2014 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
May 14-15, Espinho, Portugal

Towards efficient path planning of a mobile robot on rough terrain *

Diogo Amorim
Institute for Systems and Robotics
Instituto Superior Técnico, Universidade de Lisboa
Av Rovisco Pais, 1
1049-001 Lisbon, Portugal
Email: diogo.amorim@ist.utl.pt

Rodrigo Ventura
Institute for Systems and Robotics
Instituto Superior Técnico, Universidade de Lisboa
Av Rovisco Pais, 1
1049-001 Lisbon, Portugal
Email: rodrigo.ventura@isr.ist.utl.pt

***Abstract*—Most path planning methods for mobile robots divide the environment in two areas – free and occupied – and restrict the path to lie entirely within the free space. However, the problem of path planning in rough terrain for a field robot, e.g. tracked wheel, is still a challenging problem, for which those methods cannot be directly applied. This paper addresses the problem of path planning on rough terrains, where the local properties of the environment are used to both constrain and optimize the resulting path. Finding both the feasibility and the cost of the robot crossing the terrain at a given point is cast as an optimization problem. Intuitively, this problem models dropping the robot at a given location and determining the minimal potential energy attitude. Then, a Fast Marching Method algorithm is used to obtain a potential field free of local minima. This field is then used to either pre-compute a complete trajectory or to control in real time the locomotion of the robot. Preliminary results are presented, showing feasible paths over an elevation map of a rough terrain.**

***Keywords*—Path planning, fast marching method, rough terrain, robots**

* This work was supported by the FCT project $[PEst-OE/EEI/LA0009/2013]$

I. Introduction

This paper proposes a method to efficiently plan a path for a mobile robot on rough terrain. There already are some path planing methods that solve this type of problem in 2D, however we intend to apply these tools to the same sort of problem, and with the same goal, but with a different premise. This premise is the type of map. This represents a rough surface with information about the free space and insuperable obstacles but also the elevation of each coordinate. The elevation variations may imply new obstacles for e.g. if a slope is too steep the vehicle will not be able to climb it. The purpose of this paper is to show how to create a map in a (x, y) configuration space where each coordinate is associated with a cost based on the vehicle's pose as if it were dropped on the floor at the given coordinates. To determine the pose of the vehicle we compute the minimal potential attitude at each location. The robot's attitude is the solution of an optimization problem that solves the scenario of dropping the robot on the surface at each point of the map. We can then apply a Fast Marching Method (FMM) [1], [2] to the newly created cost map to obtain a potential field with no local minima. This field is then used to guide the robot to its goal smoothly and safely away from obstacles or hazardous situations. The motivation of this work is the implementation of an autonomous navigation method on rough terrain on the search and rescue robot RAPOSA-NG[1] [3], a track wheel vehicle designed for urban search & rescue operations. It is then necessary to develop a path planning method for 3D environments capable of dealing with sharp edges (discontinuities in map data) such as steps or sharp debris, as could be the case when dealing with these environments.

