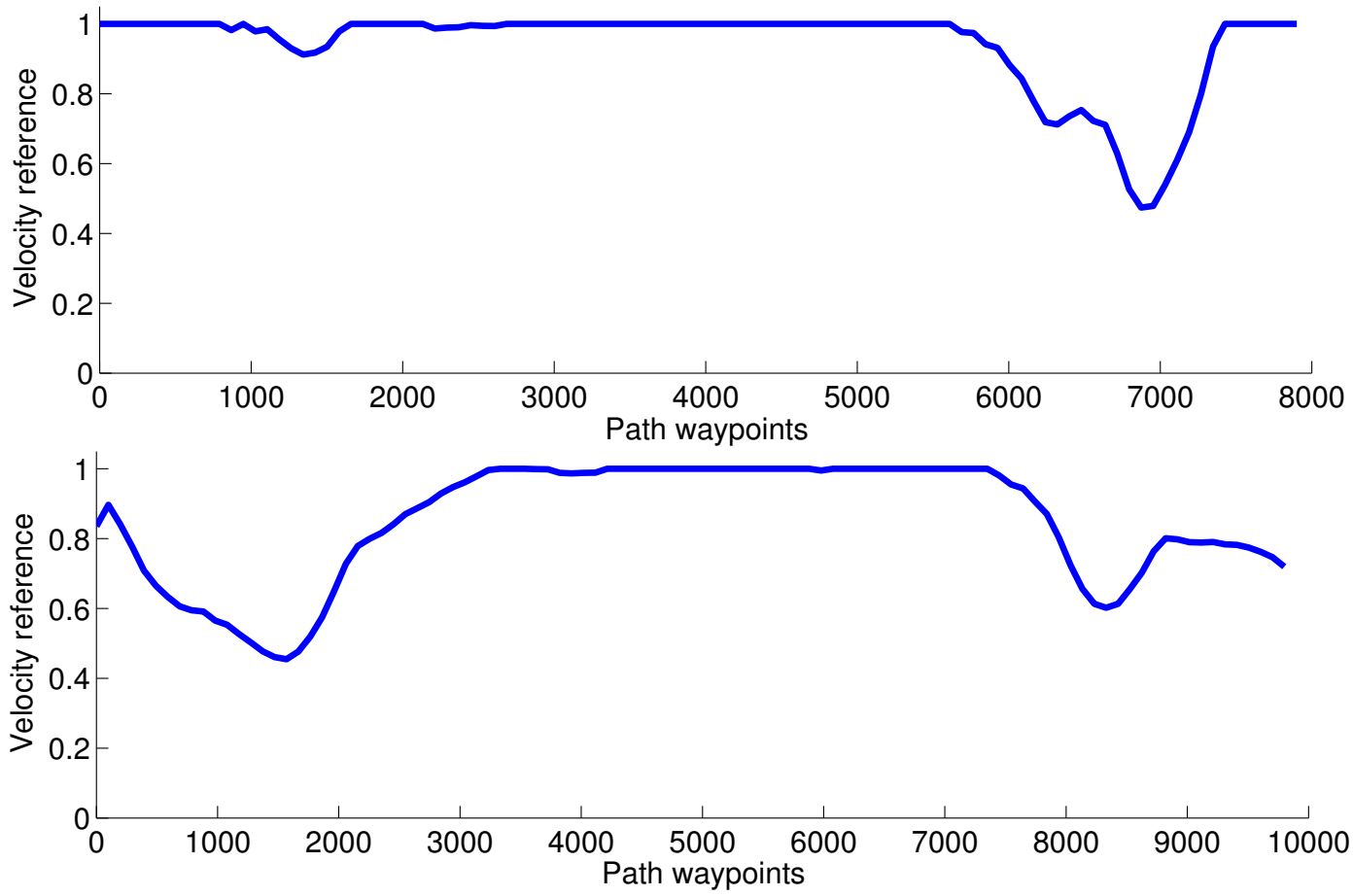


Figure: Velocity profiles of the paths: saturation levels of 0.3 and 0.6

Fast Marching Method

Application to 2 -1/2 D path planning

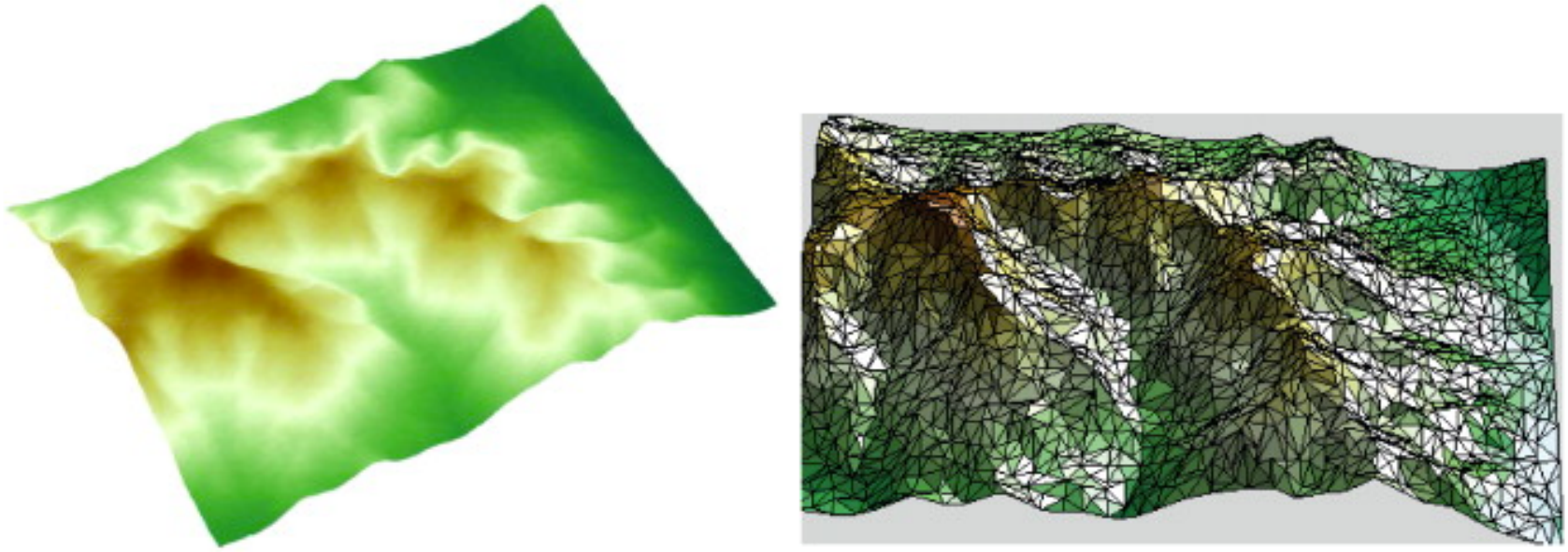


Figure: Elevation map and triangulation

Cost parameters: A elevation, G gradient and Sv spherical variance.

Fast Marching Method

Application to 2 -1/2 D path planning

