ABCM Symposium Series in Mechatronics - Vol. 2 - pp.218-225

PATH PLANNING FOR MOBILE ROBOTS OPERATING IN OUTDOOR ENVIRONMENTS USING MAP OVERLAY AND TRIANGULAR DECOMPOSITION

Alexandre R. Fonseca
Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais
arfonseca@ufmg.br

Luciano C. A. Pimenta
Programa de Pós-Graduação em Engenharia Elétrica - Universidade Federal de Minas Gerais
lucpim@cpdee.ufmg.br

Renato C. Mesquita
Departamento de Engenharia Elétrica - Universidade Federal de Minas Gerais
renato@cpdee.ufmg.br

Rodney R. Saldanha
Departamento de Engenharia Elétrica - Universidade Federal de Minas Gerais
rodney@cpdee.ufmg.br

Guilherme A. S. Pereira
Departamento de Engenharia Elétrica - Universidade Federal de Minas Gerais
gpereira@cpdee.ufmg.br

Abstract. *This paper addresses the mobile robot motion planning problem in outdoor environments, which are large, sparsely occupied workspaces with uneven terrains. We present results in path planning using an efficient discretization method based on Constrained Delaunay Triangulation (CDT) and classical graph searching algorithms. CDT has been proven to be a good method for representing complex shaped objects and regions, very common in outdoor environments. In order to combine several constraints associated with different forms of interaction between robot and workspace, map overlay techniques are used.*
Keywords: *mobile robots, navigation, triangulation, graphs*

1. Introduction

Robot motion planning is a broad field of research that includes manipulator motion planning and mobile robot navigation. In this second field, several works were developed for guiding mobile robots both in indoor and outdoor environments [Latombe, 1991]. In this sense, the mobile robot motion planning problem is basically stated as the problem of leading the robot from an initial place to another, by avoiding obstacles in the environment. When we are considering indoor motion planning, examples of obstacles are walls, stairs, furniture, people and other objects that can compromise the robot safety and the execution of the task in case of collisions. Therefore, obstacles can be thought of forbidden regions for the robot path. Besides forbidden regions, outdoor environments, which are large, sparsely occupied workspaces with uneven terrains, also present motion constraints related to the characteristics of the robot and its interaction with the environment. These constraints are not necessarily obstacles to be avoided, but also must be considered while planning the robot motion. For example, consider a wheeled robot transversing a golf grass field with small sand regions. Suppose the robot's mobility is compromised (but is not impossible) in the sand. Then, it would be interesting (but not mandatory) to the robot to avoid the sand regions and navigate in the grass. However, depending on the size and position of those regions, even losing efficiency in the sand, it could be more efficient to the robot to cross a specific sand region instead of avoiding it. In this paper we present a path planning approach that considers dynamic and static constraints to the robot motion while it is navigating in outdoor environments. We allow for several simultaneous constraints such as mobility, communication and power consumption, and also include the traditional obstacles represented by forbidden regions. Since we are dealing with large workspaces with complex shaped regions we use an efficient discretization algorithm, over which paths are constructed based on classical graph search algorithms.

In [Kobilarov and Sukhatme, 2005, Guivant et al., 2004, Guo et al., 2003, Yahja et al., 2000, Mitchell, 1991] solutions to the outdoor path planning problem are proposed. In [Guo et al., 2003], the outdoor environment map is decomposed into a regular grid and the A^* shortest path algorithm is used to minimize a cost function composed by terrain roughness, slope and distance between cells. The apparent problem of that approach is the use of regular grid decomposition, which is inefficient to represent large workspaces. [Kobilarov and Sukhatme, 2005] also decompose the environment in a regular

grid. They define a cost metric for each cell based on the time the robot spend to transverse it. A kinodynamic simulation is needed to define this metric and a Probabilistic Roadmap [Latombe, 1991] approach is used to plan the robot's trajectory. [Yahja et al., 2000] proposes a quadtree based discretization of the environment map. Quadtree is a non uniform representation of the map that has the advantage of representing regions of interest, such as obstacles, with high resolution cells, and other regions with a low resolution discretization, thus yielding in a smaller number of cells. Their framed-quadtree approach proved to be more efficient than the regular grid and standard quadtree decomposition. After cell decomposition the authors use the D^* algorithm [Stentz, 1995], which is a dynamic version of A^*, to find the shortest path in the map. In their case, a distance based cost function is used. To improve map discretization [Guivant et al., 2004] used a hybrid representation of the environment based on feature maps and other metric functions. The map is divided in connected triangular regions, defined by the position of three environment landmarks, and a local function, which can represent for example the terrain occupancy, is defined inside the triangles. Differently from all these works, in [Mitchell, 1991], paths are searched in a continuous map. The author uses a continuous version of the Dijkstra algorithm and exploits the fact that shortest paths obey Snell's Law of Refraction at region boundaries. Constant costs are defined for different kinds of terrains and used in the path minimization algorithm.

