Robot Navigation in Multi-terrain Outdoor Environments

Guilherme A. S. Pereira, Luciano C. A. Pimenta, Alexandre R. Fonseca, Leonardo de Q. Corrêa, Renato C. Mesquita, Luiz Chaimowicz, Daniel S. C. de Almeida and Mario F. M. Campos

The International Journal of Robotics Research 2009; 28; 685
DOI: 10.1177/0278364908097578

Robot Navigation in Multi-terrain Outdoor Environments

Guilherme A. S. Pereira
Luciano C. A. Pimenta
Alexandre R. Fonseca
Leonardo de Q. Corrêa
Renato C. Mesquita

Departamento de Engenharia Elétrica,
Universidade Federal de Minas Gerais,
Belo Horizonte, MG 31270-010, Brazil
gpereira@ufmg.br

Luiz Chaimowicz
Daniel S. C. de Almeida
Mario F. M. Campos

Departamento de Ciência da Computação,
Universidade Federal de Minas Gerais,
Belo Horizonte, MG 31270-010, Brazil

Abstract

This paper presents a methodology for motion planning in outdoor environments that takes into account specific characteristics of the terrain. Instead of decomposing the robot configuration space into "free" and "occupied", we consider the existence of several regions with different navigation costs. In this paper, costs are determined experimentally by navigating the robot through the regions and measuring the influence of the terrain on its motion. We measure the robot's vertical acceleration, which reflects the terrain roughness. The paper presents a hybrid (discrete–continuous) approach to guide and control the robot. After decomposing the map into triangular cells, a path planning algorithm is used to determine a discrete sequence of cells that minimizes the navigation cost. Robot control is accomplished by a fully continuous vector field that drives the robot through the sequence of triangular cells. This vector field allows smooth robot trajectories from any position inside the sequence to the goal, even for a small number of large cells. Moreover, the vector field is terrain dependent in the sense it changes the robot velocity according to the characteristics of the terrain. Experimental results with a differential driven, all-terrain mobile robot illustrate the proposed approach.

KEY WORDS—robot motion planning, field robotics, continuous vector fields

1. Introduction

The motion planning problem for mobile robots can be loosely stated as the problem of finding and executing an obstacle-free path from an initial position to a pre-specified goal. When dealing with indoor environments, examples of obstacles are walls, stairs, furniture, and other objects that may compromise robot safety and the overall task execution. Therefore, obstacles are considered as forbidden regions in the robot's configuration space. Outdoor environments, which are sparsely occupied and composed of different regions, pose new challenges in robot navigation. In general, robots will have motion constraints related to their interaction with the environment. Robots will face different surfaces (grass, concrete, gravel), uneven terrains, and other constraints, which are not necessarily obstacles to be avoided, but must be considered while planning the robot motion.

In this paper, we propose a methodology for motion planning in outdoor environments that takes into account these characteristics. Instead of decomposing the configuration space into "free" and "occupied", we consider the existence of several regions with different navigation costs. Costs are determined experimentally by navigating the robot through these regions and measuring its performance. Robot control is accomplished by a fully continuous vector field that drives the robot through different discrete regions. An important feature of this vector field is the possibility of specifying a velocity profile for the robot. We use this characteristic to bound some undesirable effects of the terrain in the robot motion.

The International Journal of Robotics Research
Vol. 28, No. 6, June 2009, pp. 685–700
DOI: 10.1177/0278364908097578

Figure 6 appears in color online: http://ijr.sagepub.com

The idea of decomposing the environment in regions and applying different weights for each in the path planning was proposed by Mitchell and Papadimitriou (1991). In that paper, the total length of a path is defined to be the weighted sum of (Euclidean) lengths of the subpaths within each region and the "continuous Dijkstra" algorithm is used for path planning. The paper develops interesting proofs, but it is mainly theoretical and no experimental results are presented. Instead of considering a continuous environment, some works propose the use of a discretization algorithm before the path planning. Guo et al. (2003), for example, used a regular grid and the A^* algorithm in simulations. Another approach (Yahja et al., 2000) uses a framed quadtree and the D^* algorithm for path planning.

In addition to decomposing the environment, the path planning needs a cost metric for each region. In this paper, we are particularly interested in using the terrain roughness as a metric. A good way to estimate terrain roughness is through robot vibration, which can be obtained by the acceleration in the z-axis measured by an accelerometer mounted vertically on the robot. This idea was proposed initially by Brooks and Iagnemma (2005), who analyzed data acquired through these sensors using principal components and compared it with labeled data obtained off-line, allowing a planetary rover to identify the type of terrain it is navigating. A similar but simpler approach is presented by Sadhukhan et al. (2004), in which a fast Fourier transform (FFT) is applied on the sensor data and is used as an input to a neural network that performs classification among three different types of terrain. Despite being a simple approach, vertical acceleration has been used in other successful initiatives such as *Stanley*, the winner of the DARPA Grand Challenge in 2005, which uses vertical acceleration as one of the metrics to dynamically adjust velocity according to the terrain (Thrun et al., 2006). For a more sophisticated analysis, other types of information about the terrain may be needed. In this case a combination of several sensors (gyros, accelerometers, encoders, voltage, etc.) can be used (Ojeda et al., 2006).

Similarly to other outdoor navigation works, our approach decomposes the environment into discrete regions. However, differently from Guo et al. (2003) and Yahja et al. (2000), we decompose the environment using the constrained Delaunay triangulation (CDT) (Shewchuk, 1996). An important advantage of CDT over regular grid and quadtree representations is that it usually generates a much smaller number of cells when representing complex structures, typical of outdoor environments. For computing the navigation costs, we also use the terrain roughness estimated measuring the vertical acceleration in each terrain. However, leveraging previous works, we use it both in path planning and motion control. Finally, differently from other works on outdoor navigation, among our contributions is the design of a controller based on a continuous, terrain cost-dependent vector field that leads the robot through a sequence of different triangular regions. The presence of a continuous vector field inside the sequence of discrete cells allows smooth robot trajectories from any position inside the sequence to the goal, even for a small number of large cells. This property is very important in reducing the motion planning's computational cost in situations where the robot is traversing large outdoor environments. In addition, the methodology inherits several important properties of many vector field approaches, such as robustness to small localization errors and the possibility of local replanning only to avoid unmodeled obstacles.

The construction of this vector field is based on the indoor approach presented by Belta et al. (2005) and on its modified version that allows for fully continuous vector fields initially presented by Pimenta et al. (2007). For indoor navigation, other vector field approaches that are able to compute continuous fields on a discrete environment have been proposed (Conner et al., 2003; Lindemann and LaValle, 2005). All of them could, at first, be adapted to allow outdoor navigation as proposed in this paper. However, the main motivation for seeking a new methodology, that at the end became simpler and computationally more efficient than the previous ones, was the necessity to directly and automatically incorporate some properties of the terrain into the vector field computation, thus generating bounded velocity profiles for the robot.

This paper is organized as follows. Section 2 presents the problem statement. Our three-layered motion planning and control architecture is presented in Section 3, which also includes mathematical and computational analyses of the methodology. Experimental results with a differential driven all-terrain robot in a multi-terrain environment are shown in Section 4. Finally, conclusions and future work are discussed in Section 5.

2. Problem Statement

Consider a robot R operating in an outdoor environment. The environment is represented by a thematic map $\mathcal{W}$. A thematic map is a map that represents not only geographical features (such as borders and roads), but also the spatial variation of either a single or a small number of geographic distributions (such as population and types of terrain) (de Berg et al., 2000). In this paper the thematic map is defined as $\mathcal{W} = \{(x, y, g(x, y)) | x_{\min} \leq x \leq x_{\max}; y_{\min} \leq y \leq y_{\max}\}$, where $0 \leq g(x, y) \leq +\infty$ is a function that expresses the cost of traversing a region per unit distance at robot position (x, y). It is influenced by some characteristics of the environment, such as obstacles, slopes, terrain roughness, and others. If necessary, $g(x, y)$ can also represent a composition of several cost functions, as proposed by Fonseca et al. (2005). At first, no map discretization is assumed and $(x, y) \in \mathbb{R}^2$. Larger values of $g(x, y)$ represent challenging regions for the robot or for the completion of its task. In the limit, $g(x, y) = +\infty$ is assigned to regions where the robot cannot go. Thus, we define

the forbidden regions of the map, $\mathcal{O}$, as the regions where $g(x, y) = +\infty$ and the free regions of the map as $\mathcal{F} = \mathcal{W} \setminus \mathcal{O}$.

The main objective of this work is to drive the robot from its initial position $\mathbf{q}_0 = (x_0, y_0)$ through a path Υ to the goal position $\mathbf{q}_g = (x_g, y_g)$ both in $\mathcal{F}$, such that the cost functional given by

$$I = \int_{\Upsilon} g(x, y)\, ds, \tag{1}$$

where ds is the differential of the arc length, is minimized. In addition, we pursue real-world, computationally efficient implementations that are robust to localization and actuation errors.

The methodology we developed to address this problem is described in the next section.

3. Methodology

Our solution to solve the problem stated in the previous section is based on three hierarchical layers: (i) a high-level discrete planner responsible for finding a low-cost sequence of cells to be traversed by the robot; (ii) an intermediate level that computes a continuous vector field inside the sequence of cells; and (iii) a low-level controller that enforces the robot to track the vector field and avoid unmodeled and dynamic obstacles. The first two layers are detailed in Sections 3.1 and 3.2 while a practical solution for layer (iii), which is not the main objective of this paper, is presented in Section 3.3.

3.1. Discrete Planning

The first layer of our approach aims at finding a rough path between $\mathbf{q}_0$ and $\mathbf{q}_g$. This is accomplished by well-known and efficient graph search algorithms. We start by representing the thematic map of the environment using a planar map. A proper data structure to computationally represent this kind of map is a *planar straight line graph* (PSLG; Gangnet et al. (1989)). A PSLG subdivides the plane into vertices, edges, and faces. Since a PSLG assumes that each face is a polygonal region, circular regions and arcs must be approximated by a set of line segments. In the case of large and sparsely occupied workspaces this is not an important constraint, as shown in Section 4.

The way we represent the cost to traverse a given region in a specific thematic map is similar to Mitchell and Papadimitriou (1991). The face, $f_i \subset \mathbb{R}^2$, represents a specific region of the thematic map and has associated to it a traversing cost $c_i \in [0, +\infty]$ per unit distance. Therefore, the cost function $g(x, y)$ presented in the previous section can be defined as

$$g(x, y) = c_i \quad \forall (x, y) \in f_i, \quad i = 1, 2, \ldots, m, \tag{2}$$

where m is the number of faces in the map $\mathcal{W}$. Note that $g(x, y)$ is chosen to be discontinuous since, in practice, neighboring regions may have very different characteristics (e.g. two distinct types of terrain with an abrupt transition between them).