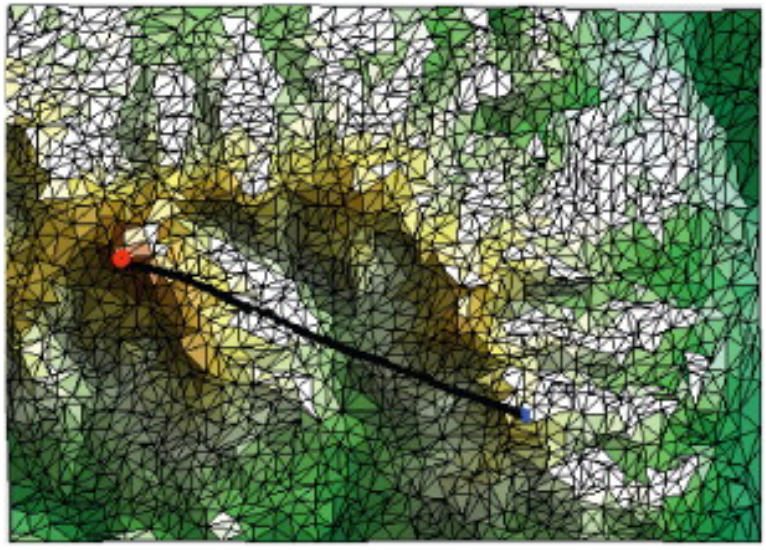


Fig. 7. Geodesic calculated without using matrix *W*.

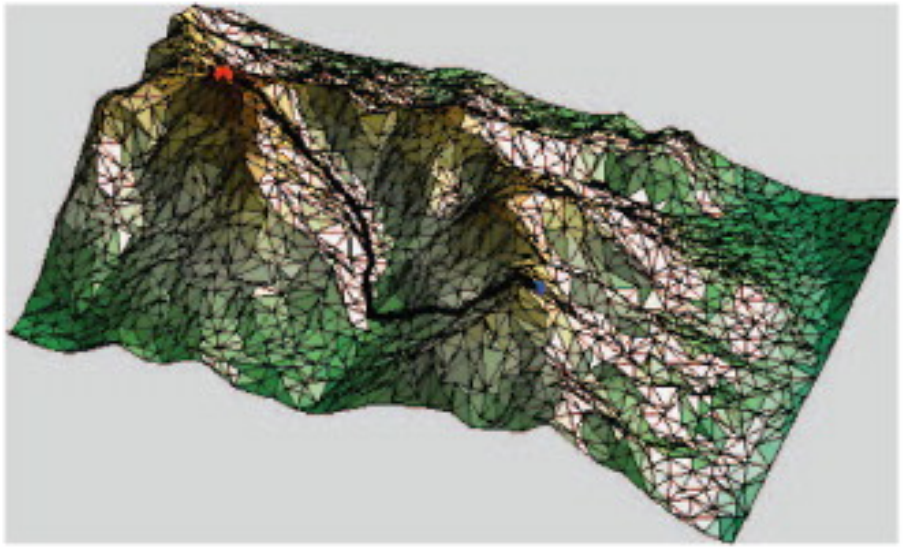


Fig. 9. Path calculated when *W*=*A*.

Figure: Shortest geodesical path

Fast Marching Method

Application to 2 -1/2 D path planning

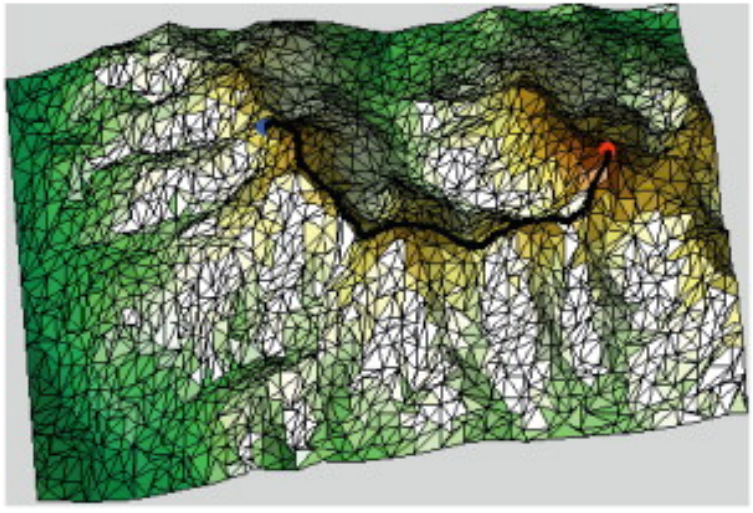


Fig. 10. Path calculated when $W=G$.

