2007 IEEE International Conference on
Robotics and Automation
Roma, Italy, 10-14 April 2007

Fully continuous vector fields for mobile robot navigation on sequences of discrete triangular regions

Luciano C. A. Pimenta, Guilherme A. S. Pereira, and Renato C. Mesquita

Abstract— **Several recent works have combined discrete and continuous motion planning methods for robot navigation and control. The basic idea of some of these works is to plan a path, by determining a sequence of neighboring discrete regions of the configuration space, and to assign a vector field that drives the robots through these regions. This paper addresses the problem of efficiently computing vector fields over a sequence of consecutive triangles. Differently from previous numerical approaches, which were not able to compute fully continuous fields in triangulated spaces, this paper presents an algorithm that is able to compute guaranteed continuous vector fields over a sequence of adjacent triangles.**

I. Introduction

The mobile robot motion planning problem can be loosely stated as the problem of finding and executing an obstacle free path from an initial position to a pre-specified goal. In the past years, several different solutions for this problem have been proposed [1]. Among these, vector fields based methodologies, such the ones based on the gradient of artificial potential functions [2], have been extensively used, mainly because they allow the combination of planning and control in the same solution.

The main drawback of most potential field approaches is the presence of local minima in the function that originated the field, which may prevent the robot to reach the target. Although, some free of local minima potential functions can be determined [3], [4], [5], in order to improve efficiency, some recent works have constructed vector fields by combining discrete and continuous algorithms. Among these works, [6], [7], [8], [9], [10] and [11] have proposed similar solutions that are basically divided in three parts: (i) a polygonal decomposition of the robot configuration space is performed, resulting in finite number of cells; (ii) a sequence of consecutive cells connecting initial and goal configuration is selected using a discrete algorithm; (iii) a vector field is assigned inside the selected cells.

In this context, the proposal of [9] is to triangulate the configuration space and to determine the best sequence of consecutive triangles that connects the triangle that contains the robot initial configuration with the one that contains the target. This is performed by searching for the shortest path in the graph that has the barycenter of the triangles as nodes and the segment that connect the nodes as edges. Using the efficient methodology proposed in [6] the authors compute a *piecewise continuous* vector field that is able to drive the robot from the initial triangle to the last one. Bounds for the robot velocity inside each triangular cell are automatically imposed by the method. The same methodology was used to compute vector fields in [8] and [11]. In [8], the authors use a temporal logic algorithm to select the sequence of triangles to be traversed by the robot, and in [11], a graph search algorithm is used to find a minimum cost sequence of triangles in a multi-terrain outdoor environment. In the last one, a terrain metric is used during the vector field computation to limit the robot velocity in each terrain.

Although it is well known that triangulation is, in most cases, a better cell decomposition choice for complex configuration spaces, rectangular partitions followed by the assignment of a vector field have been used in [7]. As proposed in [6], a fully continuous vector field over a sequence of rectangular cells can be computed very efficiently. Since the methodology of the authors could not compute continuous fields on triangular cells, it was necessary to use quadtree representation to approximate the environment. Two recent methodologies that may be able to handle this task are presented in [12] and [10]. The first one is based on the analytic computation of harmonic functions, which are solutions to the Laplace's Equation, and the second one creates and composes face vector fields using bump functions. However, none of them seems to be as simple and efficient as the methodology proposed in [6], although the fields are not continuous in the latter one.

In order to allow the use of triangular decompositions of the configuration space, this work presents a numeric algorithm that computes fully continuous vector fields over sequences of triangles. The methodology presented in this paper is a natural extension of the ideas proposed by Belta *et al.* in [6] and [9]. Thus, the works [8], [9], and [11] will benefit from our results.

Next section formally defines the problem being considered. Section III presents the methodology proposed to compute fully continuous vector fields. Proofs for the methodology are presented in Section IV. Section V present two examples of fields and a brief comparison with a previous methodology. Trajectories for holonomic robots are also presented. Finally, conclusions and directions for future work are presented in Section VI.