The total cost $\psi(\mathbf{q}_1, \mathbf{q}_2)$ of a line segment that links two points, $\mathbf{q}_1$ and $\mathbf{q}_2$, placed in face f_i is given by

$$\psi(\mathbf{q}_1, \mathbf{q}_2) = c_i |\mathbf{q}_1 \mathbf{q}_2|, \tag{3}$$

where $|\mathbf{q}_1 \mathbf{q}_2|$ is the Euclidean distance between them. However, Equation (3) is only valid for any point of f_i if this face is convex. In this paper we guarantee convexity by subdividing each non-convex face into a set of triangles. Each triangle is considered to be a new face of the map. Actually, to avoid an extra test for convexity and to maintain homogeneity, we discretize all faces of the map using the CDT (Chew, 1987; Shewchuk, 1996). CDT maintains conformity with the original boundaries of the faces.

In order to plan a path from the initial position $\mathbf{q}_0 \in f_0$ to the goal position $\mathbf{q}_g \in f_g$ we first create a graph $G(\mathcal{V}, \mathcal{E})$, where the set of graph nodes $\mathcal{V}$ is composed by the midpoints of each map edge and points $\mathbf{q}_0$ and $\mathbf{q}_g$, and the segments linking the nodes at the same face constitute the edge set $\mathcal{E}$. The costs associated with the graph edges are computed using Equation (3), where $\mathbf{q}_1$ and $\mathbf{q}_2$ are replaced by the positions of the nodes.

Our continuous problem may be now transformed into a discrete problem stated as follows: find the path of minimum cost from $\mathbf{q}_0$ to $\mathbf{q}_g$ in the graph $G(\mathcal{V}, \mathcal{E})$, given the edge costs ψ given by Equation (3). This path searching can be performed by well-known algorithms such as A^* (Russell and Norvig, 2003) or Dijkstra (Dijkstra, 1959). The latter guarantees the optimal path for the given graph. Note that we do not find the optimal path for the continuous problem, but for the discrete problem.

Figure 1 shows an example of the methodology applied to a typical outdoor environment. Figure 1(a) presents an outdoor environment. Figure 1(b) is a (already triangulated) planar map representation of the environment. Note that it is constituted from a set of polygonal regions. In this figure, the higher the cost of the region, the darker its representation. Small details of the environment, such as trees and electrical poles as well as moving objects, are not considered in this step of the methodology. Figure 1(c) shows a graph over the triangulated environment and the optimal path on this graph.

Although the previous methodology is able to compute a path from $\mathbf{q}_0$ to $\mathbf{q}_g$ (Fonseca et al., 2005), the next layer of our approach, described in the next section, only relies on a sequence of adjacent triangles between f_0 and f_g. Figure 1(d) shows the chosen sequence that contains the optimal graph path and is obtained directly from the graph search algorithm.

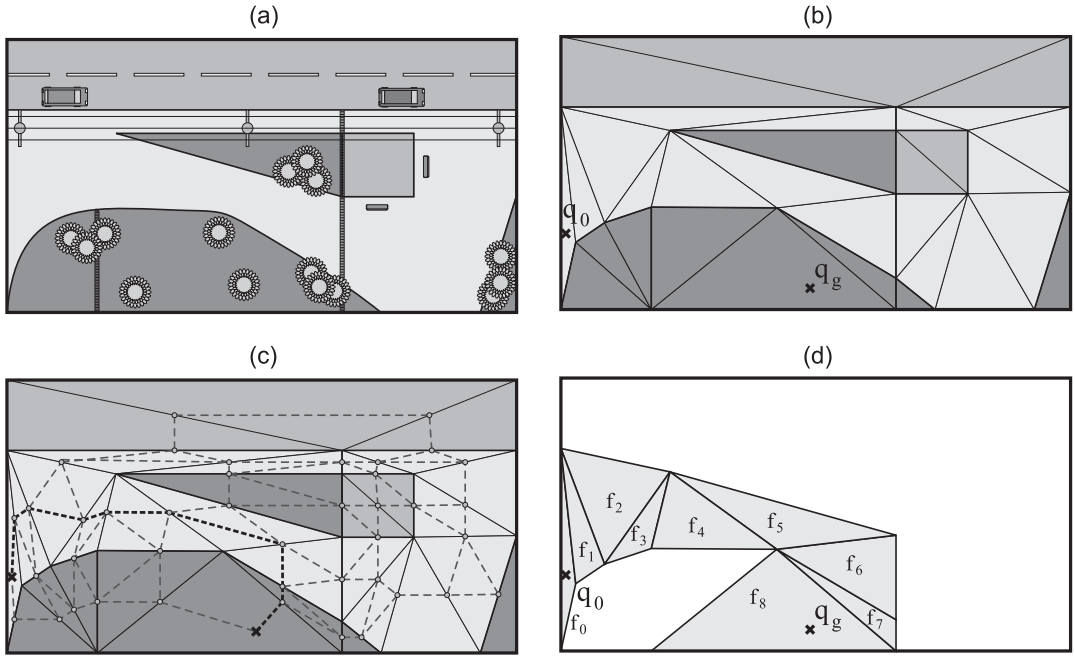

Fig. 1. Example of the planning methodology. (a) A typical outdoor environment composed by different terrains containing bushes, electrical poles, etc. (b) A triangulated planar map representation of the environment. Traversing costs were mapped to different gray levels. Here $\mathbf{q}_0$ and $\mathbf{q}_g$ are the initial and goal position, respectively. (c) A search graph and the best path on the graph. (d) The resulting sequence of triangles.

3.2. Vector Field Computation

The second layer of the proposed approach is responsible for computing a continuous vector field inside the sequence of triangles resulting from the previous level. The integral curves of this vector field must converge to $\mathbf{q}_g$.

Let $\mathcal{S} = \{f_0, f_1, f_2, \ldots, f_n\}$ be an ordered sequence of n consecutive triangles f_i, where $f_n = f_g$ (see Figures 1(d) and 3(a), for examples). The edges at the boundary of this sequence constitute the set $\mathcal{B}$. The common edges between any two triangles form the set $\mathcal{X}$. We want to build a continuous vector field $\mathbf{u}(\mathbf{q})$ that drives a holonomic robot with kinematics $\dot{\mathbf{q}} = \mathbf{u}(\mathbf{q})$ from $\mathbf{q}_0 \in f_0$ to $\mathbf{q}_g \in f_n$, that fulfills the following requirements:

(R1) for any time t, $\mathbf{q}(t) \in f_i$ and $f_i \in \mathcal{S}$;

(R2) i increases monotonically.

Some approaches have already been proposed to compute a vector field with the previous requirements (Conner et al., 2003; Belta et al., 2005; Lindemann and LaValle, 2005). The methodology used in this paper is based on the interpolation of a set of vectors at the triangles' vertices. This idea was first introduced by Belta et al. (2005), where the authors propose a methodology to compute a piecewise continuous field. In this paper we adopt a similar methodology that produces fully continuous vector fields. This methodology was initially proposed by Pimenta et al. (2007).

Suppose that each vertex of the triangles in $\mathcal{S} \setminus f_n$ has an associated base vector, which satisfies the following constraints:

(C1) its projection on the outward normal vector of an incident edge is negative or null if the edge is in $\mathcal{B}$; and

(C2) its projection on the outward normal vector of an incident edge is positive if the edge is in $\mathcal{X}$.

Call $\mathbf{v}_{\mathbf{q}_i}$, $\mathbf{v}_{\mathbf{q}_j}$, and $\mathbf{v}_{\mathbf{q}_k}$ the base vectors at vertices (x_i, y_i), (x_j, y_j), and (x_k, y_k) of f_l, counterclockwise ordered. A vector field, $\mathbf{u}(\mathbf{q})$, that simultaneously fulfills requirements (R1) and (R2) can be computed by the convex combination of these vectors given by

$$\mathbf{u}(\mathbf{q}) = \frac{A_i \mathbf{v}_{\mathbf{q}_i} + A_j \mathbf{v}_{\mathbf{q}_j} + A_k \mathbf{v}_{\mathbf{q}_k}}{A_i + A_j + A_k}, \tag{4}$$

where A_i, A_j, and A_k are the areas of the three triangles constructed by connecting the vertices $\mathbf{q}_i$, $\mathbf{q}_j$, and $\mathbf{q}_k$ of a given triangle and point $\mathbf{q}$ as shown in Figure 2.

Since $\mathbf{v}_{\mathbf{q}_i}$, $\mathbf{v}_{\mathbf{q}_j}$ and $\mathbf{v}_{\mathbf{q}_k}$ satisfy constraints (C1) and (C2), it is straightforward that $\mathbf{u}(\mathbf{q})$ never drives the robot outside the sequence of triangles (requirement (R1)) and always moves the

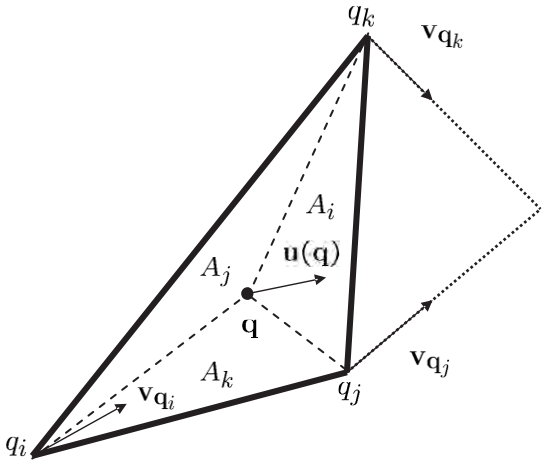

Fig. 2. The convex combination of vectors $\mathbf{v}_{\mathbf{q}_i}$, $\mathbf{v}_{\mathbf{q}_j}$, and $\mathbf{v}_{\mathbf{q}_k}$ generates a vector field $\mathbf{u}(\mathbf{q})$ as in Equation (4).

robot to the next triangle in the sequence (requirement (R2)). Observe that, since two adjacent triangles share two vertices (with the same associated vectors) at the common edge and the interpolation is linear and depends only on these two vectors along this edge, the interpolated vector field is continuous. For example, consider that configuration $\mathbf{q}$ in Figure 2 is moved to the common edge $\overline{\mathbf{q}_j\mathbf{q}_k}$. In this situation A_i vanishes and, therefore, $\mathbf{u}(\mathbf{q})$ depends only on $\mathbf{v}_{\mathbf{q}_j}$ and $\mathbf{v}_{\mathbf{q}_k}$ (see Equation (4)). Discontinuity only appears when a single vector cannot be used at one of the common vertices, as shown in the following. The next section proposes a methodology to correctly choose the base vectors $\mathbf{v}_{\mathbf{q}_l}$, $l = i, j$, and k.