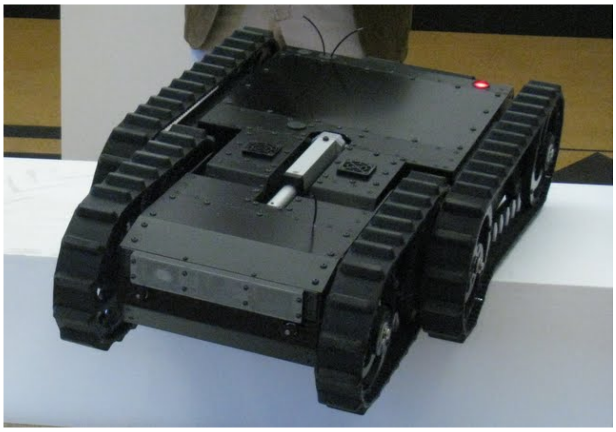

Fig. 1. The mobile robot Raposa NG

II. Related Work

Path planning is a widely studied problem that has been approached in many different ways over the years as literature demonstrates, see for example the B. Siciliano and O. Khatib, *Springer Handbook of Robotics* [4] and S. LaValle, *Planning Algorithms* [5] for extensive reviews. However, most of these methods assume a prior division of the environment between free and occupied space, while robot movement is constrained to the free space. Rough terrains, for which such a binary division is not trivial, often require an alternative approach. In [6] two path planning methods are compared to show their ability to solve the same problem in different ways with different degrees of satisfaction of the final resulting path. The first uses a genetic path planner which only requires an approximate description of the terrain and operates on

[1]http://raposa.isr.ist.utl.pt

2014 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
May 14-15, Espinho, Portugal

the basis of evolutionary process and stochastic search to generate a near optimal path. The second is a global planner which incorporates kinematics and dynamics of the robot, and requires more knowledge about the environment and the vehicle. In the work here presented we analyze the terrain in detail and are not worried with the specifications of the vehicle other than its general shape, contact points and center of mass. So we can overlook these methods as we are developing a path planning algorithm that generates the optimal path with detailed information about the map. One relevant work is described in a paper from S. Garrido and his team [7] that applies the fast marching method to outdoor motion planing on rough terrain. Despite the similarity in Garrido's work the terrain is locally approached by a plane, which may pose problems for discontinuous terrains, e.g. stairs or other discontinuities like debris. The premise is the same elevation map but instead of working directly with the data, in first place, Garrido and his team generate a Delaunay triangulation and then add the third coordinate (elevation) to the mesh. They obtain a triangulated surface and from that obtain three characteristics they call: Spherical variance, Saturated gradient and height. Our approach has the possibility of being much faster as it works directly with the elevation map data and it extracts both characteristics (pose and number of contact points) in a single sweep of the map.

III. Proposed Approach

When planning the path for a mobile robot one needs to ensure the path is safe and smooth all the way to the goal point. We can also require speed or efficiency depending on, for example, the mission energy budget. It is necessary to find a way of determining the robot's world characteristics in a fast manner in order to, then, address the path planning issue. The characteristics we have found relevant are the robot's pose at each point of the map and whether or not the contact with the ground is stable. Defining the problem as a constrained optimization problem with non-linear constraints it is possible to determine the pose of the mobile robot as well as the number of contact points with the ground, an important factor to determine whether or not it is possible for it to stand on that position. Our approach is based on a two-step process. Firstly, a cost map is obtained by computing a cost of moving at each point of a 2-D grid covering the environment. This cost is set to infinity if that point is unfeasible for the robot to cross. In this paper we set the cost to a function of the rotation of the vehicle with respect to its pose on a horizontal planar ground. Take for instance a vehicle on a ramp: the cost is zero if the ramp is horizontal, and it increases with the inclination of the ramp. To compute this deviation we consider the robot dropped vertically at the given position, and then we determine the robot attitude that minimizes its potential energy i.e., the one that minimizes the height of the center of mass of the vehicle. This problem is cast as an optimization problem which is numerically solved for each point of the grid. The second step corresponds to running a Fast Marching Method algorithm over this cost map, for a given goal position. The result is a potential field, with two fundamental properties, as far as path planning is concerned:

1) it shows no local minima
2) the gradient descent over the field is the optimal path to the goal, on given a cost map

A. *Computing the cost map*

The robot's world is defined in a (x, y, z) configuration and it is defined in the inertial frame of reference (*I*). The robot is defined as an n points $p_{j,R} = (x_{j,R}, y_{j,R}, z_{j,R})$ structure Fig. 2, all fixed to its reference frame, the robot's reference frame (*R*) where the origin is set at the center of mass of the robot.