II. Problem Definition

Let a single robot R in the world $\mathcal{W}$ be represented by the configuration $\mathbf{q}$ in its configuration space $\mathcal{C}$, and consider $\mathcal{F} \subseteq \mathcal{C}$ to be the free configuration space for R. In this paper we consider limited, two dimensional configuration spaces

L. C. A. Pimenta, G. A. S. Pereira and R. C. Mesquita are with the Department of Electrical Engineering, Universidade Federal de Minas Gerais, Av. Antônio Carlos 6627, 31270-901, Belo Horizonte, MG, Brazil. {lucpim, gpereira, renato}@cpdee.ufmg.br

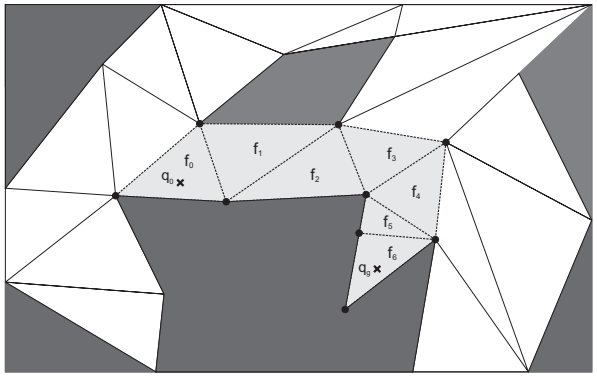

Fig. 1. A triangulated environment and sequence of triangles from f_0 to $f_g = f_6$ determined by a discrete algorithm.

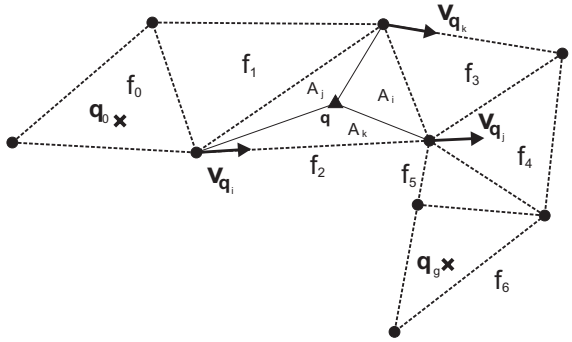

Fig. 2. The convex combination of vectors $\mathbf{v}_{\mathbf{q}_i}$, $\mathbf{v}_{\mathbf{q}_j}$, and $\mathbf{v}_{\mathbf{q}_k}$ generates a vector field $\mathbf{u}(\mathbf{q})$ as in Equation (1).

such that $\mathcal{C} = \{(x, y)|x_{min} \leq x \leq x_{max}; y_{min} \leq y \leq y_{max}\}$. At first, we also consider point holonomic robots with dynamics given by $\dot{\mathbf{q}} = \mathbf{u}(\mathbf{q})$, where $\mathbf{u}$ is the robot control input.

In a standard motion planning problem it is desirable to steer the robot from its initial configuration $\mathbf{q}_0 \in \mathcal{F}$ at time $t = t_0$ to the goal configuration $\mathbf{q}_g \in \mathcal{F}$ at some time $t = t_f > t_0$, such that $\mathbf{q} \in \mathcal{F}\ \forall t \in (t_0, t_f]$.

Now, suppose that $\mathcal{F}$ is discretized in a small number of triangular cells. This operation can be performed using several triangulation algorithms, such as the Constrained Delaunay Triangulation (CDT) [13], which were successfully used for robot navigation in [11]. CDT maximizes the internal angles of each triangle and maintains conformity with the original boundaries of $\mathcal{C}$. After the triangulation, each cell f_i is numbered from f_0 to f_n.

In order to plan a path in the triangulated configuration space, which constitutes a sequence of consecutive cells from the initial cell f_0 ($\mathbf{q}_0 \in f_0$) to the goal cell f_g ($\mathbf{q}_g \in f_g$), several discrete algorithms can be used. For instance, temporal logic is used in [8] and graph search algorithms are used in [9], [10] and [11].

For illustration, in this paper we will describe the simple algorithm used in our previous work [11]. First, we create a weighted graph $G(\mathcal{V}, \mathcal{E})$, where the set of graph nodes $\mathcal{V}$ is composed by the midpoints of each triangular edge and points $\mathbf{q}_0$ and $\mathbf{q}_g$, and the segments linking the nodes at the same cell constitute the edge set $\mathcal{E}$. The weights associated with the graph edges are the distances between them. Our continuous problem is now transformed into a discrete one stated as follows: find the path of minimum cost from $\mathbf{q}_0$ to $\mathbf{q}_g$ in the graph $G(\mathcal{V}, \mathcal{E})$. This path searching can be performed by well known algorithms such as A^* or Dijkstra. The latter one guarantees the optimal path for the given graph.