3.2.1. Base Vector Computation

Initially, except for f_0 and f_n, each vector $\mathbf{v}_{\mathbf{q}_l}$ of f_i is chosen to be parallel to one of the incident edges of vertex l that are in $\mathcal{B}$, pointing towards the direction determined by the sequence of triangles. Note that we need to choose between two vectors (see Figure 3). The chosen vector is the vector that simultaneously satisfies constraints (C1) and (C2). Except for collinear incident edges, just one vector will satisfy both constraints. When none of the vectors satisfies constraints (C1) and (C2) it can be proved that no other fixed base vector can satisfy these constraints simultaneously (Belta et al., 2005). This fact is inherent to the geometry of the problem. A simple solution is to split the sequence of triangles into two subsequences, resulting in a discontinuity in the vector field.

To avoid discontinuity, when no single vector at vertex $\mathbf{q}_j$ satisfies constraints (C1) and (C2) simultaneously, we subdivide the triangle where the problem appears into two or three triangles and consider a rotating base vector (i.e. a continuous set of vectors) at $\mathbf{q}_j$. Figure 4(a) presents the case where the robot must move from triangle $f_{r-1} = \mathbf{q}_i\mathbf{q}_a\mathbf{q}_j$ to $f_r = \mathbf{q}_i\mathbf{q}_j\mathbf{q}_k$. In this case we could subdivide f_r into two triangles $T_1 = \mathbf{q}_i\mathbf{q}_j\mathbf{q}_m$ and $T_2 = \mathbf{q}_m\mathbf{q}_j\mathbf{q}_k$. Inside T_1 we make $\mathbf{v}_{\mathbf{q}_j} = \alpha(\mathbf{q}_m - \mathbf{q}_j)/\|\mathbf{q}_m - \mathbf{q}_j\|$, where α is a positive constant that determines the vector magnitude. Note that $\overline{\mathbf{q}_j\mathbf{q}_m}$ is the extension of the edge of f_{r-1} that is incident to $\mathbf{q}_j$ ($\overline{\mathbf{q}_a\mathbf{q}_j} \in \mathcal{B}$ in Figure 4). Inside T_2 and inside the next incident triangles $f_{r+1} \ldots f_{r+h}$ to $\mathbf{q}_j$, we make $\mathbf{v}_{\mathbf{q}_j}$ the rotating base vector $\alpha(\mathbf{q}-\mathbf{q}_j)/\|\mathbf{q}-\mathbf{q}_j\|$, where $\mathbf{q}$ is the robot configuration. At $\mathbf{q}_m$ the vector is $\alpha(\mathbf{q}_m-\mathbf{q}_i)/\|\mathbf{q}_m-\mathbf{q}_i\|$, the same as in the triangles where we do not have a problem. Since at the interface between T_1 and T_2, $(\mathbf{q}-\mathbf{q}_j)$ is parallel to $(\mathbf{q}_m-\mathbf{q}_j)$, continuity is guaranteed.

Figure 4(b) shows a case where it is necessary to subdivide f_r into three triangles. In this case, T_1 is formed just like before, and T_2 and T_3 are formed by the inclusion of a node $\mathbf{q}_n$ such that $(\mathbf{q}_j - \mathbf{q}_n)$ is parallel to $\mathbf{v}_{\mathbf{q}_k}$. The base vector at $\mathbf{q}_n$ is obviously equal to that at $\mathbf{q}_m$, ($\mathbf{v}_{\mathbf{q}_n} = \mathbf{v}_{\mathbf{q}_m}$). Inside triangles T_2 and T_3, and also inside the next incident triangles to $\mathbf{q}_j$, $f_{r+1} \ldots f_{r+h}$, we use the rotating base vector $\alpha(\mathbf{q} - \mathbf{q}_j)/\|\mathbf{q} - \mathbf{q}_j\|$.

The decision of whether to split triangle f_r into two or three triangles is based on a simple test: if the unitary vector in the direction of $(-\mathbf{v}_{\mathbf{q}_k})$ can be written as a convex combination of the unitary vectors in directions $(\mathbf{q}_m - \mathbf{q}_j)$ and $(\mathbf{q}_k - \mathbf{q}_j)$, then we use three triangles; otherwise we use two. The reason for this will become clear during the proofs in the next section.

For triangle f_0, one of the vertices is not shared by other triangles (see Figure 3(a)). The vector at this vertex can be any convex combination of the direction vectors of the incident edges. For f_n, since the robot must stop at $\mathbf{q}_g$, the vectors at each vertex l are computed as $\beta(\mathbf{q}_g - (x_l, y_l))$, where β is a positive constant that is explained in the following (see Figure 3(b)).

So far, we have emphasized the direction of the base vectors $\mathbf{v}_{\mathbf{q}_l}$, but nothing has been said about their magnitudes. Since we are working in multi-terrain environments, the magnitudes of the vectors can be used to determine the robot velocity in different terrains. Thus, each triangle f_i (which, by definition, represents a single terrain) is associated with a maximum robot velocity. The magnitude α of the vector $\mathbf{v}_{\mathbf{q}_l}$ is then defined as the minimum velocity associated with each of the incident triangles to vertex l. Using Equation (4), we guarantee that the maximum robot velocity inside each triangle is at most equal to the maximum velocity associated to each triangle. It is important to mention that we must maintain the ratio among the magnitudes of $\mathbf{v}_{\mathbf{q}_i}$, $\mathbf{v}_{\mathbf{q}_j}$, and $\mathbf{v}_{\mathbf{q}_k}$ of the last triangle f_n, in order to guarantee that the robot will reach the goal. This ratio can be maintained by normalizing the base vectors of the last triangle according to the magnitude of the largest one and multiplying such normalized vectors by α, thus determining the parameter β presented before.

Finally, one could claim that it is possible to have a rotating base vector or even a discontinuity problem at the last trian-

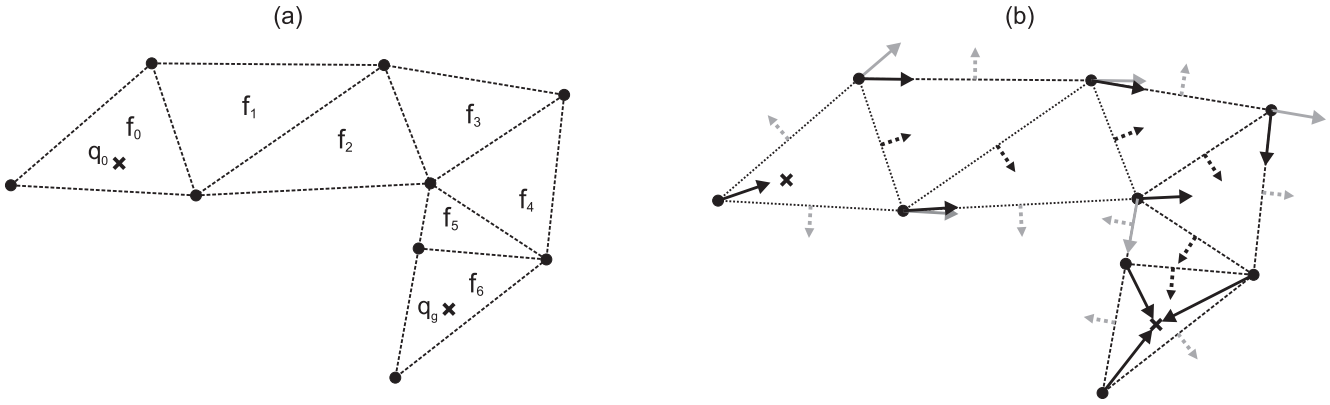

Fig. 3. (a) Typical sequence of triangles. Here $\mathbf{q}_0$ and $\mathbf{q}_g$ represent initial and goal position, respectively. (b) Choice of the vector field base vectors. Dashed gray normal vectors are constraints associated to the boundary edges (edges in $\mathcal{B}$). These constraints are used to guarantee that the robot never leaves the sequence of triangles. Dashed black vectors are constraints associated to the output edges of each triangle (edges in $\mathcal{X}$). These constraints are used to make sure the robot will never move backwards. Black and gray continuous vectors indicate the vectors that respect and violate the constraints, respectively.

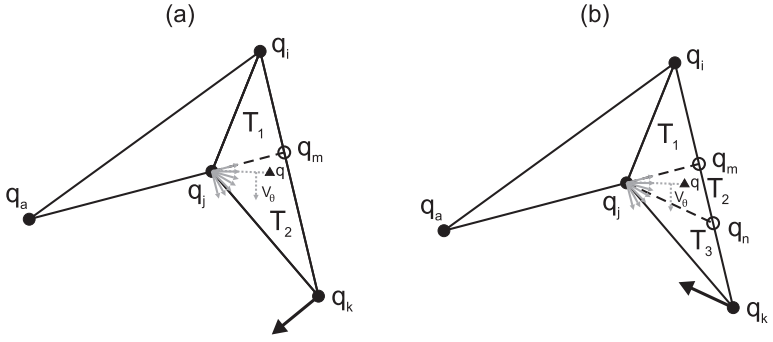

Fig. 4. Subdivision of triangle f_r into (a) two and (b) three triangles.

gle, and then it would be impossible to assign the correct base vectors in order to properly drive the robot to the goal. If these situations occur, it is always possible to subdivide the last triangle in such a way that the target is isolated in a new triangle that is free of such a problem.

3.2.2. Mathematical Analysis

As we mentioned before, by the simple fact that two adjacent triangles share two vertices (with the same associated vectors) at the common edge, and the interpolation is linear along this edge, the interpolated vector field is continuous. Moreover, since $\mathbf{q}(t)$ changes continuously, it is obvious that the inclusion of rotating vectors that points toward the current robot configuration does not introduce discontinuity into the vector field. Therefore, it is already clear that the objective of having a fully continuous vector field is accomplished by the proposed algorithm. In this section, we prove that our algorithm properly drives the robot to the goal.

Before proving the efficacy of the proposed vector field it is necessary to present a geometric property of the triangles, which we have formulated as a lemma.