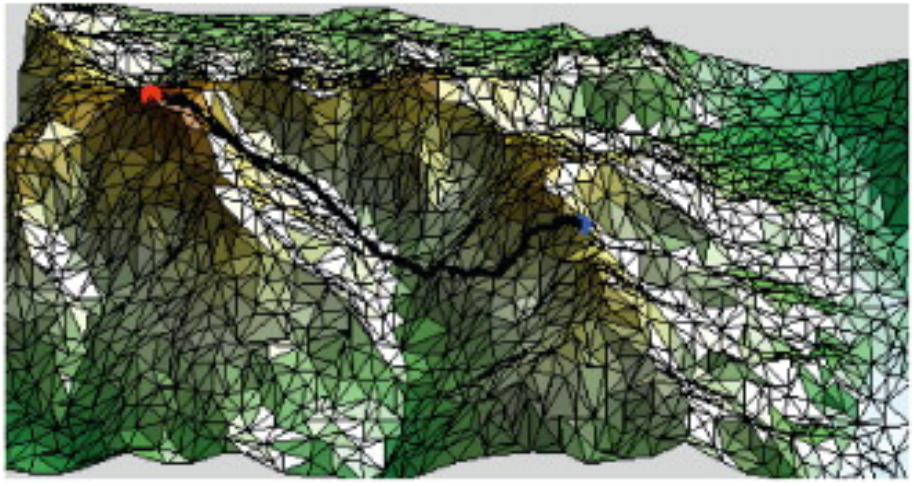


Fig. 11. Path calculated when $W=Sv$.

Figure: Other paths according different cost criteria

Fast Marching Method

Application to 2 -1/2 D path planning

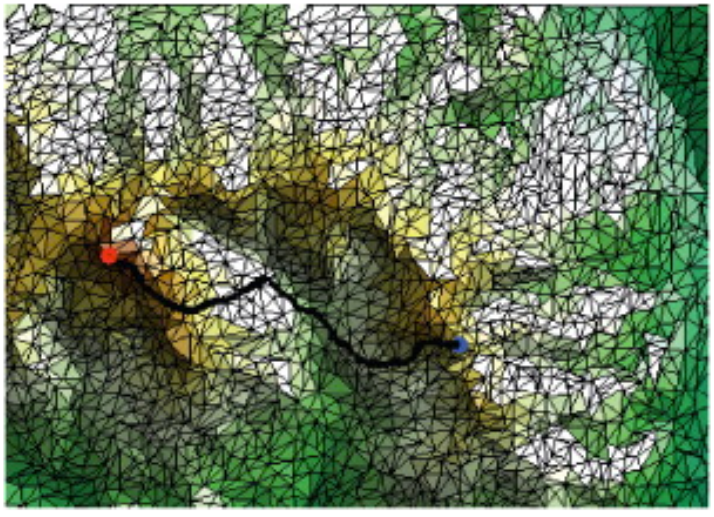


Fig. 12. Path calculated using W=0.15∗A+0.05∗Sv+0.85∗G.

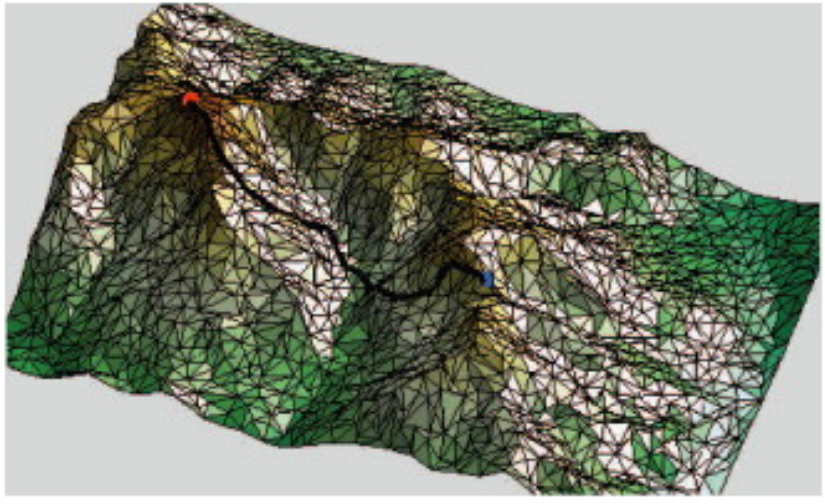


Fig. 13. Path calculated using W=0.15∗A+0.05∗Sv+0.85∗G

Figure: Other cost criteria

Applications

- 3D planning
- 2D / 3D robot formations
- UAVs
- Path learning

Videos –>

Fast Marching Square

Limitations

- Fast Marching square is basically a fast marching method that uses a scalar field as a speed limit or as a difficulty to the motion. That originates a different propagation speed at each cell.
- In spite of each cell has a different propagation speed it is assumed the isotropy of the medium at each cell, that means the propagation speed at each cell is identical for all motion directions.
- However there exists situations where this isotropy is neither valid nor convenient.
- Solution —> introduce anisotropy —> FMVF