Among all these, our work is closely related to [Guo et al., 2003], [Yahja et al., 2000] and [Mitchell, 1991], but with important differences. First, differently from [Guo et al., 2003], which computes cost values for each cell of a decomposed map, we propose a formal way to compose cost functions based on overlay of continuous maps. Second, although we work initially with continuous maps like in [Mitchell, 1991], our path searching approach is based on discrete maps. However, instead of regular or quadtree decomposition, we decompose the environment using the Constrained Delaunay Triangulation (CDT) [Shewchuk, 1997]. Similar to quadtree, CDT is a non-uniform decomposition that yields in high resolution cells in the complex regions of the environment. An important advantage of CDT over the quadtree representation is the smaller number of cells when representing non-poligonal, high complex structures, common in outdoors environments [Shewchuk, 1997]. Observe that, differently from [Guivant et al., 2004] we do not define metric functions inside the triangular cells (that in our case tend to be much smaller than the ones defined in [Guivant et al., 2004]), but due to the triangulation process, to each triangle is assigned a constant weight, based on the combination of several motion constraints. We use Dijkstra's graph search algorithm (or even the suboptimal versions of them, A^* or D^*) for finding the robot path.

2. Problem Formulation

Consider a robot R operating in an outdoor environment. The environment is represented by a set of k thematic maps $\mathcal{M} = \{M_1, M_2, \ldots, M_k\}$. Each map is defined as $M_i = \{(x, y, z_i(x, y))|x_{min} \leq x \leq x_{max}; y_{min} \leq y \leq y_{max}\}$, where $0 \leq z_i(x, y) \leq +\infty$ is a function that represents a specific characteristic of the environment in the robot position (x, y), such as, obstacles, slope, communication intensity, density of people, and others. Actually, $z_i(x, y)$ gives a cost of transposing per unit distance. At first, no map discretization is assumed and $(x, y) \in \mathbb{R}^2$. Higher values of $z(x, y)$ represent challenging regions for the robot or the completion of its task. In the extreme, $z(x, y) = +\infty$ represents a impossibility for the robot motion. We propose the composition of all environment characteristics to be represented by the map $\mathcal{W}$ given by:

$$\mathcal{W} = \{(x, y, g(x, y))|g(x, y) = \sum_{i=1}^{k} w_i z_i(x, y)\}, \tag{1}$$

where w_i are weighting factors, used to establish priorities between the maps. We now define the forbidden regions of the map, $\mathcal{O}$, as the regions where $z(x, y) = +\infty$ and the free regions of the map as $\mathcal{F} = \mathcal{W}/\ \mathcal{O}^\epsilon$, where $\mathcal{O}^\epsilon$ is a region within ϵ distance of $\mathcal{O}$. In this case, the constant ϵ plays the role of a safety margin for the robot's motion. It can also encapsulates the robot size and errors in the robot's pose estimation. Our goal is to find the path given by a curve $\mathcal{L} \in \mathcal{F}$, which connects the start robot position $q_0 = (x_0, y_0)$ to the goal position $q_g = (x_g, y_g)$, and minimizes the cost functional given by:

$$I = \int_{\mathcal{L}} g(x, y)ds\,, \tag{2}$$

where ds is the differential of the arc length.

3. Path Planning Approach

Our solution for the problem formulated in the previous section is divided in three main parts, namely: (i) map overlaying, (ii) map discretization and (iii) path searching. Each of these parts will be explained in details next.