Lemma 1. *Given any triangle $\mathbf{q}_i\mathbf{q}_j\mathbf{q}_k$, counterclockwise ordered, it is possible to guarantee that $(\mathbf{q}_j - \mathbf{q}_i)\cdot\mathbf{v}_e > 0$ and $(\mathbf{q}_k - \mathbf{q}_i)\cdot\mathbf{v}_e > 0$, where $\mathbf{v}_e$ is orthogonal to $(\mathbf{q}_k - \mathbf{q}_j)$ and points the triangle outwards.*

Proof. To prove that $(\mathbf{q}_j - \mathbf{q}_i)\cdot\mathbf{v}_e > 0$ we first write the vector $\mathbf{v}_e$ in terms of both the vector $(\mathbf{q}_j - \mathbf{q}_i)$ and its projection on $(\mathbf{q}_j - \mathbf{q}_k)$ (see Figure 5):

$$\mathbf{v}_e = \gamma\left[(\mathbf{q}_j - \mathbf{q}_i) - (\mathbf{q}_j - \mathbf{q}_i)\cdot(\mathbf{q}_j - \mathbf{q}_k)\frac{(\mathbf{q}_j - \mathbf{q}_k)}{\|\mathbf{q}_j - \mathbf{q}_k\|^2}\right], \quad (5)$$

where γ is a positive constant related to the magnitude of $\mathbf{v}_e$.

By replacing $\mathbf{v}_e$ in $(\mathbf{q}_j - \mathbf{q}_i)\cdot\mathbf{v}_e$ by Equation (5) we obtain

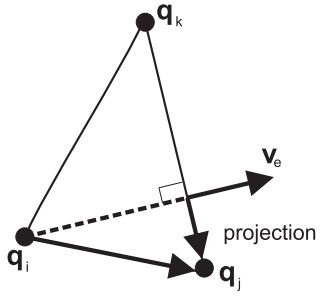

Fig. 5. A counterclockwise ordered triangle $\mathbf{q}_i\mathbf{q}_j\mathbf{q}_k$. This figure is useful in understanding the proof of Lemma 1.

$$(\mathbf{q}_j - \mathbf{q}_i)\cdot\mathbf{v}_e = \gamma\left\{\|\mathbf{q}_j - \mathbf{q}_i\|^2 - \left[\frac{(\mathbf{q}_j-\mathbf{q}_i)\cdot(\mathbf{q}_j-\mathbf{q}_k)}{\|\mathbf{q}_j-\mathbf{q}_k\|}\right]^2\right\} > 0, \quad (6)$$

since the second term on the right-hand side of the equation is always smaller than the first. The first term is the squared norm of a given vector $\mathbf{v}$, and the second term is the squared norm of the projection of $\mathbf{v}$ in a non-collinear unitary vector. If we replace the index j with k, and k with j in the above expressions, then it follows that $(\mathbf{q}_k - \mathbf{q}_i)\cdot\mathbf{v}_e > 0$. □

We have now the following proposition.

Proposition 1. *Given a holonomic robot with kinematics* $\dot{\mathbf{q}} = \mathbf{u}(\mathbf{q})$, *the vector field* $\mathbf{u}(\mathbf{q})$ *computed using Equation (4) with base vectors as in Section 3.2.1 attends requirements* (R1) *and* (R2).

Proof. If no continuity problem appears in the sequence, our algorithm always chooses base vectors which attend constraints (C1) and (C2). Therefore, since the vector field inside the triangles is generated by a convex combination of the incident base vectors (Equation (4)), it is guaranteed that the robot will stay inside the sequence of triangles and will move to the next triangle. We still have to prove that this is also true when a continuity problem appears.

Since rotating vectors never point outwards the sequence of triangles, it is clear that constraint (C1) is satisfied. Thus, to finish this proof we only need to show that the robot always moves to the next triangle when it is inside the triangles incident to the vertices where rotating vectors appear.

Suppose that a continuity problem happened in a triangle $\mathbf{q}_i\mathbf{q}_j\mathbf{q}_k$, the robot must move towards the edge $\overline{\mathbf{q}_j\mathbf{q}_k}$, and a rotating vector is assigned to $\mathbf{q}_j$. Assume this is the case when three triangles are used to subdivide the original problematic triangle (see Figure 4(b)). So, inside T_1 we can write that $\mathbf{v}_{\mathbf{q}_i} = \gamma\,\lambda(\mathbf{q}_j-\mathbf{q}_i)+\gamma\,(1-\lambda)(\mathbf{q}_k-\mathbf{q}_i)$, where γ is a positive constant related to the magnitude of the vector and $\lambda \in [0;1]$, $\mathbf{v}_{\mathbf{q}_m} = \gamma\,(\mathbf{q}_k - \mathbf{q}_i)$, and $\mathbf{v}_{\mathbf{q}_j} = \gamma\,(\mathbf{q}_m - \mathbf{q}_j)$. By using Lemma 1 it is clear that $\mathbf{v}_{\mathbf{q}_i}\cdot\mathbf{v}_e > 0$, $\mathbf{v}_{\mathbf{q}_j}\cdot\mathbf{v}_e = 0$, and $\mathbf{v}_{\mathbf{q}_m}\cdot\mathbf{v}_e > 0$. Therefore, constraint (C2) is satisfied. In triangle T_2, suppose that there is a vector $\mathbf{V}_\theta$ perpendicular to the rotating vector pointing in direction of the rotation (see Figure 4(b)). If we prove that $\mathbf{u}(\mathbf{q})\cdot\mathbf{V}_\theta > 0$, then it is proved that the robot will reach the next triangle. We can write that $\mathbf{v}_{\mathbf{q}_n} = \gamma\,(\mathbf{q}_k - \mathbf{q}_i)$. Since it is possible to form triangles inside T_2 where $\mathbf{V}_\theta = \mathbf{v}_e$, by using Lemma 1, $\mathbf{v}_{\mathbf{q}_m}\cdot\mathbf{V}_\theta > 0$, $\mathbf{v}_{\mathbf{q}_j}\cdot\mathbf{V}_\theta = 0$, and $\mathbf{v}_{\mathbf{q}_n}\cdot\mathbf{V}_\theta > 0$. Therefore, $\mathbf{u}(\mathbf{q})\cdot\mathbf{V}_\theta > 0$ as desired. Now, we use the same argument in T_3. Since $\mathbf{v}_{\mathbf{q}_k} = \gamma\,(\mathbf{q}_j - \mathbf{q}_n)$, by using Lemma 1, $\mathbf{u}(\mathbf{q})\cdot\mathbf{V}_\theta > 0$. It is straightforward to check that this argument is also true inside the next triangles incident to $\mathbf{q}_j$.

Suppose now we are in the case where two triangles are used to subdivide the original problematic triangle (see Figure 4(a)). Obviously, except for T_2, the analysis is the same as developed in the previous paragraph. Inside T_2 we can write $\mathbf{v}_{\mathbf{q}_k} = \gamma\,\lambda(\mathbf{q}_j - \mathbf{q}_m) + \gamma\,(1-\lambda)(\mathbf{q}_k - \mathbf{q}_i)$, where $\lambda \in [0;1]$. Since it is possible to form triangles inside T_2 where $\mathbf{V}_\theta = \mathbf{v}_e$, by using Lemma 1, $\mathbf{v}_{\mathbf{q}_k}\cdot\mathbf{V}_\theta > 0$, $\mathbf{v}_{\mathbf{q}_j}\cdot\mathbf{V}_\theta = 0$, and $\mathbf{v}_{\mathbf{q}_m}\cdot\mathbf{V}_\theta > 0$. Therefore, $\mathbf{u}(\mathbf{q})\cdot\mathbf{V}_\theta > 0$ and the proof is now complete. □

The previous proposition shows that a robot following $\mathbf{u}(\mathbf{q})$ always reaches the triangle that contains the target configuration $\mathbf{q}_g$. The following proposition guarantees it reaches $\mathbf{q}_g$.

Proposition 2. *The vector field in the last triangle converges to the goal.*

Proof. This proposition can be proved by showing that $\mathbf{u}(\mathbf{q})\cdot(\mathbf{q}_g - \mathbf{q}) \geq 0$, $\forall\mathbf{q} \in f_n$. If we replace the area terms, A_l, $l = i, j, k$, in Equation (4) by the corresponding determinants, replace the base vectors by $\beta(\mathbf{q}_g - (x_l, y_l))$, where β is a positive constant related to the normalization of the vectors (see Section 3.2.1), and perform some simple algebra, we can show that

$$\mathbf{u}(\mathbf{q}) = \beta(\mathbf{q}_g - \mathbf{q}).$$

Therefore, $\mathbf{u}(\mathbf{q})\cdot(\mathbf{q}_g - \mathbf{q}) = \beta\|\mathbf{q}_g - \mathbf{q}\|^2 \geq 0$. Note that $\mathbf{u}(\mathbf{q})$ vanishes only at $\mathbf{q}_g$. □

3.3. Real Robot Control

The methodology, as explained thus far, assumes a pointwise, holonomic robot represented by its exact configuration $\mathbf{q}$ and working in a static environment. These assumptions do not always hold in real world. Therefore, it is natural to ask whether the methodology could be applied to control generic shaped, non-holonomic robots with estimated configuration $\hat{\mathbf{q}}$ in a dynamic environment.

Traditionally, the assumption of a pointwise robot is easily relaxed if obstacles are dilated by the size of the robot. We can

extend this idea by dilating the higher-cost faces of the map over the lower-cost regions.

To take into account non-holonomic constraints, a feedback-linearization based controller can be used to make the robot follow the vector field (Sastry, 1999). Without loss of generality, in this section we consider a differential driven robot. By defining the robot control point to be at distance d from its center of mass, we have

$$\begin{bmatrix} V \\ \omega \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\dfrac{\sin\theta}{d} & \dfrac{\cos\theta}{d} \end{bmatrix} \mathbf{u}(\mathbf{q}), \tag{7}$$

where V and ω are the linear velocity of the robot's center of mass and its angular velocity respectively, θ is the robot's orientation, and $\mathbf{u}(\mathbf{q})$ is the vector field given by Equation (4).

The estimation of $\mathbf{q}$ is one of the most difficult problems for real robots. Several good solutions have already been proposed in the literature for indoor localization, but outdoor localization is still a difficult problem. A known solution is to use recursive estimators, such as the Kalman filter and its variations, to combine information from several different sensors. The extended Kalman filter, for example, is extensively applied to combine odometry and inertial information with GPS (Global Positioning System) data (Thrun et al., 2005). In the experiments shown in the next section, our robot uses a simple combination of odometry and a gyroscope. This limits the length of the robot's path due to integration errors.

One advantage of most vector field approaches is that they are robust to small localization errors. Therefore, even if the localization estimates have small drifts and errors, and as long as these errors do not indicate that the robot is outside the sequence of triangles, the robot will move in the correct direction. However, the proposed approach in the previous sections is very sensitive to the geometry of the map. One can observe that small triangles may lead to situations where the robot can find itself outside the sequence of triangles, due to localization errors, or even leave the sequence, due to actuator errors and dynamics. In order to avoid small triangles and to obtain good practical results, small obstacles and terrain details are not included in the map. By terrain details we mean small variations in a specific kind of terrain that could be easily handled by an all-terrain robot.