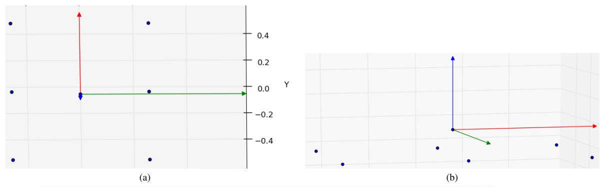

Fig. 2. Representation of the mobile vehicle and its reference frame. Fig. 2 *a* illustrates the top view of the representation of the robot and Fig. 2 *b* shows a perspective of the representation. In this case an approximation of the representation of the robot in Fig. 1 is shown.

The $n-1$ points characterize each contact point $(p_{1,R}, p_{2,R}), ..., (p_{n-1,R})$ and the last point $(p_{cm,R})$ represents the vehicle's center of mass, which, as previously mentioned, was described as the origin of the frame. The robot's pose is defined as :

$$(x_{cm,I}, y_{cm,I}, z_{cm,I}, \theta, \beta, \gamma) \quad (1)$$

where $(x_{cm,I}, y_{cm,I}, z_{cm,I})$ are the coordinates of the *R* in relation to *I*, and (θ, β, γ) are Euler angles $Z - Y - X$. The first, θ (yaw) is the angle of rotation of the *r* around the Z_R axis, β $(roll)$ the Y_R axis and γ $(pitch)$ the X_R axis. To determine the coordinates of the points defining the robot on the *I* their coordinates in *R* are multiplied by a rotation matrix and then added the position of the *R* relative to the *I*. The idea is, for every point of the map, to determine the pose with which the robot would adopt if it were to rest on the surface at that same point. The target is to minimize the z coordinate of the robot's center of mass, in relation to the *I* but with the restriction that none of the points that define the robot can pass through the surface. The vehicle also has pose limitations, it cannot be upside down, cannot climb hills steeper than γ_{max} or roll more than β_{max} [2] so its roll and pitch angle absolute maximums were set at that same value. These limitations are translated as constrictions when inserted in an optimization problem, and so, in order to solve this specific problem one can resort to a constrained (multivariate) problem solving routine.
In order to introduce constraints in the optimization function it is necessary to formulate them as inequalities, functions whose values are always positive which in this case means, for example, that the z coordinate of each point defining the robot minus the z coordinate of the point of the map directly below must be positive, and this condition is respected by the algorithm. There are $n+2$ constraints to this simple problem, one per each point that defines the robot, one for the roll and another for pitch angle limitations. More constrictions can and will be added in order to better simulate the hull of the vehicle more accurately depicting it. The task of determining the robot's pose can then be defined as a constrained optimization problem in the following way:

[2] the value stipulated for the simulation was $\gamma_{max} = \beta_{max} = 45°$

$$\begin{array}{ll} \text{Minimize:} & z_{cm,I} \\ \text{Variables:} & z_{cm,I},\ \beta,\ \gamma \\ \text{Subject to:} & \forall_j (z_{j,I} - map_{x_{j,I},y_{j,I}} \geq 0) \\ & \frac{\pi}{4} - |\beta| \geq 0 \\ & \frac{\pi}{4} - |\gamma| \geq 0 \end{array} \quad (2)$$