- Fast Marching Method subject to a Vectorial Field (FMVF)

Fast Marching Method subject to a Vectorial Field (FMVF)

- The general idea is determine the minimum time path through an environment subject to two different functions affecting the motion: an escalar function (the viscosity) and a vectorial function.
- Each cell in the map is affected by a cost function of the form:

$$f_t = f_s + f_v$$

where:

  - The first term f_s is the scalar function that represents the cost function due to the difficulty of the terrain and obstacles , we usually compute it as $f = 1 - W$ being W a cost matrix.
  - The second part corresponds to an external vectorial field f_v. The vectorial field has associated, for each cell, a vector, that means a magnitude a direction.

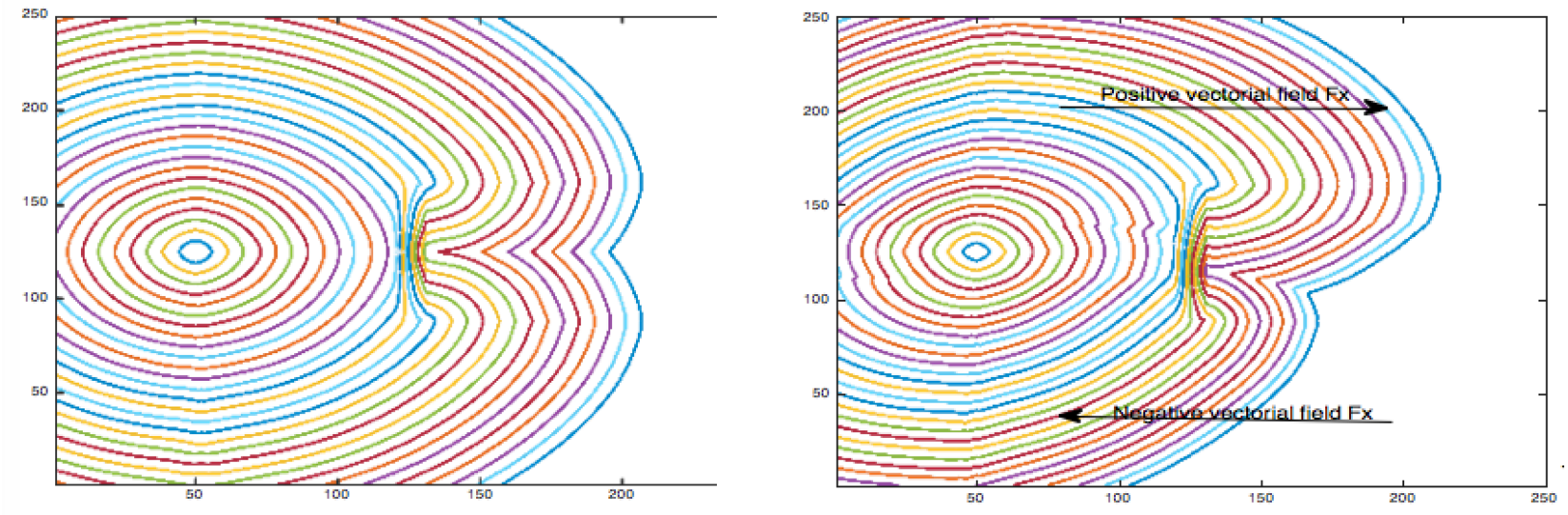
Fast Marching Method subject to a Vectorial Field (FMVF)

Anysotropy

- The use of two combined fields a directionally isotropic one (the scalar field) and another directionally anisotropic (the vectorial field) let us to cover a wide range of applications.
- Some clear areas of application are vehicles subject to directional difficulties: currents (sliding, wind or water currents)