Regarding dynamic and unmodeled obstacles in the environment, the vector field may be locally modified to allow obstacle avoidance. A detailed study of this particular problem is outside the scope of this work and will be the goal of future research. A promising approach, which was used in some of the experiments presented in this paper, is to find a suitable vector to be followed which has a positive projection on both the vector field and the vector normal to the obstacle, as proposed by Esposito and Kumar (2002). Although this methodology was directly used in the experimental part of this work, we observed it has failed in several specific situations. Further attention should be paid, for example, when the robot is close to the boundaries of the sequence of triangles. In this case the resulting vector must also have negative projection on the outward normal vectors of the boundaries.

3.4. Computational Complexity

The computational implementation of the motion planning algorithm proposed in this work starts with the triangulation of a given planar map. By using CDT in an environment with n vertices, this operation can be performed with time complexity $O(n\log(n))$ (Chew, 1987). CDT does not include additional vertices in the environment, therefore the number of triangle vertices is still n.

The second step of the algorithm is to generate the search graph. Graph nodes are the midpoints of each triangle edge. Given that the number of vertices is n the number of edges in the triangulation is $O(n)$ (de Berg et al., 2000). So, computing the graph nodes and the graph edges (including the edges weight) is an $O(n)$ operation.

Searching the graph using A^* or Dijkstra is an $O(e\log(v))$ operation, where e is the number of graph edges and v is the number of graph vertices. Since both e and v are $O(n)$, searching for the shortest path on the graph is an $O(n\log(n))$ operation.

The generation of the base vectors implies visiting each vertex of the sequence of triangles found by the graph searching algorithm. This number is bounded by the original number of triangulation vertices, thus is still $O(n)$.

All of the steps mentioned before consist of off-line operations and, given the analysis performed thus far, can be computed in $O(n\log(n))$. Field computation by vector interpolation and robot control can be a real-time operation, performed in constant time $O(1)$.

4. Experiments

Our experimental platform is a Pioneer 3 All Terrain mobile robot (P3AT; Figure 6). It is a four-wheel, differential driven robot equipped with encoders, gyroscope, laser range scanner, and GPS (not used in the experiments presented in this paper). A laptop running Microsoft Windows^{TM} XP was used on-board. Most of the programming was done using ARIA (Active Media Robotics, 2006), the software framework provided with the robot, which also includes a realistic simulator. The robot maximum linear velocity in a flat surface is about $0.8\ \mathrm{m\ s^{-1}}$.

The outdoor environment where experiments were conducted has an area of about 2,500 $\mathrm{m^2}$ with five distinct types of terrain: concrete, grass, gravel, and two types of cobblestone. This environment is represented by the polygonal map shown in Figure 6.

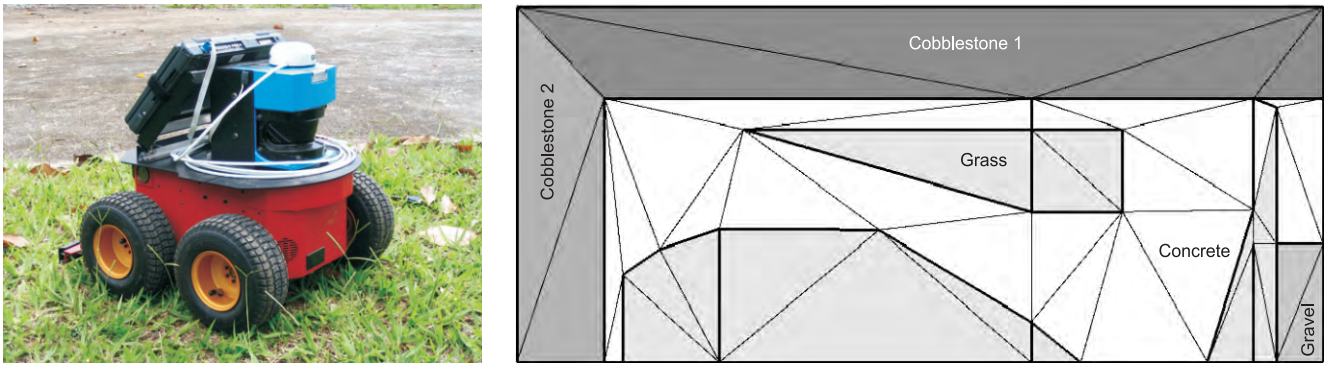

Fig. 6. Robot P3AT used in the experiments and the map representing the workspace.

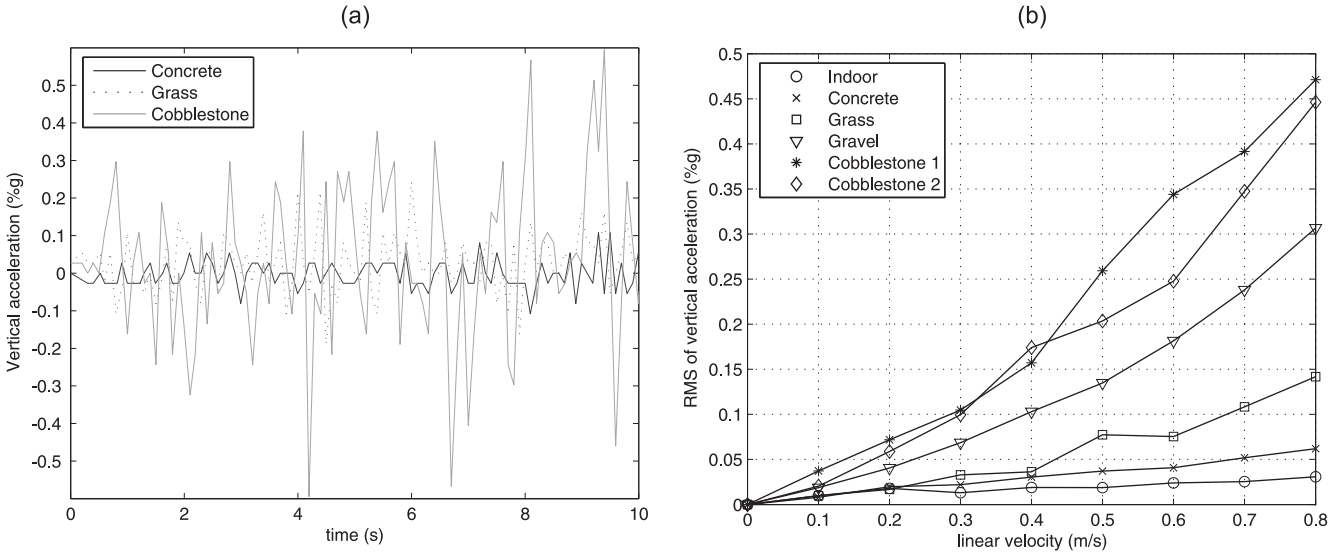

Fig. 7. (a) Vertical acceleration in different terrains for a constant velocity of $0.5m/s$ (see also Extension 1). (b) Relationship between vertical acceleration and speed for five types of terrain.

4.1. Terrain Cost Estimation

There are several ways to determine navigation costs. Our approach consists of estimating the terrain roughness by measuring the robot vibration using an accelerometer mounted vertically. A good estimate of terrain roughness is obtained with the acceleration root mean square (RMS) value. This metric is used by the automotive industry to evaluate the effectiveness of car suspensions (Bastow and Howard, 1993).

In our first experiment, the robot was programmed to traverse the five different types of terrain with several distinct velocities. Each trial resulted in a set of vertical accelerations such as those shown in Figure 7(a), for a robot linear velocity of 0.5 m s^{-1} and three types of terrain.

The RMS of the vertical accelerations for all terrains and velocities tested are depicted in Figure 7(b). All values in this graph are given as a percentage of $1g$ (about 9.8 m s^{-2}). It can be observed that the RMS value is much larger in rough terrains (e.g. gravel or cobblestone). Also, the vertical acceleration grows when velocity is increased. This graph suggests, for example, that if it is important to limit the maximum vertical acceleration (e.g. for equipment safety), the robot velocities must be limited to a maximum value for each type of terrain. In other words, the robot may go faster in smooth terrains and slower in the rough terrains. This approach is intuitive and is used by most automobile drivers. Thus, to determine costs for our weighted graph, we choose the allowed vertical acceleration in each terrain, which consequently will determine the maximum robot velocity. Costs can be computed by multiplying the inverse of the robot velocity in each face (c_i) by the graph edge length ($|\mathbf{q}_1\mathbf{q}_2|$) yielding in the edge cost (ψ), which is a metric of time that should be minimized.

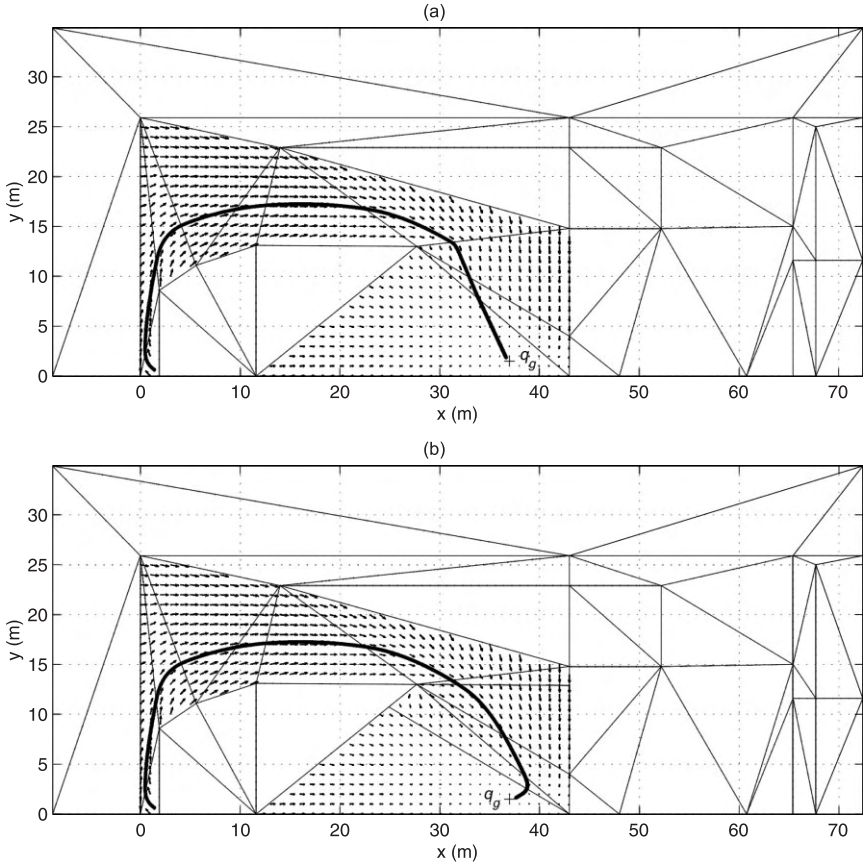

Fig. 8. Comparing fully continuous and piecewise continuous vector fields. (a) A piecewise continuous field can be obtained by splitting the sequence of triangles into two or more sequences. (b) A fully continuous vector field is computed by creating new triangles and using a rotating base vector.