The optimization function minimizes the value of $z_{cm,I}$ by manipulation of the three variables it has access to: ($z_{cm,I}, \beta$ and γ). This means the closer the position of the robot before the optimization is to its final position after the optimization problem is solved, the faster the minimization is, as less iterations are needed to determine the minimum $z_{cm,I}$. To improve the computational performance of the numerical optimization we provide a warm start obtained in the following way. It is determined by approximating a plane to the $n-1$ points that are the projection of the points defining the vehicle, on the surface. In this way we obtain an approximation of the pose. The next step is to rotate the robot's points to the approximated pose and then translate vertically to the hight where only one point touches the surface, usually the point that is above the highest point of the surface bellow it. The constrictions determine that the robot's absolute pitch (γ) and roll (β) angles don't reach values greater than γ_{max} and β_{max}. Another limitation is that none of the z coordinates of the points defining the robot ($z_{j,I}$) can be lower than the elevation of the map directly below ($map_{x_{j,I},y_{j,I}}$) thus $z_{j,I} - map_{x_{j,I},y_{j,I}}$ must be grater than 0. The results are as expected, the robot touches the ground with three or more of the n points depending on the surface roughness. All those are valid positions, but if the function returns that only two or less points are touching the surface that means that the vehicle cannot be on that position because of pose limitations introduced as constraints and that same position on the map is considered an obstacle. It is now possible to build a cost map where we use the pose angles (pitch and roll) provided by the previous routine to determine the horizontal deviation of the robot's plane . For every cell of the map, i.e., every time the optimization function returns a pose, this value is processed in the following way: we create a normalized vector v_1 with the coordinate $x = 0$ which makes a β degree angle with the Y axis and normalized vector v_2 with the coordinate $y = 0$ which makes a γ degree angle with the X axis. The vectors are added and the resulting vector's angle with the Z axis is stored in the equivalent cell of the cost map. If we were to keep the absolute value of the resulting angle, we would be admitting the cost of moving uphill or downhill on a slope with equal inclination is the same. It is intuitive to say the robot will struggle more going uphill then going downhill and this is why we kept the information about the sign of γ. We now have the information about where the map, is up or downhill and how inclined the surface is, we have also determined obstacles to the passage of the robot.

B. Path Planning

The path planning determines the best path, in the given conditions, between the position of the robot and the goal point. Rather than determining an explicit path we create a potential field and then we are able to draw a path to the goal from any position on the map, simply by following the negative gradient of the field. The potential field cannot have local minima and ensures the optimal path to the goal position avoiding obstacles or difficult patches of terrain. This field is obtained by considering, for each point x within the free region $\Omega \subset \mathbf{R}^2$ of the map, the minimal time it takes a wave to propagate from the goal location to the current position. The computation of this time for each point x in the free region Ω results in a field $u(x)$. It is well known that the path resulting from solving the ODE $\dot{x} = -\nabla u(x)$ from an initial $x(0) = x_0$ results in the optimal path from x_0 to the initial wave front. The wave front $\Gamma \subset \Omega$ is set around the goal point. The propagation of a wave, given an initial wave front $\Gamma \subset \Omega$, can be modeled by the Eikonal equation

$$|\nabla u(x)| = F(x) u(\Gamma) = 0 \quad (3)$$

where $x \in \Omega$ is the free space of robot position, $\Gamma \subset \Omega$ the initial level set, and $F(x)$ is a cost function. This cost function allows the specification, in an anisotropical way, that is, in a directionally independent way, the speed of the wave propagation. In particular, for a point x, the wave propagation speed is $1/F(x)$. This cost allows the resulting path to maintain a certain clearance to the mapped obstacles, since the optimal path tends to keep away from areas with higher costs. To solve this equation we use the FMM algorithm, introduced by J. A. Sethian [8], given a discretization of the map in a grid, we supply to this method the region of free space Γ, the cost function $F(x)$, and the goal point, and in return we obtain a numerical approximation to the solution of the Eikonal equation on the grid points. The cost function $F(x)$ is obtained as described in the previous subsection. The FMM is a numerical algorithm that approximates the viscosity solution of the Eikonal Eq.3 which describes the moving boundary of a disturbance and can describe light propagation in a non-homogeneous medium. It is related with Maxwell's equations of electromagnetics, and provides a link between physical (wave) optics and geometric (ray) optics. Constant values of the Eikonal equation represent surfaces of constant phase, i.e. wave fronts and the normals to these surfaces are rays. The equation provides a mean of tracing a ray in a medium of varying index of refraction. The FMM is designed for problems in which the speed function never changes sign meaning there are no reflections, diffractions or interferences. Inputing the cost function previously created as a speed function in the FMM we can now calculate the propagation time for every cell of the map. The gradient vector field of the resulting time function can be compared to a energy potential field. It is lowest at the origin of the propagating wave (the goal) and highest at the point of the map most difficult to pass through or stand on, by the robot.