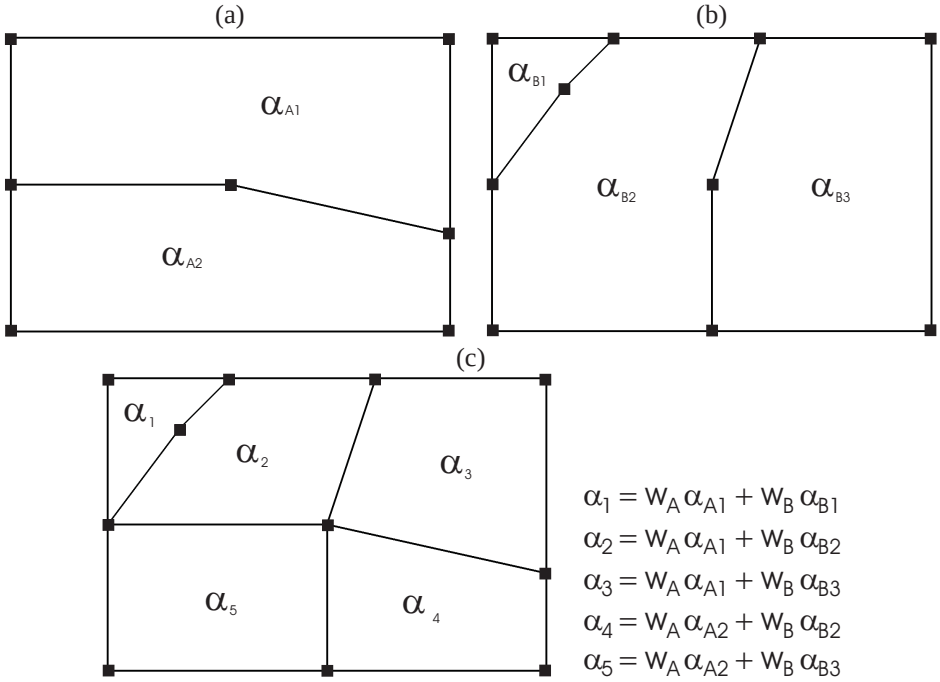

Figure 1. Map Overlaying. (a) Map A. (b) Map B. (c) Combined map.

3.1 Map Overlaying

A proper data structure to represent thematic maps is a *planar map* [Gangnet et al., 1989]. A planar map subdivides the plane into vertices, edges, and faces. In this paper, it is considered that the outdoor environment is represented by a set of planar maps.

The way we represent the cost to transpose a given region in a specific thematic map is similar to [Mitchell, 1991]. The f^{th} face of the map, $\Upsilon_f \subset \mathbb{R}^2$, represents a specific property in a thematic map and have assigned to it a transposing cost $\alpha_f \in [0, +\infty]$ per unit distance. Therefore, the cost function $z_i(x, y)$ presented in the previous section can be defined as:

$$z_i(x, y) = \alpha_f \quad \forall (x, y) \in \Upsilon_f, f = 1, 2, \ldots, n, \tag{3}$$

where n is the number of faces in the map i.

To each edge e of the map it is also associated a cost $\alpha_e \in [0, +\infty]$. In general, in case of two adjacent faces f_1 and f_2 the cost assigned to a separation edge is the lowest one among the two faces:

$$\alpha_e = \min(\alpha_{f_1}, \alpha_{f_2}). \tag{4}$$

The total cost $\beta(P_1, P_2)$ of a line segment that links two points P_1 and P_2 placed on a face f is given by:

$$\beta(P_1, P_2) = \alpha_f |P_1 P_2|, \tag{5}$$

where $|P_1 P_2|$ is the Euclidean distance. Equation (5) can also be used to compute the total cost in the case of P_1 and P_2 located on an edge e. The only modification is the use of α_e instead of α_f.

Map overlaying is a technique to combine information from different thematic maps. In order to perform the overlay of two maps we initially compute the intersection among all segments of the two maps [Chazelle and Edelsbrunner, 1992, Balaban, 1995]. The next step is to classify the regions of intersection and combine the original costs per unit distance [U. Finke, 1995]. Figure 1 shows two maps A (Figure 1(a)) and B (Figure 1(b)) and also the combined one (Figure 1(c)). We propose to use a weighted sum of the original costs per unit distance to compute the costs of the overlayed map $\mathcal{W}$ according to Equation (1).