Propagation differences

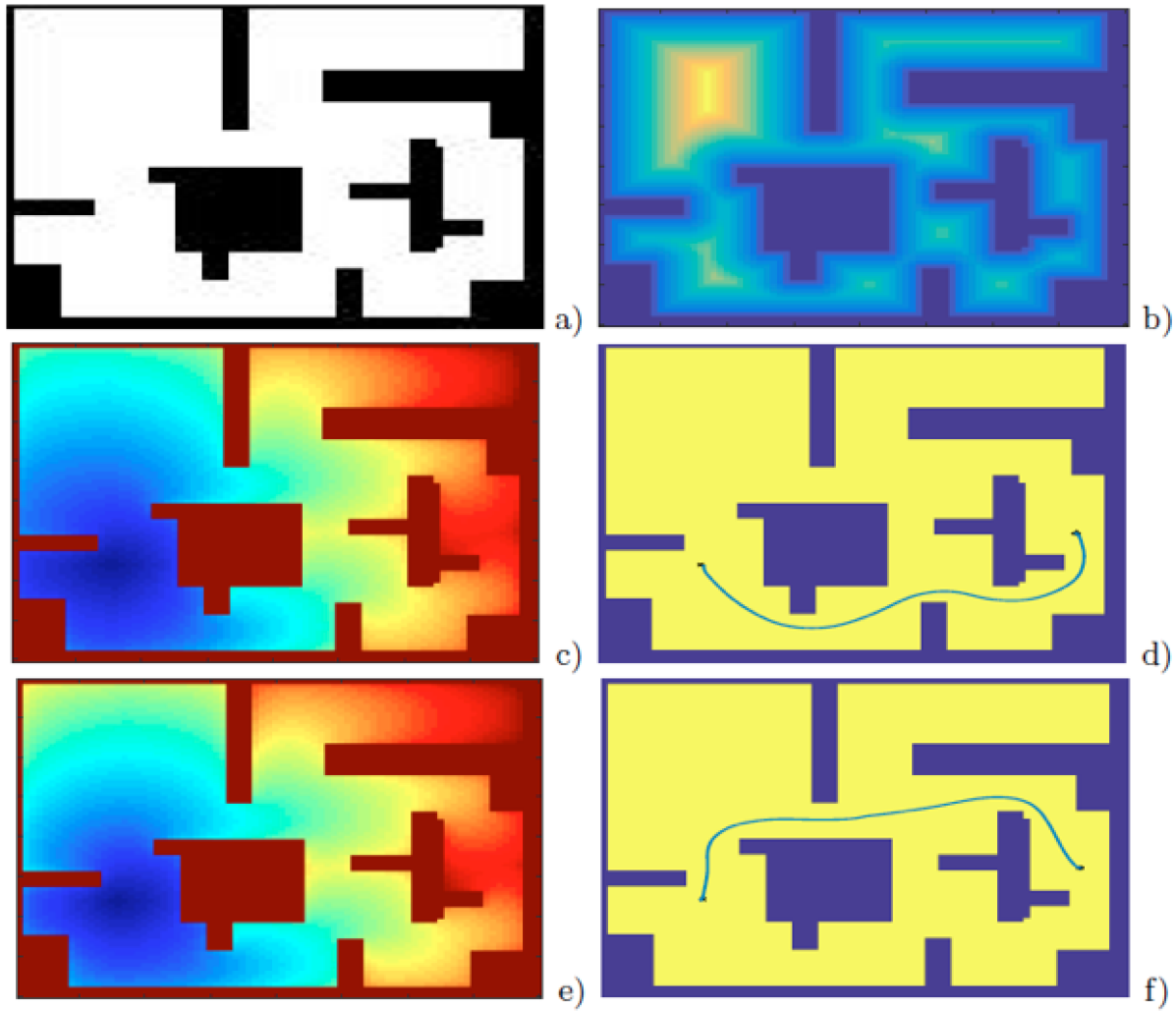
Scalar field vs Scalar plus Vectorial fields



Snapshot of the front propagation moving subject to: a scalar field (left) and a scalar plus a vectorial field (right), in the center is located a rectangular obstacle