In spite of the discrete algorithm used, the result of this step is a sequence of neighboring triangular cells, as shown in Figure 1. In order to control the robot from $\mathbf{q}_0$ to $\mathbf{q}_g$ a vector field with a single vanishing point at $\mathbf{q}_g$ must be assigned to the sequence $f_0 \ldots f_g$. The computation of a guaranteed continuous vector field is the main contribution of this work and is described in the next section.

III. Vector Field Computation

The idea used in this paper to efficiently compute a vector field inside the sequence of triangles was first introduced in [6] and [9]. Basically, a set of vectors at the triangles vertices are interpolated to generate the vector field. Thus, the methodology can be divided in two steps: (i) vector interpolation, and (ii) vector determination. The contribution of this paper is in step (ii), that was improved from [9] in order to guarantee fully continuous fields in opposite to piecewise continuous ones.

Let $\mathcal{S} = \{f_0, f_1, f_2, \cdots, f_n\}$ be an ordered sequence of n consecutive triangles, where $f_n = f_g$ (see Fig. 1). The edges at the boundary of this sequence constitute the set $\mathcal{B}$. The common edges between 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 is monotonically crescent.

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

(C1) its projection on the outward normal vector of an incident edge is negative 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:

$$\mathbf{u}(\mathbf{q}) = \frac{A_i \mathbf{v}_{\mathbf{q}_i} + A_j \mathbf{v}_{\mathbf{q}_j} + A_k \mathbf{v}_{\mathbf{q}_k}}{A_i + A_j + A_k}, \qquad (1)$$

where A_i, A_j, and A_k are the areas of the sub-triangles formed between the original vertices of the triangles and the point $\mathbf{q}$ as shown in Fig. 2.

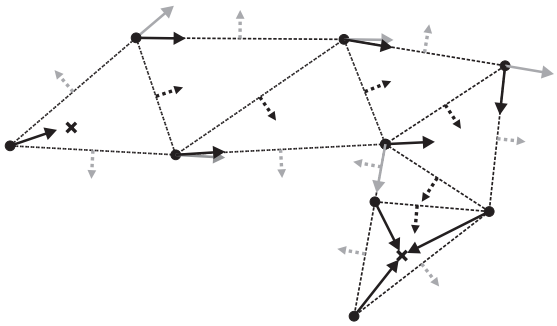

Fig. 3. Choice of the vector field base vectors. Dashed gray normal vectors are constraints associated to the boundary edges (edges in $\mathcal{B}$). These constraints are used to guarantee that the robot never leaves the sequence of triangles. Dashed black vectors are constraints associated to the 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.

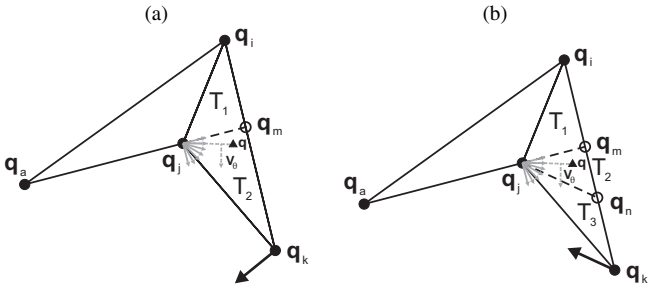

Fig. 4. Subdivision of triangle f_r in two (a) and three (b) triangles.

Once constraints (C1) and (C2) are satisfied, it is easy to prove that $\mathbf{u}(\mathbf{q})$ never drives the robot outside the corridor (requirement (R1)) and always moves the 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 along this edge, the interpolated vector field is continuous. Discontinuity only appears if a single vector cannot be used in one of the common vertices, as it will be shown next.

Differently from [9], where a linear programming technique is used to compute $\mathbf{v}_{\mathbf{q}_i}$, $\mathbf{v}_{\mathbf{q}_j}$ and $\mathbf{v}_{\mathbf{q}_k}$, in this paper the direction of these vectors are only chosen based on the geometry of the cells, and their magnitude are chosen by the maximum desirable robot velocity inside the cell. Although the algorithm in [9] yields a potentially better vector field (in the sense of shorter robot paths) when it can guarantees continuity, our method is more intuitive and less expensive. Besides these aspects, both methods are apparently interchangeable and the algorithm in [9] can easily incorporate the ideas proposed in this paper to guarantee continuity. We will now proceed to explain our technique.