3.2 Map Discretization

In our methodology we discretize the resultant overlayed map. We propose the use of the Constrained Delaunay Triangulation where the initial robot's position q_0 and the target q_g are forced to be vertices. Different triangles sizes can be obtained in this process, which makes this method powerful and flexible. In regions where the geometry is more

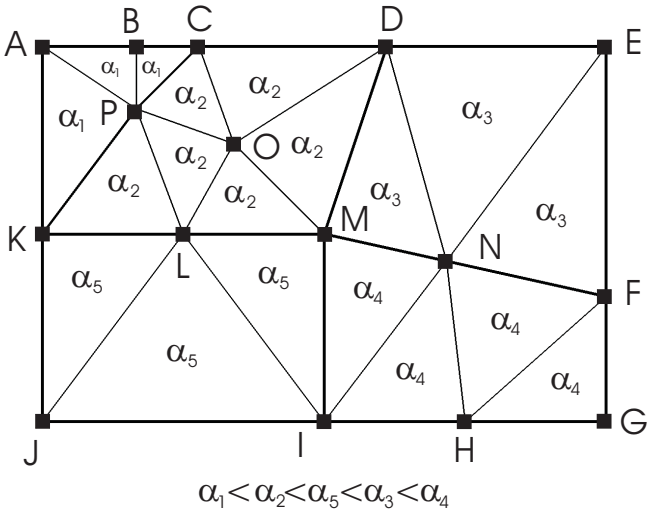

Figure 2. Constrained Delaunay Triangulation for the overlayed map presented in Figure 1.

complicated, higher triangles densities are used. On the other hand, in regions with simpler geometry less triangles may be used. If it is desirable to have a good approximation of the continuous optimal path, then this type of discretization is useful for reducing the computational complexity of the computation for high complex geometry maps. Notice that other triangulation processes could produce similar results. One advantage of CDT over other triangulation approach is that, because it is a very well known triangulation, several free computational tools are available to compute it.

The algorithm we have implemented in this paper was the *Delaunay Refinement* [Shewchuk, 1997]. This technique operates by maintaining a *Delaunay Triangulation*, which are refined by the insertion of additional vertices. The placement of these vertices is chosen to enforce boundary conformity, to satisfy the constraints imposed by the points q_0 and q_g, and to improve the quality of the triangulation. Figure 2 shows a possible CDT computed for the overlayed map presented in Figure 1. The Delaunay triangulation of a set of points, $\mathcal{P}$, is a set of triangles connecting the points satisfying an "empty circle" property: the circumcircle of each triangle does not contain any of the points of $\mathcal{P}$ in its interior. This triangulation has many interesting properties as presented in [de Berg et al., 2000]. Further details can be found in [Shewchuk, 1997].

3.3 Path Searching

Based on the generated triangulation of the overlayed map we must search for a minimum cost path from the initial vertex q_0 to the goal vertex q_g. In order to compute such a path, we first create a graph $G(v, e)$ where the triangulation vertices are the graph nodes, v, and the triangulation edges are the graph edges, e. The costs per unit distance associated with the graph edges are the same ones associated with the edges of the triangles.

As we can see in Figure 2 to each triangle we assign a cost per unit distance. Indeed, the triangles are faces of the discretized overlayed map and they inherit the cost value of the non-discretized faces they belong. Likewise, the edges of the overlayed map, which are also edges of the triangulation, keep the same cost per unit distance they had before the triangulation. The new edges created by the process of triangulation receive the same cost assigned to the triangles with minimum cost that share the respective edges. This is basically the same procedure described in Equation (4), where faces f_1 and f_2 are triangles that share the given edge e of the triangulation.

Therefore, given two graph nodes v_1 and v_2 sharing an edge e, the total cost β_e of following e from v_1 to v_2 is computed by Equation (5). Figure 3 shows the graph we generated from the example of the triangulated map in Figure 2 and the corresponding costs equations.

Our continuous problem is now transformed to a discrete one stated as follows: Find the minimum cost path from q_0 to q_g in the graph $G(v, e)$, given the edge costs β_e computed as described before. This path searching can be performed by very well known algorithms such as A^*, D^* or Dijkstra [Dijkstra, 1959]. By using the last one we can guarantee we find the optimal path for the given graph. Notice that we are not saying we find the optimal path for the continuous problem. The higher the discretization the better is our approximation.

Using a high resolution map decomposition can be computationally expensive. So, we use a postprocessing procedure to improve our initial solution without using a large number of nodes. Given three consecutive points P_1, P_2 and P_3 and the segments $\overline{P_1P_2}$ and $\overline{P_2P_3}$ in the path, we substitute these two segments by the segment $\overline{P_1P_3}$ if the last one is less expensive than the sum of the other two. An example of such processing is presented in Figure 4.