IV. Preliminary Results

Every result presented in this paper was obtained with the use of a single process Python algorithm, run on an Intel® Core™ i7-2630QM CPU @ 2.00 GHz processor. The simulation is run on three test scenarios i) a representation of a mountain area generated by software, the surface is rendered from a gray scale image Fig.3 that represents an elevation map, on ii) an image of irregular polygons at different heights emulating debris and iii) a RoboCup Robot Rescue arena[3]. The

[3]This elevation map is derived from running octomap in simulation over the RoboCup Robot Rescue arena, http://www.isd.mel.nist.gov/projects/USAR/arenas.htm

2014 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
May 14-15, Espinho, Portugal

resulting surfaces are a representation of a rough terrain area as shown in Fig.4 and a debris like structure in Fig.6. The maps can be scale up/down for our convenience, so that the scale of the robot is not too small. The first map is 248x248 in a total of 61504 of cells to be processed, the second is 491x491 in a total of 241081 cells and the last has a total of 117572 cells.

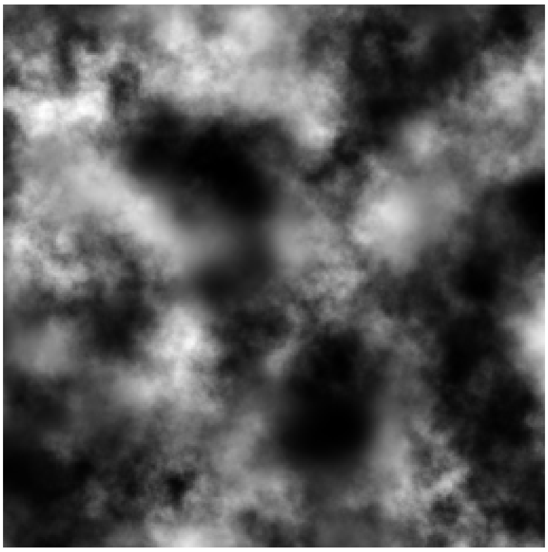

Fig. 3. Gray scale image from where the testing surface was rendered. It is an elevation map generated by the Terragen software available for anyone to use

The simulated vehicle, a raw approximation to the tracked wheel robot RAPOSA-NG Fig.1, is defined by 6 contact points with the ground, 3 per each simulated track (one at its beginning another at the end and one at the middle point), and one other point that defines its center of mass as seen in Fig.2. More points can easily be added, for better accuracy, however at the expense of a larger computational burden.
The process explained before was applied to the already mentioned surfaces Fig.4 and 6. The optimization function used is available from the scipy package from python, it uses the **SLSQP** (**S**equential **L**east **SQ**uares **P**rogramming) method to minimize our function of a single variable $(z_{cm,I})$ with a combination of bounds and inequality constraints (Eq.2), originally implemented by Dieter Kraft [9]. Processing the surface from Fig.4 we obtain the equivalent to a energy potential information that can be depicted as seen in Fig.5. Fig.4 shows the surface as it exists, in shape, but it is scaled down so that the highest peak is not bigger than the robot's length. Fig.5 shows the same physical space as the later but it is already processed to allow path planning to occur. Notice that, the surface is as if tilted to the right, this is because the goal was set on the right side of the map and it is the point with the lowest energy potential. The represented peaks are softer than the Fig.4 and possibly not even in the same place because they do not represent elevation but a higher difficulty for the autonomous vehicle. The same reasoning applies to the maps in Fig.6 and 8 and resulting energy potential in Fig.7 and 9 respectfully. The highest peaks seen in Fig.7 are representing obstacles, points in space where the robot is incapable to travel trough. The cost at those points is so high (the peaks are scaled down to better visualize the results), the path will never include them. Fig.8 represents a complex environment and with a variety of obstacles as walls and steps. Taking the world's representation as in Fig.5, 7 or 9 the optimal path to the goal corresponds to the gradient descent from a given initial position.