4.2. Path Planning and Robot Control

In this section we show how our methodology can be used to make the robot traverse different terrains towards a prespecified final position given a limit to its vertical acceleration. Based on Figure 7(b) we chose the maximum values for robot velocities. These values were used to determine the costs for each graph edge and also to limit the magnitude of the vector field base vectors.

We first present simulation results that aim to show the efficacy of the methodology in computing fully continuous vector fields inside a discrete sequence of triangles obtained after graph minimization. Figure 8(a) presents a path followed by the robot when a piecewise continuous vector field is used. In this case the sequence of triangles was split into two new sequences. On the other hand, Figure 8(b) shows the robot path when a fully continuous vector field is used. Velocity profiles for both trajectories are presented in Figure 9. The angular velocity profile in Figure 9 presents the discontinuity at 100 s.

We also performed real robot experiments with the same sequence of triangles used in the simulations. Figure 10 shows the robot path for the continuous vector field. One can observe that the robot did not reach its exact goal position. This is an issue related to the robot hardware. As this four-wheel differential driven robot has relatively large all-terrain wheels, some lateral slippage happens when the robot is turning. This causes an extra load torque for each robot motor, which acts as a perturbation for the motor controllers. When the robot is close to the goal and the vector field has a very small magnitude, this perturbation, which is intense on the grass, prevents the motors from achieving their set-points and causes high current values on the motors. This situation is interpreted by the robot control system as a failure, which automatically stops the robot. The velocity profiles in Figure 11 shows that after a time of 125 s, the controller makes several attempts to turn but is always blocked by the system. This issue indicates that, for this specific sequence of triangles, a piecewise continuous field such that in Figure 8(a) could be a better choice. It is important to mention that the maximum robot velocity in

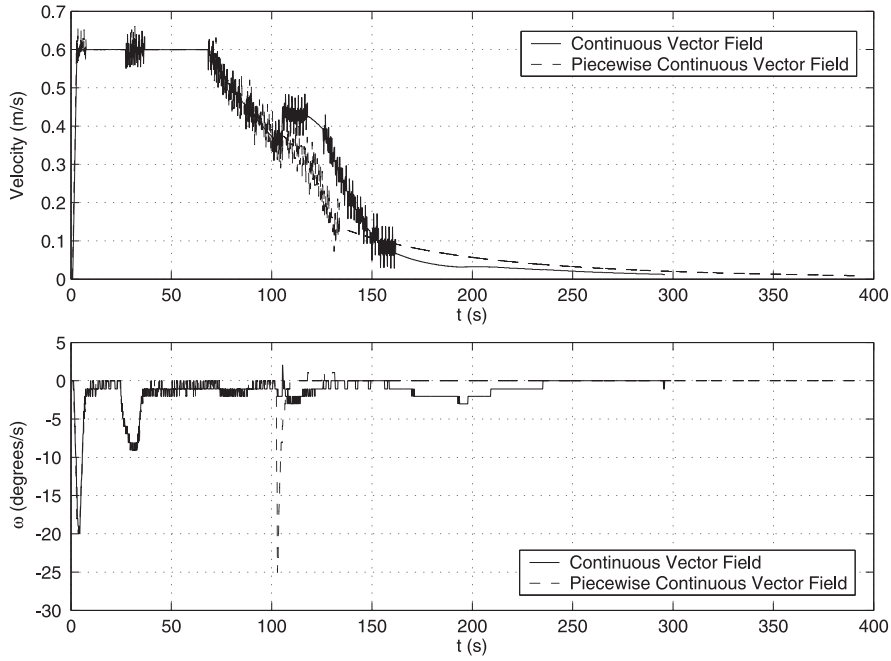

Fig. 9. Linear and angular velocity profiles for the simulated paths in Figure 8.

the simulator is 0.6 m s^{-1} while the real robot can go as fast as 0.8 m s^{-1}. This explains the main differences between the profiles in Figures 9 and 11.

The last real robot experiments presented in this section have two main objectives: (i) to show that the proposed methodology is able to bound the robot vertical acceleration; and (ii) to show that the field may be locally changed to allow for dynamic and unmodeled obstacles. Thus, in some experiments piecewise continuous vector fields were used. Figure 12(a) shows two sequences of triangles composed of different terrains. In the first sequence we have only grass and concrete (see the map in Figure 6). For this sequence of triangles, we aimed at keeping the RMS values of vertical acceleration around 0.06% of g. The maximum velocity was limited to be 0.8 m s^{-1} for concrete and 0.3 m s^{-1} for grass. A typical robot path (Υ_1) and the vector field for this sequence is also shown in Figure 12(a). Robot velocities and vertical accelerations profiles for this specific path are presented in Figure 13. The computed maximum acceleration RMS value for the path was 0.052% of g on concrete. Since a piecewise continuous vector field was used, it is possible to see the effects of a discontinuity in the field at a time of 26 s. Furthermore, observe that the robot linear velocity increases in the concrete region, but also gradually decreases before it reaches the next region (grass). This is due to the fact that both the concrete and the grass triangle share the same base vectors, which have the magnitude determined by the maximum velocity on the grass.

For the second sequence of triangles in Figure 12(a) we limited the robot velocity to 0.65 m s^{-1} for concrete, 0.2 m s^{-1} for gravel and 0.1 m s^{-1} for cobblestone, in order to bound the acceleration RMS value to approximately 0.05% of g. Robot velocities and vertical accelerations profiles for a specific path (Υ_2) in this sequence are presented in Figure 14. Maximum RMS accelerations for this path were 0.060% of g on gravel and 0.088% of g on concrete. This apparently large value of vertical acceleration is due the presence of a small amount of gravel scattered over the concrete region.

In the experimental trial presented in Figure 12(b) (see also Extension 2) the robot was exposed to dynamic and unmodeled obstacles. The dynamic obstacle was a walking person that intercepted the robot path at the point marked with a black box in Figure 12(b). The unmodeled obstacle was a small tree. This obstacle is represented by a gray box. As we mentioned before, since we chose not to have very small triangles in our sequence, we did not model the environment in all of its detail. As we see in Figure 12(b), the laser-based obstacle avoidance controller was able to handle such details well.

5. Conclusion and Research Perspectives

In this paper we have presented a low-cost, terrain-based outdoor robot motion planning methodology. Although the resulting path is not necessarily optimal, it is a low-cost path in the sense of distance and a terrain metric, which was vertical acceleration in our case. Low computational cost is mainly due to the discretization methodology used, which provides a good representation of the environment with a small number of

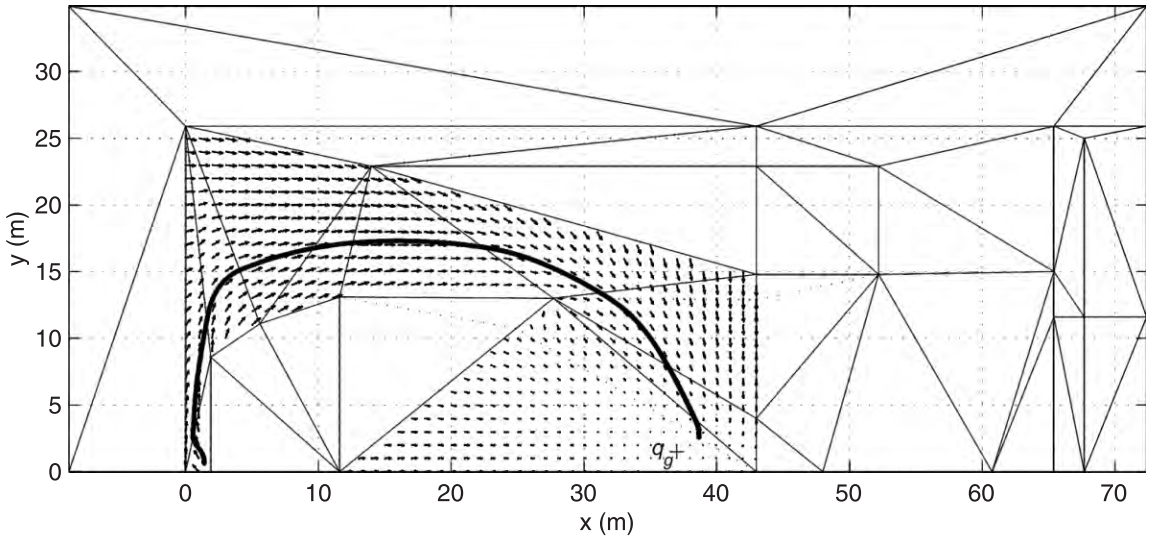

Fig. 10. Path for a real robot following an continuous vector field.

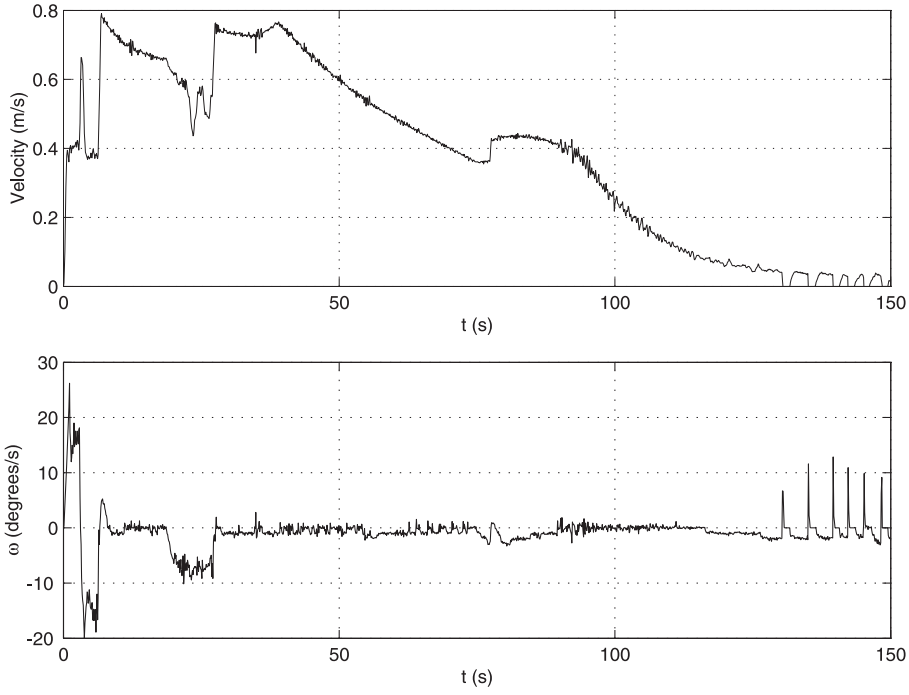

Fig. 11. Linear and angular velocity profiles for the real robot path in Figure 10.

cells. Observe in Figure 6 that only 46 triangles were needed to represent a 2,500 m^2 area. To represent the same area, a traditional methodology based on regular square cells with the size of the robot diameter (0.5 m) would require about 10,000 cells, while a quadtree representation with minimum cell size of 0.5 m would require 2,380 cells, as shown in Figure 15.