Simulated results that exemplify all three steps of our approach are presented in the next section.

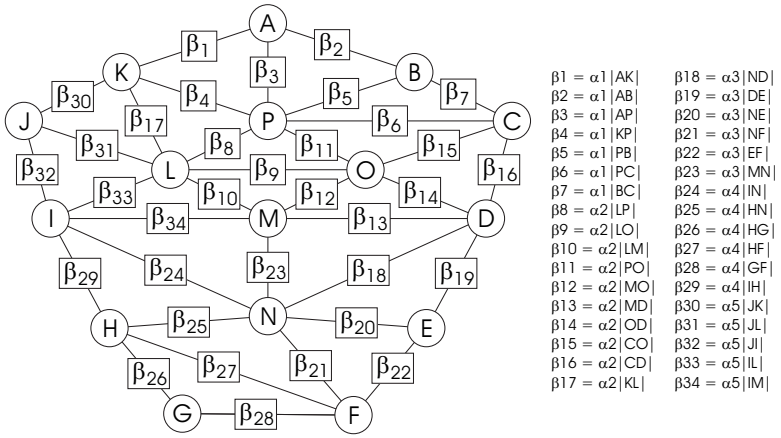

Figure 3. Graph created from the triangulated map presented in Figure 2.

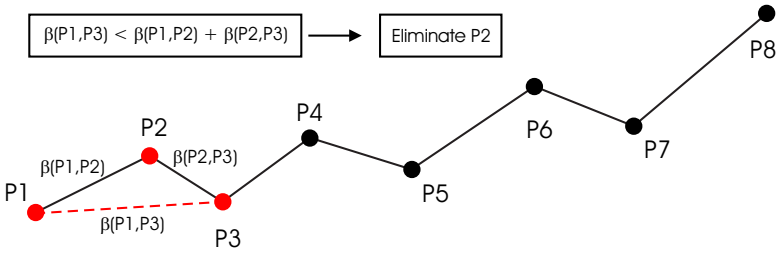

Figure 4. Postprocessing procedure.

4. Simulations

In order to illustrate our path planning methodology we have used a hypothetic environment, which is represented by four maps, as shown in Figure 5. The proportion between the two dimensions of these maps is 2×1. Our software was developed in C++ with the additional use of some specific data structures from CGAL (Computational Geometry Algorithms Library) [CGAL, 2005]. More specifically, we have used "Planar Nef Polihedra" and "2D Planar Maps" data structures which allows for robust map overlay computation. Furthermore, we have used "Triangle" [Shewchuk, 1996] program to perform the Constrained Delaunay Triangulation.

Figure 5(a) shows the terrain map. For each kind of terrain we have defined values related to the robot motion capability. In this specific case, water received the higher value, and was followed by sand, grass and pavement. The environment has two access ramps that received a value between sand and grass. The map in Figure 5(b) represents the obstacles in the environment. Notice that the only obstacle is a retangular building. Figure 5(c) is a map associated to the probability the robot has to meet people in its way. Observe that high probabilities occur in the neighborhoods of the building in Figure 5(b). Figure 5(d) represents the communication intensity map. Three fixed antennas were responsible for the robot communication with its operator. Because it is desirable that the robot communicates its sensor data while moving, low cost values were assigned to the regions with high signal intensity.

Initially, we will search for robot paths in the terrain map only (Figure 5(a)). The first step of the proposed methodology is the environment triangulation. The triangularized terrain map can be seen in Figure 6. Notice that the resolution of the triangulation is higher at the more complex regions of the map. In Figure 6 the cost values of each region are represented by the gray tones. The darker is the region, the higher is its cost. After triangulation, we have applied the Dijkstra algorithm over the resultant graph and found the path shown in Figure 6(a). After postprocessing, the final trajectory is shown in Figure 6(b). In both cases, observe that the path may lead the robot safely to the goal since it avoids the forbidden regions, such as water. Also, observe that, even that the distance from start to goal position passing through the ramp on the top seems to be smaller, the path is mostly confined in the pavement, where the locomotion cost is smaller than grass and sand.

We now combine the terrain map with obstacle and probability of people maps by using the overlay technique proposed in Section 3.1. The resultant map is shown in Figure 7(a). By comparing the gray tones in this map with the ones in Figure 6, it can be perceived how the overlay of maps have changed the cost associated with some regions of the terrain map. Figure 7(b) shows the resultant path after the application of our path planning approach. Notice how the robot avoids the regions where there is probability of people. By avoiding these regions, the robot do not need to use real time obstacle avoidance algorithms, which would change the robot path.