Initially, except for f_0 and f_n, each vector $\mathbf{v}_l$ 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. Notice that we need to choose between two vectors (see Fig. 3). The chosen one is the vector that simultaneously satisfies constraints (C1) and (C2). Except for collinear incident edges, just one vector may 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 [9]. 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) we subdivide the triangle where the problem appears into two or three triangles and consider a continuous set of vectors, or in another point of view, a rotating base vector 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 $f_r = \mathbf{q}_i\mathbf{q}_j\mathbf{q}_k$ and 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 use $\mathbf{v}_{\mathbf{q}_j} = \alpha(\mathbf{q}_m - \mathbf{q}_j)/\|\mathbf{q}_m - \mathbf{q}_j\|$, where α is a positive constant related to the maximum desirable robot velocity. Notice 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 $f_{r+1} \dots f_{r+h}$ to $\mathbf{q}_j$, we use the rotating base vector $\alpha(\mathbf{q} - \mathbf{q}_j)/\|\mathbf{q} - \mathbf{q}_j\|$, where $\mathbf{q}$ is the robot configuration. At $\mathbf{q}_m$ we use $\alpha(\mathbf{q}_m - \mathbf{q}_i)/\|\mathbf{q}_m - \mathbf{q}_i\|$ just like in the triangles we do not have problem. Since at the interface between T_1 and T_2,$(\mathbf{q}-\mathbf{q}_j)$ is parallel to $(\mathbf{q}_m-\mathbf{q}_j)$ we guarantee continuity.

Figure 4(b) shows the case where we 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 equals to the one 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 $\mathbf{q}_j$, $f_{r+1} \dots f_{r+h}$, we use the rotating base vector $\alpha(\mathbf{q}-\mathbf{q}_j)/\|\mathbf{q}-\mathbf{q}_j\|$. The decision of splitting 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 is explained in the next section.

For triangle f_0, one of the vertices is not shared by other triangles (see Fig. 2). 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 which is explained next, (see Fig. 3).

So far, we have emphasized the direction of the base vectors. The magnitude of the base vectors $\mathbf{v}_{\mathbf{q}_l}$, except for the ones associated to the last triangle, is defined as the maximum desirable velocity α. In this way, by Equation (1), we guarantee that the maximum robot velocity inside each triangle is smaller than or equal to α. It is important to mention that we must maintain the proportion among the magnitudes of $\mathbf{v}_{\mathbf{q}_i}$, $\mathbf{v}_{\mathbf{q}_j}$, and $\mathbf{v}_{\mathbf{q}_k}$ of f_n, in order to guarantee that the robot will reach the goal. This proportion 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, determining then 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 triangle, and then it would be impossible to assign the correct base vectors in order to properly drive the robot to the goal. If such situations occur, it is always possible to subdivide the last triangle in such a way that the target is isolated in a new triangle free of those problems.

IV. Mathematical Analysis

As we said 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. More over, 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 to the vector field. Therefore, at this point it is already clear that the objective of having a fully continuous vector field is accomplished by our algorithm. In this section, we aim to prove that our algorithm properly drives the robot to the goal. This means that the robot must reach the last triangle without moving outside the corridor (requirement (R1)) and always moving to the next triangle in the sequence (requirement (R2)). Also, once in the last triangle the robot must always move towards the goal.

We start with a geometric property of triangles which will be shown to be very useful. We are not going to present a proof of this property, but we believe it is quite trivial to see it is true, since triangles cannot have parallel and collinear edges.

Property 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 $\mathbf{v}_e \perp (\mathbf{q}_k - \mathbf{q}_j)$ and points outward the triangle.*

Proposition 1 *Given a holonomic robot with kinematics $\dot{\mathbf{q}} = \mathbf{u}(\mathbf{q})$, the vector field $\mathbf{u}(\mathbf{q})$ computed by the algorithm proposed in Section III attends the requirements (R1) and (R2).*

Proof: If no continuity problem appears in the sequence, our algorithm always choose base vectors which attend constraints (C1) and (C2) (see last section). Therefore, since the vector field inside the triangles is generated by a convex combination of the incident base vectors (equation (1)), it is guaranteed the robot will stay inside the corridor and will move to the next triangle. We still have to prove this is also true when a continuity problem appears.

Since rotating vectors never point outwards the corridor, it is clear that constraint (C1) is attended. 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 these rotating vectors appear.