We control the robot by computing a continuous vector field over a sequence of discrete regions. The vector field can also be generated very efficiently, since it does not need to be computed for regions far from the robot path. Also, it has the advantages of many potential field approaches, such as robustness to small localization errors and the possibility of avoid-

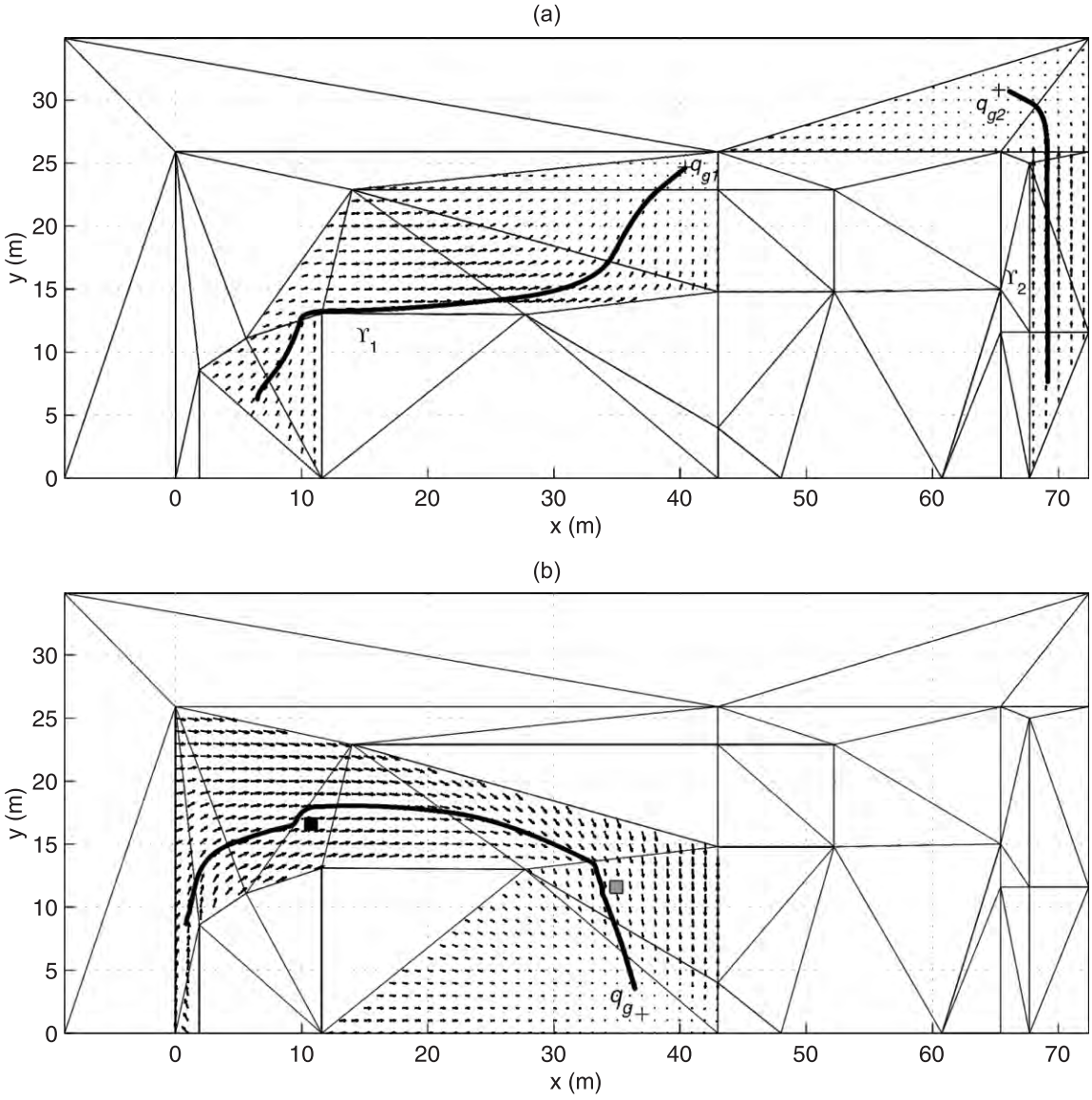

Fig. 12. (a) Two sequences of triangles and their respective vector fields. Two actual robot paths Υ_1 and Υ_2 are also overlaid on the map. (b) The small boxes represent dynamic (black box) and unmodeled (gray box) obstacles. The robot path shows that these obstacles were avoided in execution time by means of a laser-based obstacle avoidance controller (see also Extension 2).

ing unknown obstacles without replanning. This vector field also guarantees that: (i) the robot will never leave the "corridor" determined by the sequence of triangles; and (ii) at the same time, the robot will never move backwards. Other nonholonomic controllers could be used to control the robot. Since the first derivative of the field exists, a feedforward term could be included in these controllers to improve field tracking.

The main limitations of the approach are: (i) it depends on a good map of the environment, although this map does not need to have a great level of detail; (ii) the computed vector field does not guarantee that the optimal path will be followed; (iii) since triangles in the sequence share base vectors, a low maximum velocity in one face limits the maximum velocity in its neighbors.

Based on these limitations, there are important directions for future research. Although it seems intuitive that the resultant sequence of triangles is optimal, in the sense that it contains the optimal path for the continuous problem, it is easy to show a counterexample to this fact. Actually, finding the optimal path in the continuous space can be done by using an

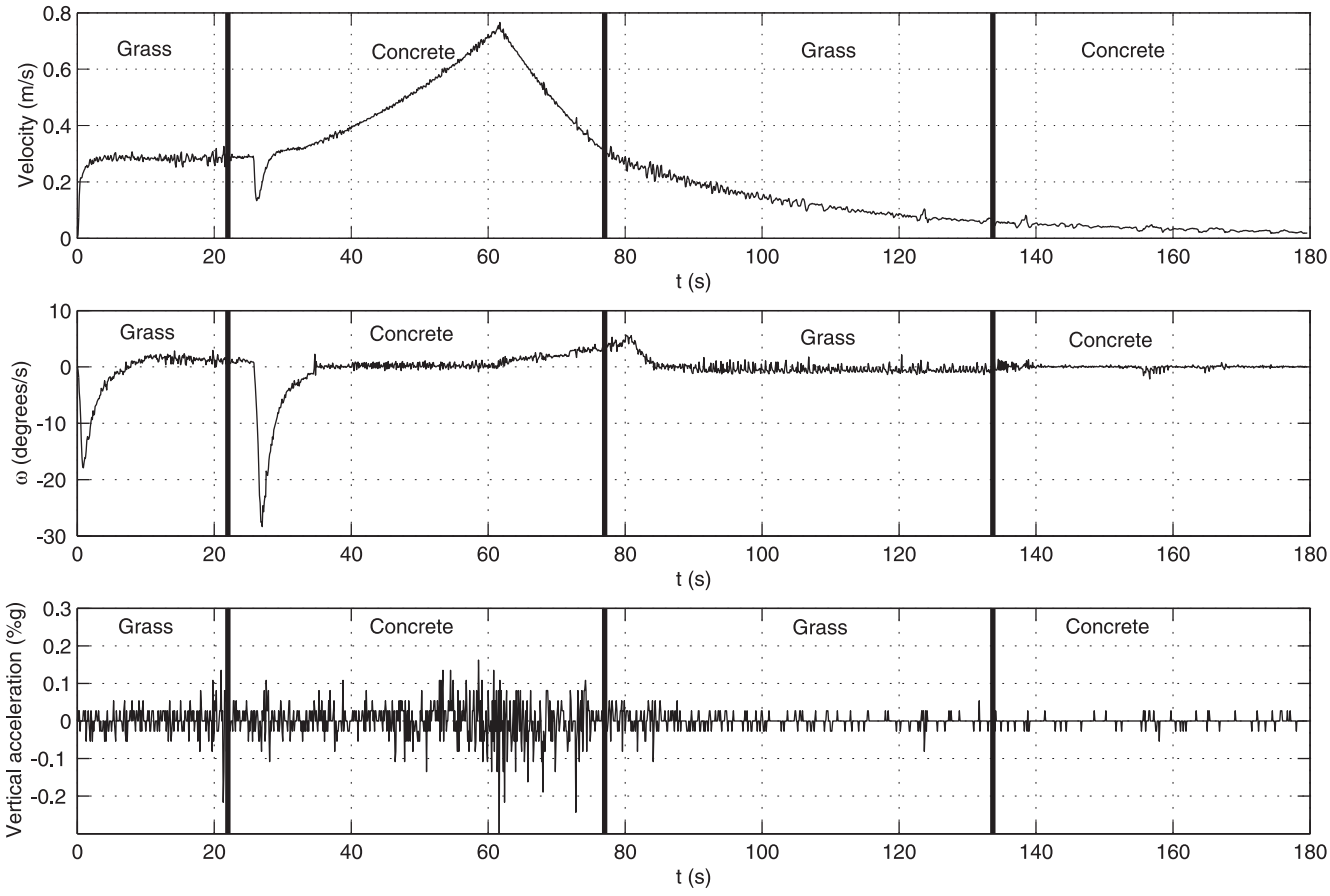

Fig. 13. Linear velocity, angular velocity (ω), and vertical acceleration profiles for path Υ_1 in Figure 12(a). In this path, the robot traversed two types of terrain.

approach based on Snell's law (Mitchell and Papadimitriou, 1991), which is very hard and time consuming. We intend to investigate efficient ways to determine the optimal sequence of triangles.

Next steps also include investigating other metrics for navigation cost and map building. An interesting metric is related to localization. Note in Figures 13 and 14 that the angular velocity variance depends on the terrain. It suggests that odometry-based localization would result in different pose estimates for different terrains.

Finally, we will pursue to extend the methodology for three-dimensional configuration spaces. This will allow both aerial and underwater robots to be controlled as well as planar robot orientation.