In the last simulation presented in this paper, we have overlayed all maps in Figure 5. The resultant map is shown

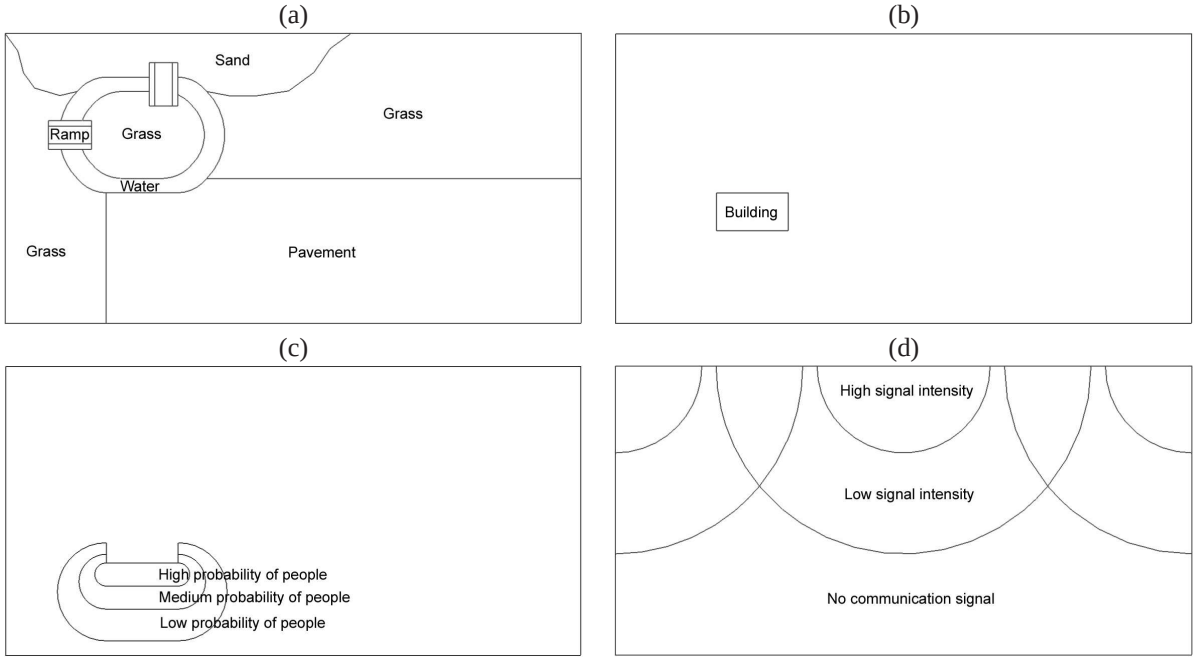

Figure 5. Four maps from the same workspace. (a) Terrain map; (b) Building map; (c) Probability of people map; (d) Communication signal intensity map.

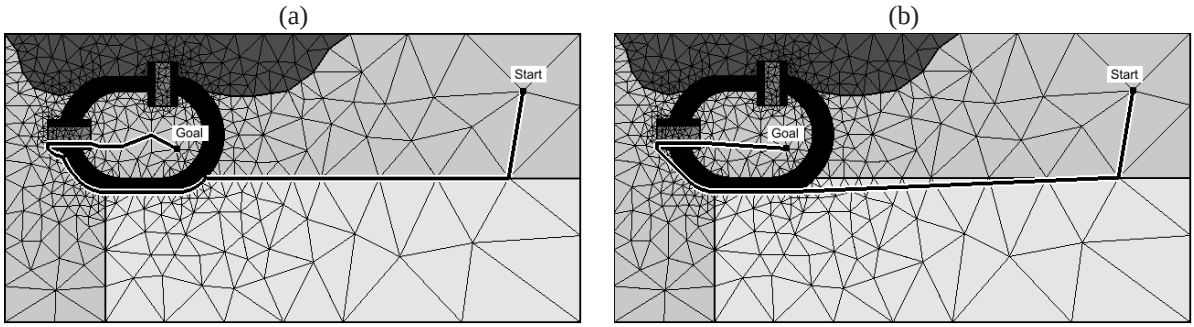

Figure 6. Paths planned in the terrain map. (a) Path resultant from Dijkstra; (b) Postprocessed path.

in Figure 8(a). Notice that the introduction of constraints related to communication signal strength demanded a fine discretization in the regions we had large triangles, such is the case in the right hand regions of the workspace. This fact was expected since we have a more complex geometry. The path found for this map is in Figure 8(b). In order to maintain communication the robot performs a different path and cross a small piece of sand, what in other situations was avoided.