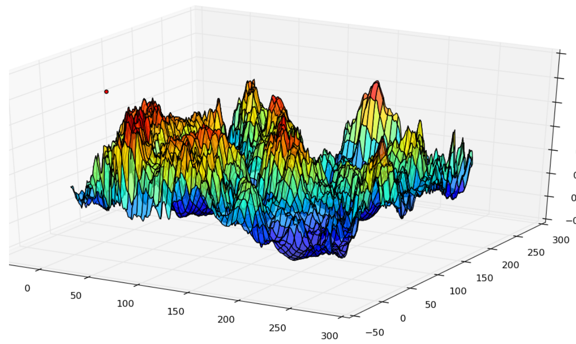

Fig. 4. Rendering of the gray scale elevation map

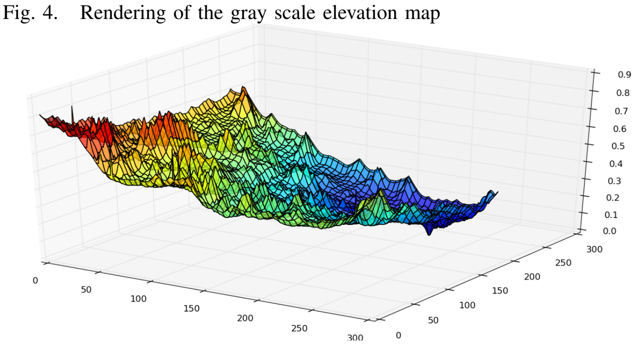

Fig. 5. Energy potential representation of the processed Fig.4 with the goal set at $(220, 215)$

From the representation of the potential we run the FMM to obtain the optimal path, for the given cost map Fig.10 and Fig.11. The path represented in Fig.10 was obtained from the representation of the world depicted in Fig.5 and Fig.11 from Fig.7. The maps show isochrone lines, which draw same travel time distances to the goal, that represent same travel cost to the goal from the lowest cost in dark blue to the highest in red. The path planning algorithm chooses the minimal total cost route based on the gradient of the lines. With this map representation we can easily obtain the robot's orientation and speed making it theoretically easy to develop a controller for the vehicle. The map in Fig.4 takes about 0.007 seconds/cell to be processed (extracting the world characteristics) but there is still room for speed improvement as we are still developing concepts. To compute the entire path shown in Fig.10 it takes 0.0066 s, which means that after processing the environment the vehicle can draw its own path in real time. The other scenario, depicted in Fig.6 is processed in 0.0045 seconds/cell and the computations of the path planning represented in Fig.11 took a total of 0.17 seconds.

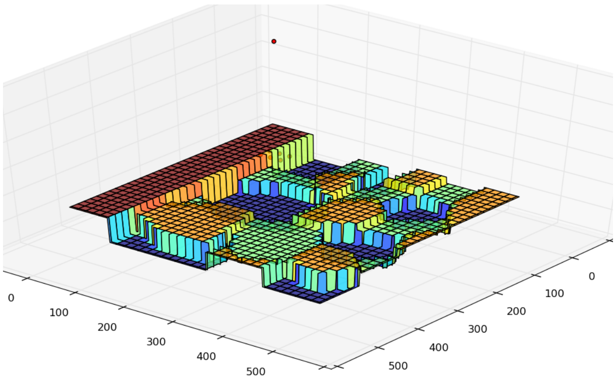

Fig. 6. Rendering of an environment simulating debris, which includes discontinuities