Suppose 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 where 3 triangles are used to subdivide the original problematic triangle (see Fig. 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 Property 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 attended. In triangle T_2, imagine there is a vector $\mathbf{V}_\theta$ perpendicular to the rotating vector pointing in direction of the rotation (see Fig. 4(b)). If we prove $\mathbf{u}(\mathbf{q})\cdot\mathbf{V}_\theta > 0$ then it is proved 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 Property 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 will use the same argument in T_3. Since $\mathbf{v}_{\mathbf{q}_k} = \gamma(\mathbf{q}_j - \mathbf{q}_n)$, by using Property 1, $\mathbf{u}(\mathbf{q})\cdot\mathbf{V}_\theta > 0$. It is straight forward to check that this argument is also true inside the next triangles incidents to $\mathbf{q}_j$.

Suppose now the case where 2 triangles are used to subdivide the original problematic triangle (see Fig. 4(a)). Obviously, except for T_2 the analysis is the same as the one developed in the last 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 Property 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. ■

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, in equation (1) 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 III), and $l = i, j, k$, and perform some simple algebra, then we can show that

$$\mathbf{u}(\mathbf{q}) = \beta(\mathbf{q}_g - \mathbf{q}). \tag{2}$$

Therefore, $\mathbf{u}(\mathbf{q})\cdot(\mathbf{q}_g - \mathbf{q}) = \beta\|\mathbf{q}_g - \mathbf{q}\|^2 \geq 0$. Notice that $\mathbf{u}(\mathbf{q})$ is equal zero only at $\mathbf{q}_g$, which means the vector field only vanishes at the goal. ■

In the next section we present some simulations to illustrate the effectiveness of our approach.

V. Simulations

In this section we illustrate the proposed methodology for fully continuous vector field computation in two different sequences of triangles. The first sequence fits in the case explained in Section III where we propose to subdivide the problematic triangle into two new triangles. On the other hand, the second sequence fits in case where we propose to subdivide it into three new triangles.

Figure 5(a) presents the referred first sequence where the problematic triangle is marked with dashed horizontal

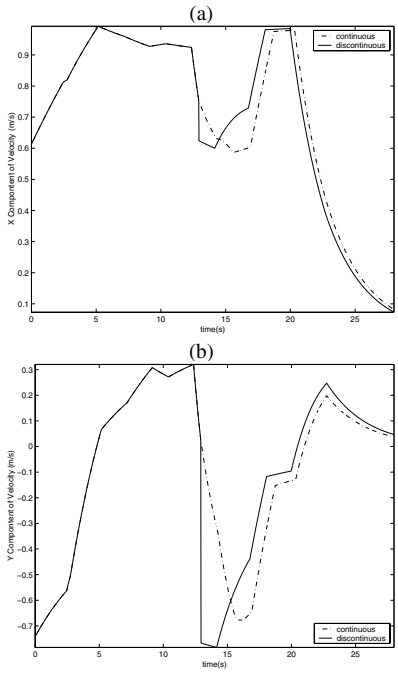

Fig. 6. Velocity components for the trajectories shown in Figure 5.

lines. This sequence is quite similar to the one presented in [9] since we aim to compare the resultant vector fields. Actually, we compare the proposed fully continuous vector field with a piecewise vector field generated by the algorithm we presented in [11] which is an adaptation of the algorithm in [9]. Both algorithms suffer from the same drawback related to the discontinuity caused by the split of the original sequence of triangles into two new sequences.

Figure 5(b) presents the vector field computed by the the old algorithm (presented in [11]) and also a typical holonomic trajectory. The vector field and the analogous trajectory obtained by the novel approach is in Figure 5(c). This figure also shows the two new triangles created by the proposed subdivision of the problematic triangle. By looking at these two figures, one can clearly visualize the improvement in the smoothness with the new algorithm. This improvement can also be verified in Figures 6(a) and (b) which present the components of the velocity vectors along the trajectories presented before. The discontinuity occurs about the instant 13s where we notice the separation between the corresponding curves.

The second sequence of triangles is in Figure 7(a). The corresponding vector field computed by the proposed methodology and a typical holonomic trajectory are in Figure 7(b). In Figure 7(b), it is also presented the three new triangles created by the algorithm. As expected, by simple inspection it is obvious that the vector field is continuous. In the next section we present conclusions of this work and also give some possible future directions.