Performance measurements of our implementation are shown in Table 1. The simulations were executed in a Pentium IV, 1.6 GHz, 512 MB of RAM running Windows XP. One should notice that the time spent to build the needed data structures is also included in the respective presented numbers. From Table 1 we conclude that the more complex are the maps the more expensive is the computation. Furthermore, the discretization and the postprocessing steps are irrelevant comparing to the other two. Although it does not seem to be possible to do all this computation in real time, we believe our approach is feasible since we do not intend to have a deliberative planning being computed on the fly.

	Simulation - Figure 6	Simulation - Figure 7	Simulation - Figure 8
Number of Vertices in the Overlayed Map	112	226	318
Number of Vertices after Triangulation	595	1093	2848
Overlay time (s)	-	27.85	72.88
Map Discretization time (s)	0.38	0.33	0.43
Graph Search time (s)	7.05	22.13	62.80
Postprocessing time (s)	0.06	0.06	0.04

Table 1. Performance measurements for the simulated examples.

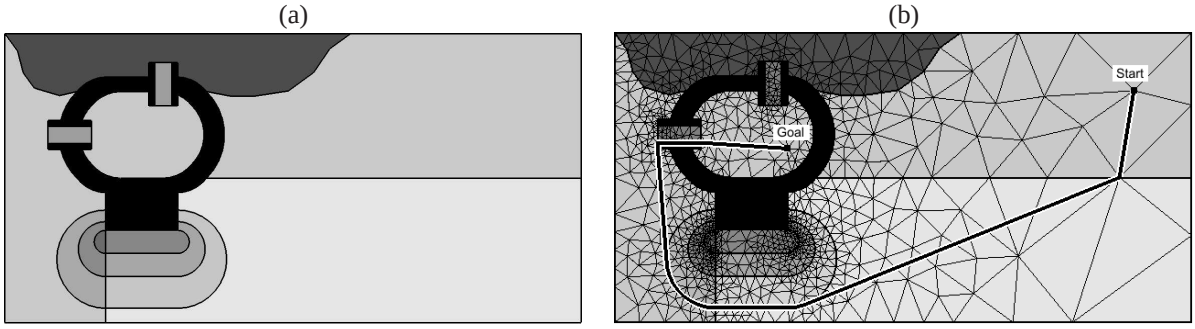

Figure 7. (a) Overlay of terrain, obstacles and probability of people maps; (b) Postprocessed path.

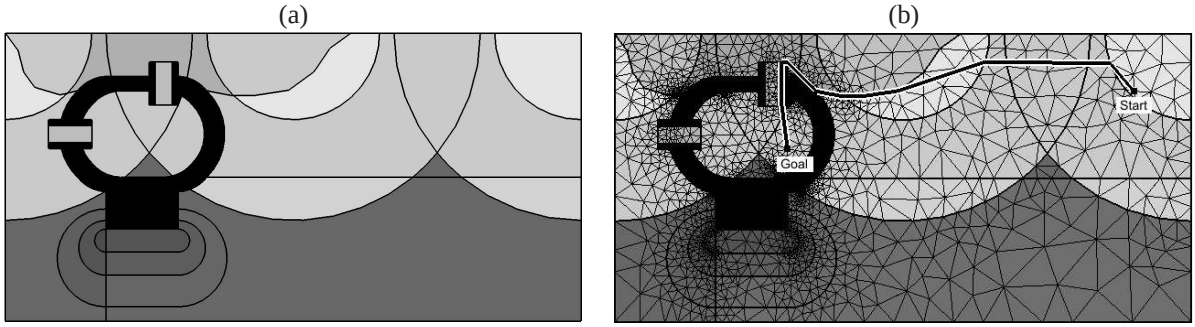

Figure 8. (a) Overlay of the maps of terrain, obstacles, probability of people and communication; (b) Postprocessed path.