FMFV

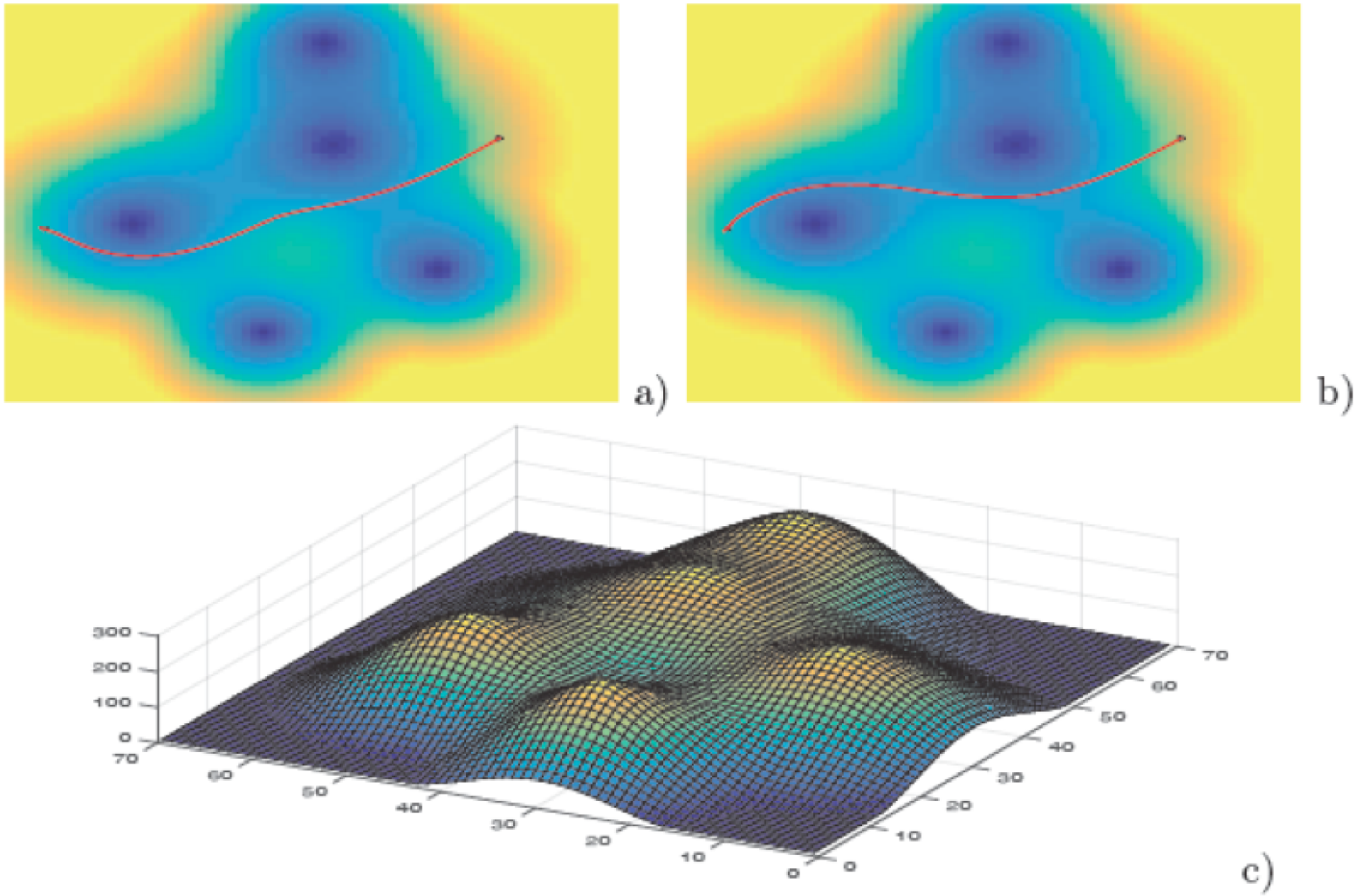
Example



Path calculated using FMM when there is an external vectorial field.
a) Original binary map of the room. b) Velocities map taking into account the distance to obstacles.
c) and e) Wave expansion (arrival time) with an external field that goes downwards or upwards respectively.
d) and f) Path extracted from the wave expansion.

FMFV

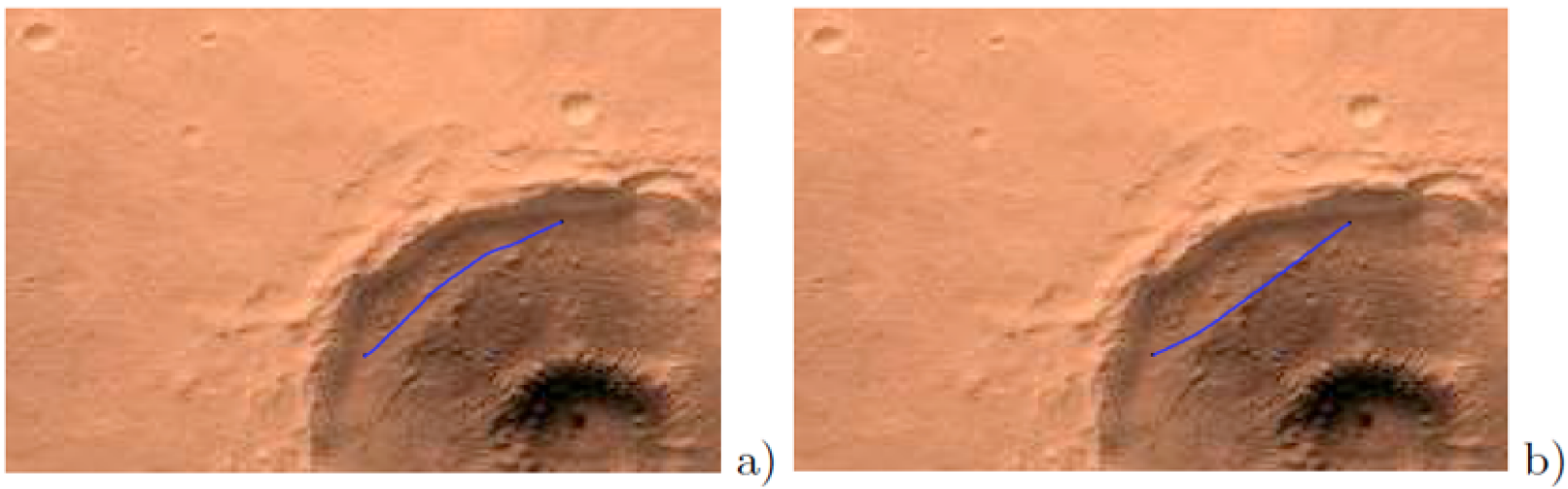
Example 2



Path calculated penalising the change in height, when there is an external field a) downside, b) upside and c) original map.