VI. Conclusions and Future Directions

This paper proposed a simple and efficient methodology to create fully continuous, two dimensional vector fields on sequences of discrete triangles. We present proofs showing, that, besides continuous, the computed vector field keeps the robot inside the sequence and drives it to the goal. The methodology is complementary to the one proposed in [6] and may be used to model other kinds of hybrid systems.

Next steps of this work include the extension to three dimensions, where the configuration space is divided into a set of tetrahedra. This will allow the hybrid control of rotating and translating ground robots. Also, we are planning to extend our ideas for controlling outdoor mobile robots in [11] to include continuous vector fields. It is also important to look at the discrete algorithms responsible to find the optimal sequence of cells. For controlling non-holonomic robots, besides designing a non-linear controller to follow the vector field, it is also necessary to guarantee the robot can make all the turns imposed by the field. We believe this guarantees may be obtained by acting on the geometry of the sequence and not on the vector field methodology itself.

Acknowledgment

The authors would like to thanks the financial support of FAPEMIG and CNPq – Brazil. The authors also thanks Alexandre Ramos Fonseca for his help with the figures.

References

[1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L. Kavraki, and S. Thrun, *Principles of Robot Motion: Theory, Algorithms, and Implementations*. The MIT Press, 2005.

[2] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.

[3] C. I. Connnolly, J. B. Burns, and R. Weiss, "Path planning using Laplace's equation," in *Proc. IEEE Int. Conf. Robot. Automat.*, 1990, pp. 2102–2106.

[4] E. Rimon and D. E. Koditschek, "Exact robot navigation using artificial potencial functions," *IEEE Trans. Robot. and Automat.*, vol. 8, no. 5, pp. 501–517, 1992.

[5] L. C. A. Pimenta, A. R. Fonseca, G. A. S. Pereira, R. C. Mesquita, E. J. Silva, W. M. Caminhas, and M. F. M. Campos, "On computing complex navigation functions," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2005, pp. 3463–3468.

[6] C. Belta and L. Habets, "Constructing decidable hybrid systems with velocity bounds," in *Proc. of the IEEE Conference on Decision and Control*, 2004, pp. 467–472.

[7] M. Kloetzer and C. Belta, "A framework for automatic deployment of robots in 2d and 3d environments," in *Proc. IEEE/RSJ Int. Conf. Intel. Robots and Systems*, 2006.

[8] G. E. Fainekos, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for mobile robots," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2005, pp. 2032–2037.

[9] C. Belta, V. Isler, and G. J. Pappas, "Discrete abstractions for robot motion planning and control in polygonal environments," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 864–874, October 2005.

[10] S. R. Lindemann and S. M. LaValle, "Smoothly blending vector fields for global robot navigation," in *Proc. of the IEEE Conference on Decision and Control*, 2005, pp. 3553–3559.

[11] G. A. S. Pereira, L. C. A. Pimenta, L. Chaimowicz, A. R. Fonseca, D. S. C. de Almeida, L. de Q. Corrêa, and R. C. M. M. F. M. Campos, "Robot navigation in multi-terrain outdoor environments," in *Proceedings of the International Simposium on Experimental Robotics*, 2006.

[12] D. C. Conner, A. A. Rizzi, and H. Choset, "Composition of local potential functions for global robot control and navigation," in *Proc. IEEE/RSJ Int. Conf. Intel. Robots and Systems*, 2003, pp. 3546–3551.

[13] J. R. Shewchuk, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator," in *Applied Computational Geometry: Towards Geometric Engineering*, ser. Lecture Notes in Computer Science, M. C. Lin and D. Manocha, Eds. Springer-Verlag, May 1996, vol. 1148, pp. 203–222.