5. Conclusions and Future Work

This work presented a path planning approach based on map overlay, triangular decomposition, and simple graph search algorithms. The approach was used to plan mobile robot paths in outdoor environments. In this context, map overlay allowed us to work with several simultaneous motion constraints, such as terrain roughness, obstacles, and density of mobile obstacles. Triangulation proved to be an efficient non-uniform map decomposition tool, that was used to precisely represent several complex maps. In this paper we have used the Dijkstra algorithm to find the optimal trajectory for a given triangulation. For larger environments, non-optimal algorithms, such as the A^*, could also be used.

Next steps of this research includes the implementation of the proposed methodology for actual robots, such as our team of Pioneer P3-AT mobile platforms. Our plan is to have in a few years an autonomous campus tour guide that will be able to navigate at UFMG. This is a challenging goal, since, besides navigation, the robot will need to be well localized in the workspace. Pose estimation algorithms are currently been developed by our research group. In relation to navigation, although in this paper we have arbitrarily defined values for the cost of each motion constraint, theoretical and empirical studies must be executed to generate more consistent results. An example of terrain modelling can be found in [Iagnemma et al., 2004]. Also, we intend to compare the approach of this paper with a different methodology based on electromagnetic field computation. We are currently working on adapting our indoor approach based on this theory [Pimenta et al., 2005] for outdoor environments.

6. Acknowledgements

The authors would like to thank the financial support of CNPq and FAPEMIG.

7. References

Balaban, I. J. (1995). An optimal algorithm for finding segments intersections. In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 211–219, Vancouver, British Columbia, Canada.

CGAL (2005). The CGAL home page. `http://www.cgal.org`. visited in August, 2005.

Chazelle, B. and Edelsbrunner, H. (1992). An optimal algorithm for intersecting line segments in the plane. *Journal of the ACM (JACM)*, 39(1):1–54.

de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O. (2000). *Computational Geometry Algorithms and Applications*. Springer-Verlag, second edition.

Dijkstra, E. W. (1959). A note on two problems in connection with graphs. *Numerische Mathematik 1*, pages 269–271.

Gangnet, M., Hervé, J.-C., Pudet, T., and van Thong, J.-M. (1989). Incremental computation of planar maps. In *Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, pages 345–354.

Guivant, J., Nebot, E., Nieto, J., and Masson, F. (2004). Navigation and mapping in large unstructured environments. *The International Journal of Robotics Research*, 23(4-5):449–472.

Guo, Y., Parker, L. E., Jung, D., and Dong, Z. (2003). Performance-based rough terrain navigation for nonholonomic mobile robots. In *Proceedings of the IEEE Industrial Electronics Society*, pages 2811–2816.

Iagnemma, K., Kang, S., Shibly, H., and Dubowsky, S. (2004). Online terrain parameter estimation for wheeled mobile robots with application to planetary rovers. *IEEE Transactions on Robotics*, 20(5):921–927.

Kobilarov, M. B. and Sukhatme, G. S. (2005). Near time-optimal constrained trajectory planning on outdoor terrain. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1833–1840.

Latombe, J.-C. (1991). *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA.

Mitchell, J. S. B. (1991). The weighted region problem: finding shortest paths through a weighted planar subdivision. *Journal of the Association for Computing Machinery*, 38(1):18–73.

Pimenta, L. C. A., Fonseca, A. R., Pereira, G. A. S., Mesquita, R. C., Silva, E. J., Caminhas, W. M., and Campos, M. F. M. (2005). On computing complex navigation functions. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'05)*, pages 3463–3468, Barcelona, Spain.

Shewchuk, J. R. (1996). Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In Lin, M. C. and Manocha, D., editors, *Applied Computational Geometry: Towards Geometric Engineering*, volume 1148 of *Lecture Notes in Computer Science*, pages 203–222. Springer-Verlag. From the First ACM Workshop on Applied Computational Geometry.

Shewchuk, J. R. (1997). *Delaunay Refinement Mesh Generation*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania. Available as Technical Report CMU-CS-97-137.

Stentz, A. (1995). The focussed D^* algorithm for real-time replanning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 605 – 611.

U. Finke, K. H. H. (1995). Overlaying simply connected planar subdivisions in linear time. In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 119–126, Vancouver, British Columbia, Canada.

Yahja, A., Singh, S., and Stentz, A. (2000). An efficient on-line path planner for outdoor mobile robots. *Robotics and Autonomous Systems*, 32:129–143.

8. Responsibility notice

The authors are the only responsible for the printed material included in this paper.