FMFV

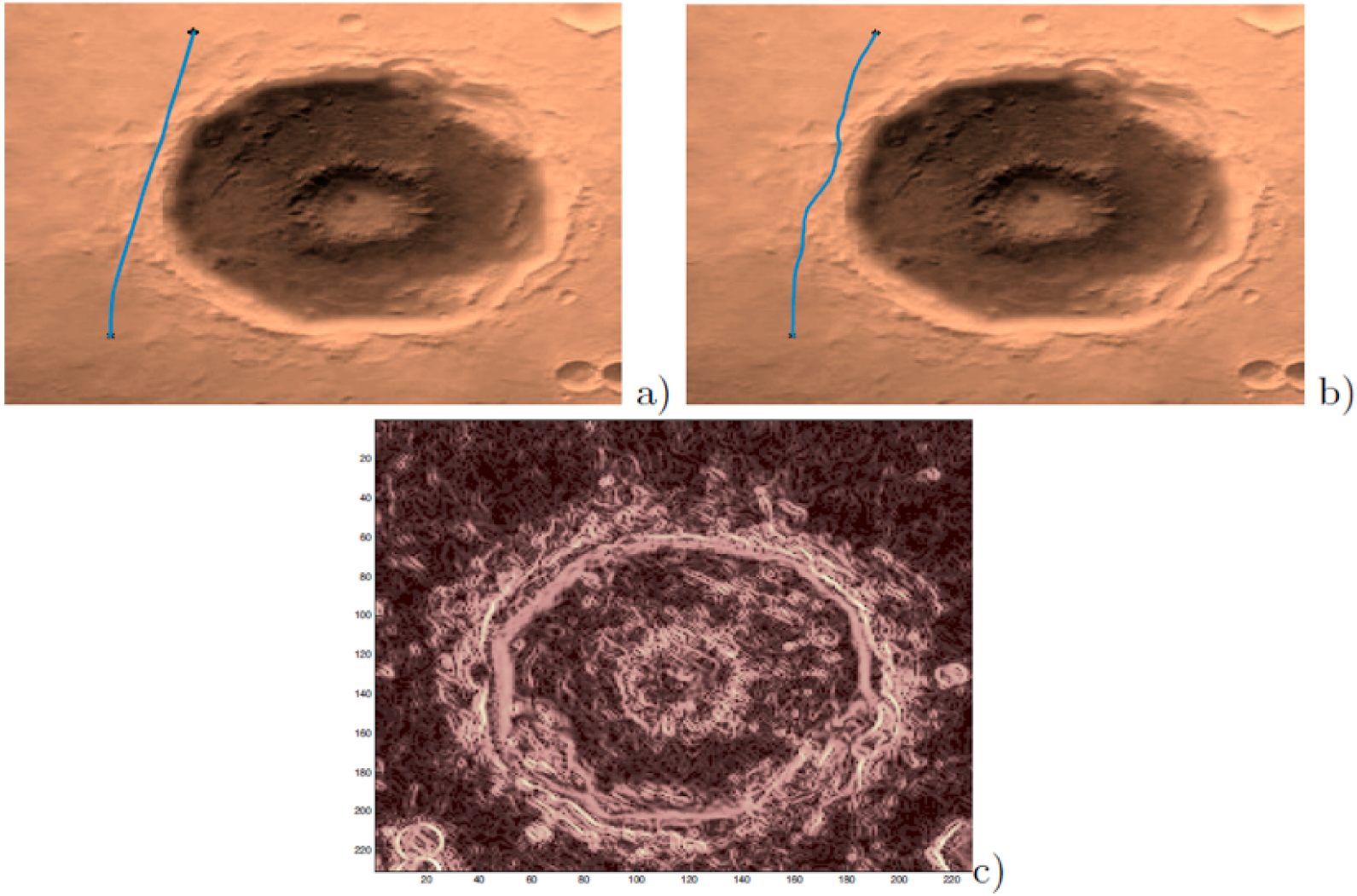
Example 3



Two different paths in the Mars Gale crater taking into account the lateral sand landslide proportional to the gradient. a) without landslide, b) with landslide.

FMFV

Example 4



Path calculated using mountains map penalising the change in height, when there is an external field a) downside, b) upside and c) original map.

Discussion

Advantages of FM

- Optimality.
- Smoothness of the paths (improves the executability).
- Computational cost is excellent for 2/3D case
- Flexibility

Discussion

Disadvantages of FM

- Extension to higher number of dimensions is difficult.
- Extensive evaluation of discretized space.
- High computational cost.