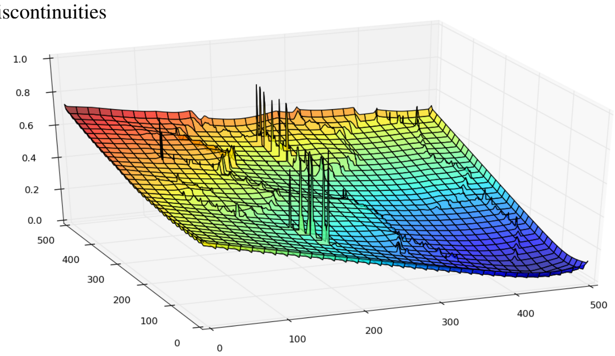

Fig. 7. Energy potential representation of the processed Fig.6 with the goal set at $(450, 50)$

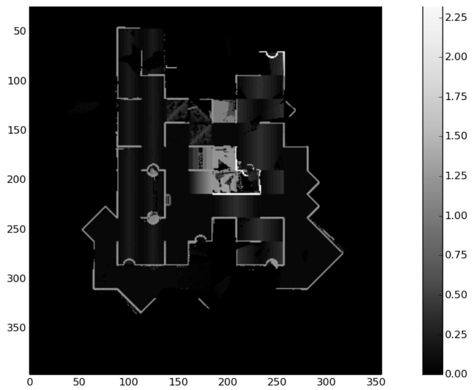

Fig. 8. Grey scale image of an elevation map of a real environment from NIST

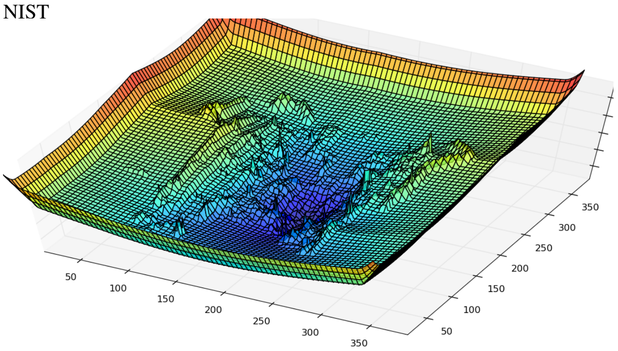

Fig. 9. Energy potential representation of the processed Fig.8 with the goal set at $(184, 203)$ which is the top of a simulated flight of stairs

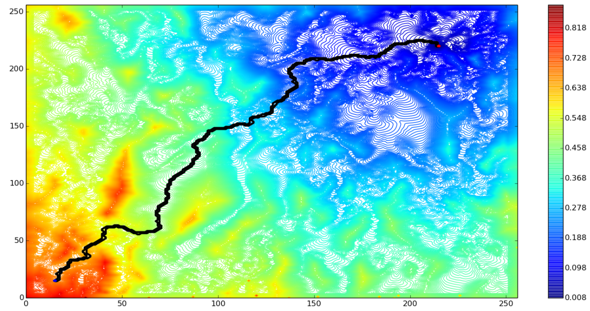

Fig. 10. Path planning over the representation of the isochrone lines of the processed surface represented in Fig.4

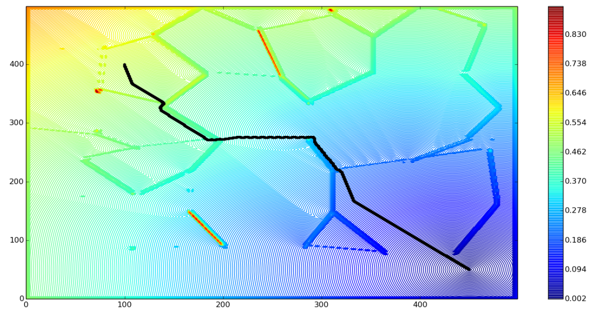

Fig. 11. Path planning over the representation of the isochrone lines of the processed surface represented in Fig.6