Acknowledgments

The authors would like to acknowledge the financial support of FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais) and CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico, Brazil).

Appendix: Index to Multimedia Extensions

The multimedia extension page is found at http://www.ijrr.org

Table of Multimedia Extensions

Extension	Type	Description
1	Video	View from a camera mounted on the robot moving on concrete at 0.5 m s^{-1}.
2	Video	Robot following a vector field and avoiding unmodeled obstacles (movie speed 5×).

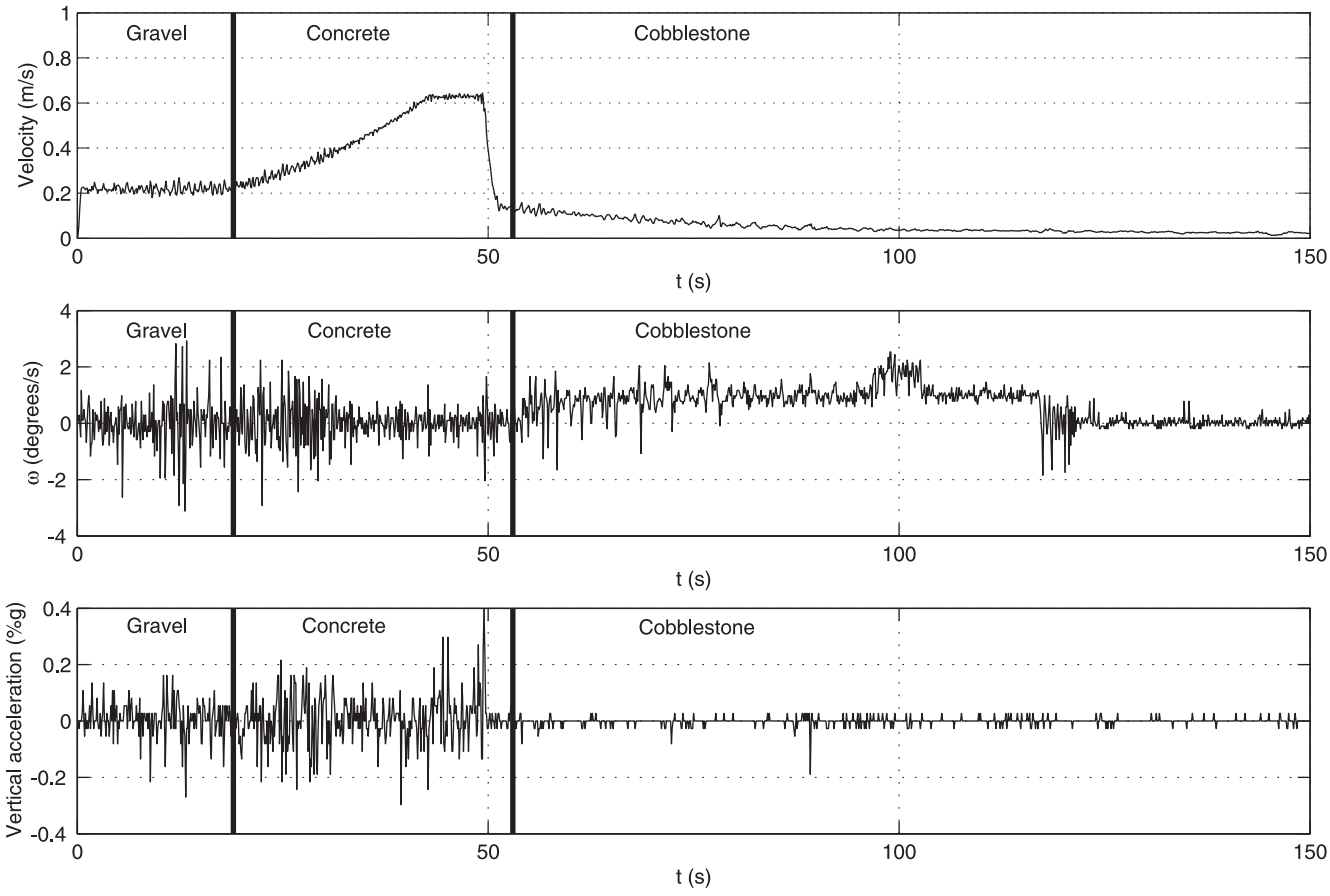

Fig. 14. Linear velocity, angular velocity (ω), and vertical acceleration profiles for path Υ_2 in Figure 12(a). In this path, the robot traversed three types of terrain.

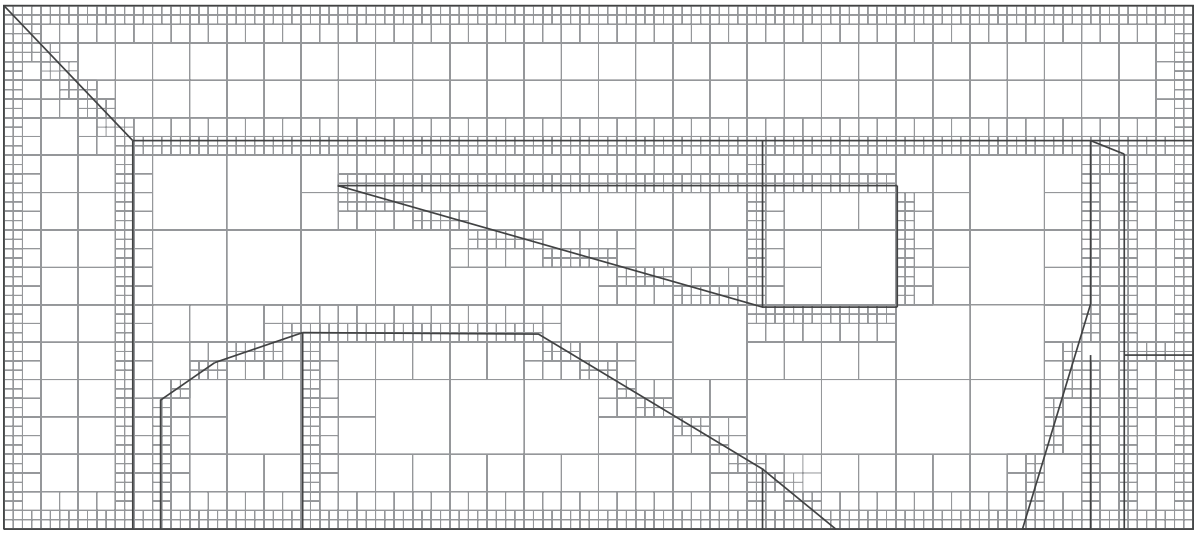

Fig. 15. Quadtree representation of the map in Figure 6. While the triangular representation of this map using CDT require only 46 cells, the quadtree representation require 2,380 cells.

References

Active Media Robotics (2006). http://www.mobilerobots.com.

Bastow, D. and Howard, G. P. (1993). *Car Suspension and Handling*. London, Pentech Press.

Belta, C., Isler, V. and Pappas, G. J. (2005). Discrete abstractions for robot motion planning and control in polygonal environments. *IEEE Transactions on Robotics*, **21**(5): 864–874.

Brooks, C. and Iagnemma, K. (2005). Vibration-based terrain classification for planetary exploration rovers. *IEEE Transactions on Robotics*, **21**(6): 1185–1191.

Chew, L. P. (1987). Constrained Delaunay triangulations. *Proceedings of the 3rd Annual Symposium on Computational Geometry*. New York, ACM Press, pp. 215–222.

Conner, D. C., Rizzi, A. A. and Choset, H. (2003). Composition of local potential functions for global robot control and navigation. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3546–3551.

de Berg, M., van Kreveld, M., Overmars, M. and Schwarzkopf, O. (2000). *Computational Geometry Algorithms and Applications*, 2nd edn. Berlin, Springer.

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

Esposito, J. M. and Kumar, V. (2002). A method for modifying closed-loop motion plans to satisfy unpredictable dynamic constraints at runtime. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1691–1696.

Fonseca, A. R., Pimenta, L. C. A., Mesquita, R. C., Saldanha, R. R. and Pereira, G. A. S. (2005). Path planning for mobile robots operating in outdoor environments using map overlay and triangular decomposition. *Proceedings of the International Congress of Mechanical Engineering*, Ouro Preto, Brazil.

Gangnet, M., Hervé, J.-C., Pudet, T. and van Thong, J.-M. (1989). Incremental computation of planar maps. *Proceedings of the 16th Annual Conference on Computer Graphics and Interactive Techniques*, pp. 345–354.

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

Lindemann, S. R. and LaValle, S. M. (2005). Smoothly blending vector fields for global robot navigation. *Proceedings of the IEEE Conference on Decision and Control*, pp. 3553–3559.

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

Ojeda, L., Borenstein, J., Witus, G. and Karlsen, R. (2006). Terrain characterization and classification with a mobile robot. *Journal of Field Robotics*, **23**(2): 103–122.

Pimenta, L. C. A., Pereira, G. A. S. and Mesquita, R. C. (2007). Fully continuous vector fields for mobile robot navigation on sequences of discrete triangular regions. *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1992–1997.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*, 2nd edn. Englewood Cliffs, NJ, Prentice-Hall.

Sadhukhan, D., Moore, C. and Collins, E. (2004). Terrain estimation using internal sensors. *Proceedings of the 10th IASTED International Conference on Robotics and Applications*, Honolulu, HI.

Sastry, S. (1999). *Nonlinear Systems: Analysis, Stability, and Control*. Berlin, Springer.

Shewchuk, J. R. (1996). Triangle: engineering a 2D quality mesh generator and Delaunay triangulator. *Applied Computational Geometry: Towards Geometric Engineering*, Lin, M. C. and Manocha, D. (eds) (*Lecture Notes in Computer Science*, Vol. 1148). Berlin, Springer, pp. 203–222.

Thrun, S., Burgard, W. and Fox, D. (2005). *Probabilistic Robotics*. Cambridge, MA, MIT Press.

Thrun, S., Montemerlo, M., Dahlkamp, H., Stavens, D., Aron, A., Diebel, J., Fong, P., Gale, J., Halpenny, M., Hoffmann, G., Lau, K., Oakley, C., Palatucci, M., Pratt, V., Stang, P., Strohband, S., Dupont, C., Jendrossek, L.-E., Koelen, C., Markey, C., Rummel, C., van Niekerk, J., Jensen, E., Alessandrini, P., Bradski, G., Davies, B., Ettinger, S., Kaehler, A., Nefian, A. and Mahoney, P. (2006). Stanley, the robot that won the DARPA Grand Challenge. *Journal of Field Robotics*, **23**(9): 661–692.

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