Discussion

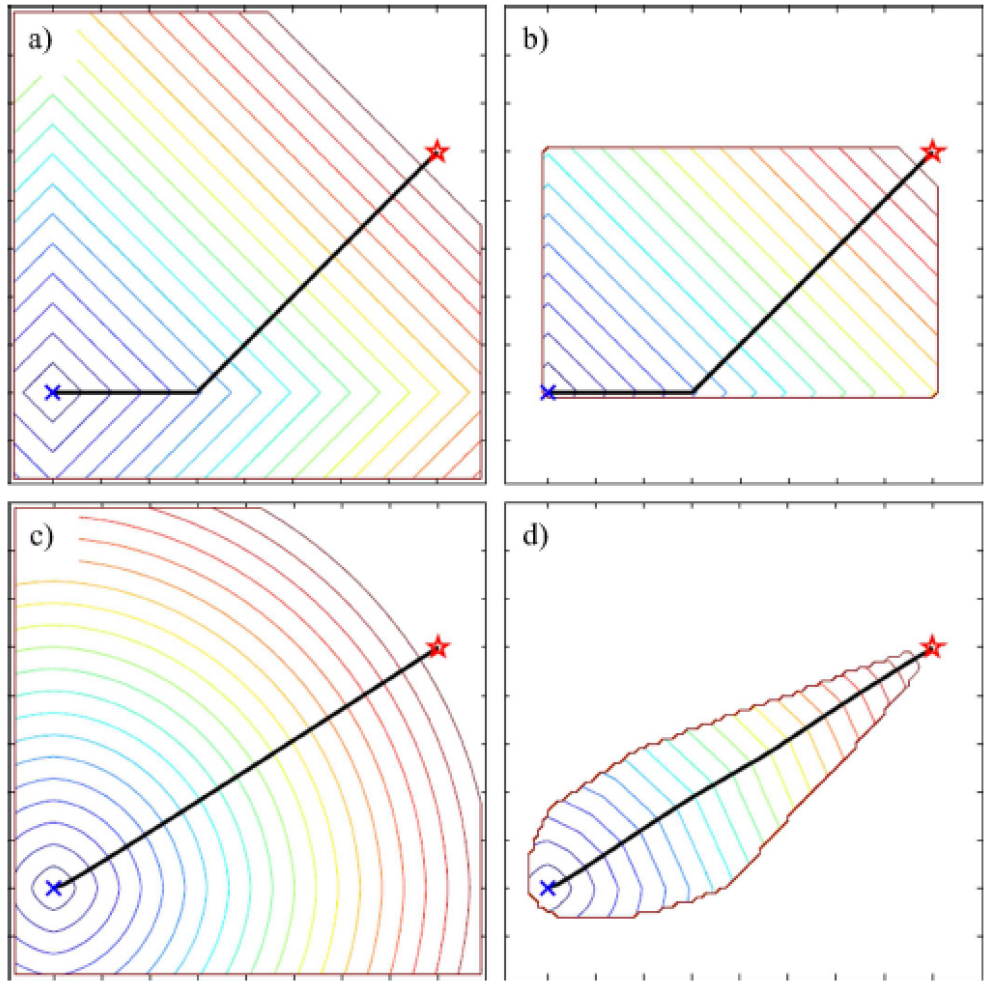
Future works, open problems

- Extension to high dimensionality problems. Ongoing.
- Application to Artificial Hand motion planning (learning and planning). Ongoing.
- Use of level set methods (that admits positive and negative speeds but requires a much more complex treatment). Ongoing.
- Using uncertainty in the refraction maps to introduce or solve probabilistic problems. Ongoing.
- Extension to cope vector speed fields . The speed in a point depends not only of the position but also of the robot pose. Ongoing.

- Questions?

Additional

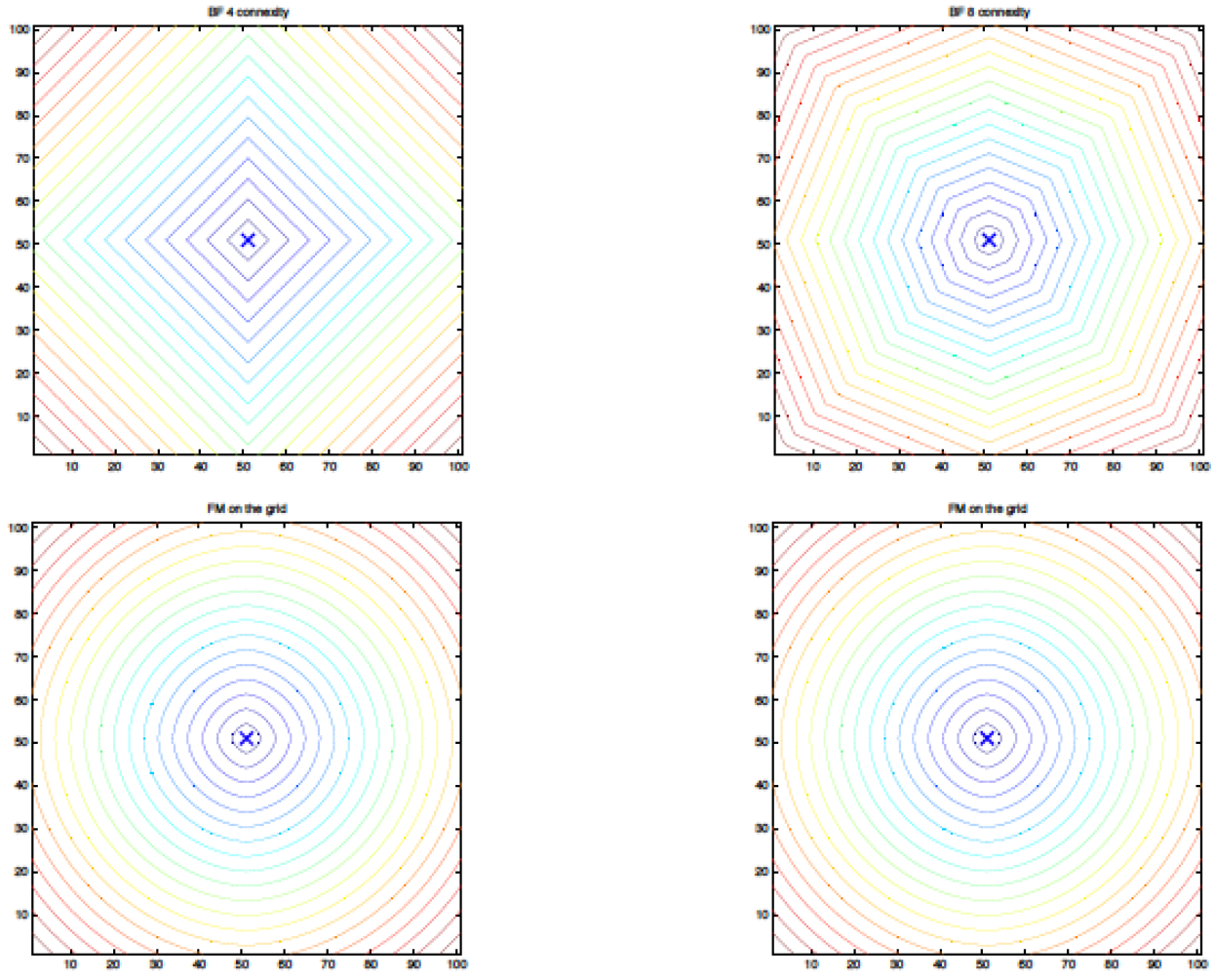
(1)



Examples of distance maps and trajectories computed over a constant 100x100 cost map ($\tau = 1$) using a 4-connectivity: a) BF, b) A* (using the "Manhattan distance" h4 as an heuristic), c) FM and d) FM* (using the "Euclidean distance" he as an heuristic) algorithms.

- Difference between Dijkstra and FM.

Additional
(2)



Examples of distance maps computed over a constant cost map using a 4- and 8-connectivity using Dijkstra and a 4- and 8-connectivity using FM.

- Difference between Dijkstra and FM.