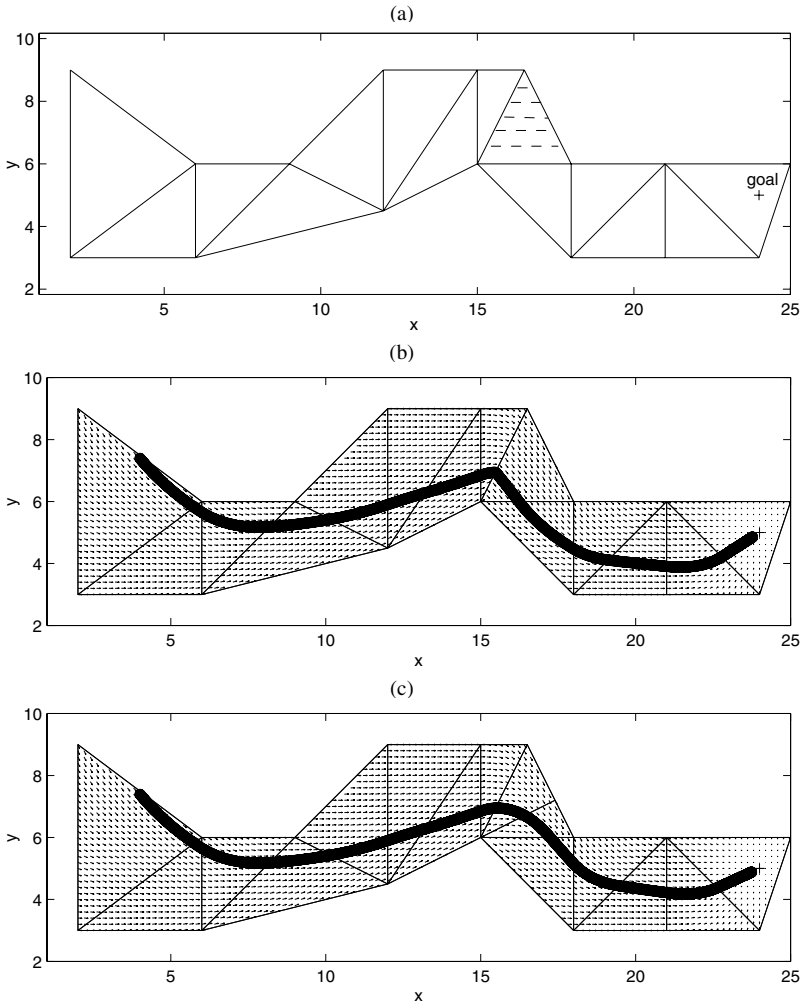

Fig. 5. (a) Sequence of triangles with a problematic triangle marked with horizontal dashed lines. (b) Trajectory overlayed on the resultant vector field computed by the methodology that do not treats the discontinuity problem. (c) Trajectory overlayed on the fully continuous vector field computed by the methodology proposed in this paper.

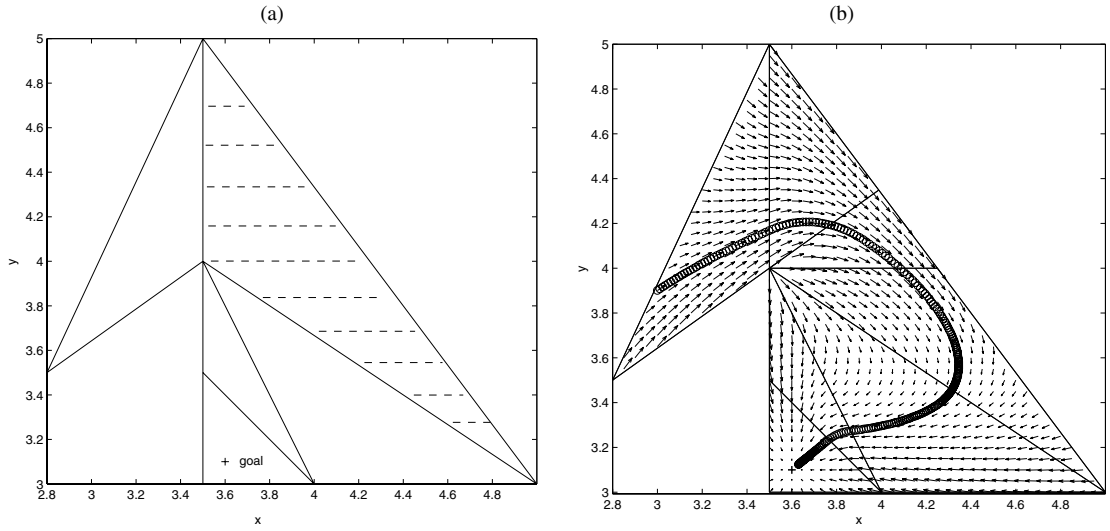

Fig. 7. (a) Sequence of triangles with a problematic triangle marked with horizontal dashed lines. (b) Trajectory overlayed on the fully continuous vector field computed by the methodology proposed in this paper.