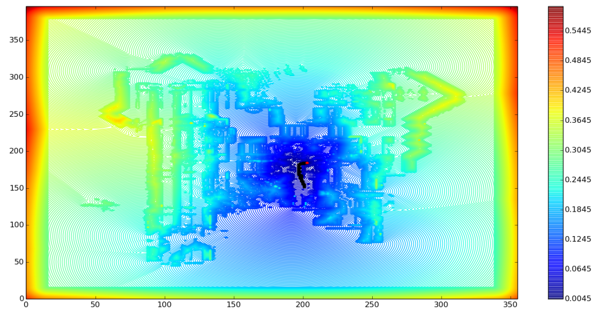

Fig. 12. Path planning over the representation of the isochrone lines of the processed surface represented in Fig.8

V. Conclusions and Future Work

This paper has presented a method for path planning for field robots on a rough terrain. The method is based on a gradient descend over a potential field. This potential field is obtained using FMM over a cost map. This cost map is computed by considering the robot attitude when placed at each point of the grid. Preliminary results are promising, as they show feasible paths over a reference map. As future work we intend to address the computational efficiency, since not all points of the grid need to be computed. We also intend to perform experiments with a real robot, the RAPOSA-NG.

References

[1] S. Garrido, L. Moreno, D. Blanco, and F. Martin, "Smooth path planning for non-holonomic robots using fast marching," in *Mechatronics, 2009. ICM 2009. IEEE International Conference on*, April 2009, pp. 1–6.

[2] J. A. Sethian, "Fast marching methods," *SIAM Review*, vol. 41, no. 2, pp. 199–235, 1999.

[3] C. Marques, J. Cristovão, P. Alvito, P. Lima, J. Frazão, M. I. Ribeiro, and R. Ventura, "A search and rescue robot with tele-operated tether docking system," *Industrial Robot*, vol. 34, no. 4, pp. 332–338, 2007.

[4] B. Siciliano and O. Khatib, *Springer Handbook of Robotics*, ser. Gale virtual reference library. Springer, 2008.

[5] S. LaValle, *Planning Algorithms*. Cambridge University Press, 2006.

[6] M. Tarokh, Z. Shiller, and S. Hayati, "A comparison of two traversability based path planners for planetary rovers," *Proc. i-SAIRAS99*, pp. 151–157, 1999.

[7] S. Garrido, M. Malfaz, and D. Blanco, "Application of the fast marching method for outdoor motion planning in robotics," *Robotics and Autonomous Systems*, vol. 61, no. 2, pp. 106–114, 2013.

[8] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996.

[9] D. Kraft, *A software package for sequential quadratic programming*. DFVLR Obersfaffeuhofen, Germany, 1988.

[10] R. Siegwart, I. Nourbakhsh, and D. Scaramuzza, *Introduction to Autonomous Mobile Robots*, ser. Intelligent robotics and autonomous agents. MIT Press, 2011.

[11] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, ser. Intelligent robotics and autonomous agents. MIT Press, 2005.

[12] C. Castejón, B. L. Boada, D. Blanco, and L. Moreno, "Traversable region modeling for outdoor navigation," *Journal of Intelligent and Robotic Systems*, vol. 43, no. 2-4, pp. 175–216, 2005.

[13] G. A. Pereira, L. C. Pimenta, A. R. Fonseca, L. D. Q. Corrêa, R. C. Mesquita, L. Chaimowicz, D. S. De Almeida, and M. F. Campos, "Robot navigation in multi-terrain outdoor environments," *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 685–700, 2009.

[14] L. Ojeda, J. Borenstein, G. Witus, and R. Karlsen, "Terrain characterization and classification with a mobile robot," *Journal of Field Robotics*, vol. 23, no. 2, pp. 103–122, 2006.

[15] J. Liang, "A path planning algorithm of mobile robot in known 3d environment," *Procedia Engineering*, vol. 15, no. 0, pp. 157 – 162, 2011, {CEIS} 2011.

[16] M. Tarokh, "Hybrid intelligent path planning for articulated rovers in rough terrain," *Fuzzy Sets and Systems*, vol. 159, no. 21, pp. 2927 – 2937, 2008, selected Papers from {NAFIPS} 2006 2006 Annual Conference of the North American Fuzzy Information Processing Society.