International Journal of Computational Geometry
& Applications
Vol. 20, No. 5 (2010) 511–525
© World Scientific Publishing Company
DOI: 10.1142/S0218195910003414

A COVERING PROJECTION FOR ROBOT NAVIGATION UNDER STRONG ANISOTROPY*

ANDREA ANGHINOLFI

Servizio Sistemi Informativi e Telematica
Provincia di Modena
Modena, Italy
anghinolfi.a@provincia.modena.it

LUCA COSTA

Autodesk SpA
Milano, Italy
luca.costa@autodesk.com

MASSIMO FERRI

Dept. of Mathematics and ARCES
University of Bologna
Bologna, Italy
ferri@dm.unibo.it

ENRICO VIARANI

ARCES
University of Bologna
Bologna, Italy
eviarani@deis.unibo.it

Received 24 June 2005
Revised 22 November 2006
Communicated by Bernard Chazalle

ABSTRACT

Path planning can be subject to different types of optimization. Some years ago a German researcher, U. Leuthäusser, proposed a new variational method for reducing most types of optimization criteria to one and the same: minimization of path length. This can be done by altering the Riemannian metric of the domain, so that optimal paths (with respect to whatever criterion) are simply seen as shortest. This method offers an extra feature, which has not been exploited so far: it admits direction–dependent criteria.

*Research accomplished under support of INdAM-GNSAGA, of MACROGeo and of the University of Bologna, funds for selected research topics.

In this paper we make this feature explicit, and apply it to two different anisotropic settings. One is that of different costs for different directions: E.g. the situation of a countryside scene with ploughed fields. The second is dependence on oriented directions, which is called here "strong" anisotropy: the typical scene is that of a hill side. A covering projection solves the additional difficulty. We also provide some experimental results on synthetic data.

Keywords: Path–planning; Riemannian metric.

1. Introduction

Path planning can be conditioned by different factors, as fuel consumption, mission duration and others.[5,14,17,22] U. Leuthäusser[11–13] has shown how to discharge these factors on a Riemannian metric on the plane or space where the task has to be performed. The optimal path then becomes the path $x(t)$ of minimal "length", where this length is classically computed by the line integral (see, e.g., Formula 4.19 of Ref. 10)

$$\int_a^b \left(\sum_{i,k=1}^{n} g_{ik}(x_j, \dot{x}_j)\dot{x}_i\dot{x}_k \right)^{1/2} dt$$

the g_{ik} being the components of the metric tensor. Of course, these components loose their original geometric meaning (Formula 3.2 of Ref. 10). See Sec. 2.1 for a sketch of their computation.

Actually, Leuthäusser's articles only treat isotropic cases, where the metric tensor is expressed by the identity matrix multiplied by a (possibly varying) scalar factor. Those papers concentrate on the very important problems related to the variational calculus involved.

Here, on the contrary, we shall be less concerned with the computational problem. We have modelled the domain as a labelled directed graph, and we have used quite standard methods for finding shortest paths between vertices. Moreover, we shall assume no dependence on velocity: The g_{ik} will depend solely on position. The ambient space will be 2-dimensional.

Of course, we do not underestimate the importance of computational issues. Still, we prefer to focus on the main, so far unexploited feature of the method, and to defer performance optimization to a possible, further collaboration with an engineering environment. In fact, the main goal of this paper is to explore the most peculiar potentiality of Leuthäusser's method not covered by that Author's papers: path planning in anisotropic environments. We shall consider two separate types of anisotropy, which we shall call weak and strong.

Weak anisotropy is a situation in which *cost* (i.e. ground resistance, fuel consumption, lack of comfort, threat, or whatever is to be minimized) depends on the direction, but not on the orientation along a direction. We mean that a displacement vector and its opposite have equal cost. This is typically the case of ploughed fields, which was anticipated in Ref. 2, and which will be our example of navigation under weak anisotropy.

Strong anisotropy is the situation in which cost is different for two opposite displacement vectors. This is the case with slopes, currents, wind. This will give us an additional problem, since this condition makes it impossible to stick with a mere metric tensor, if we insist not to have an explicit dependence on velocity. We shall overcome the problem by using a particular map which is a cross–section of a 2-fold covering projection of the tangent plane.[19]

Ground characteristics, as for resistance to the motion of a fixed vehicle, are usually expressed by maximum speeds, common to whole *ground units* with homogeneous characteristics. This is the convention used, e.g., in the N.A.T.O. Army Mobility Model AMM–75.[15] On the contrary, the eigenvalues of our matrices will be directly proportional to the difficulty of movement. In the strongly anisotropic case, we shall even have negative values.

The practical meaning of negative values is tied with situations in which movement along a direction, with one of the two possible orientations, yields a gain: E.g., a vehicle descending a slope on a fairly smooth ground might move passively and charge its batteries. A similar scenario applies to a boat sailing before the wind. The example which we shall use in our simulation, is a slope with rough ground, crossed by a zig–zag road.

Several Authors have treated robot motion on rough terrain: Ref. 18 is mainly concerned with the stability of a computed path at varying approximations of the terrain description;[6] also deals with path stability and considers the use of landmarks; some, as,[7] treat extensively the physical problems related with wheel–terrain contact. On the other hand, also dependence on direction has been studied, but mainly involving the robot's mechanical structure and the connected constraints.[4,9] But perhaps the most careful analysis of path planning in anisotropic environments is the one of Ref. 21 followed by the related one of Ref. 24: Our method could be seen as a sort of mixing of the ideas of these Authors with the one of Leuthäusser.

Section 2 of the present article describes the method from the mathematical viewpoint. Section 3 is devoted to the problems connected with discretization. Section 4 contains the descriptions and comments on some simulations. Section 5 comments on the usefulness of the method while the conclusions are drawn in Sec. 6. Sections 2–4 have separate subsections corresponding to weak and strong anisotropy.

2. Norm and Skew-norm

2.1. *Weak anisotropy*

Under weak anisotropy (or under isotropy), at each point of the ground the metric is represented by the metric tensor, in the form of a real, symmetric square matrix of order two (having fixed an orthonormal basis for each tangent space in a coherent way). The corresponding bilinear form is obtained by matrix products, at both sides of the matrix, of the component pairs of the vectors; this induces a quadratic form (here, v is a generic vector of components (a, b)):

$$v \mapsto (a, b) \mapsto \begin{pmatrix} a & b \end{pmatrix} \cdot A \cdot \begin{pmatrix} a \\ b \end{pmatrix} = \|v\|^2. \tag{1}$$

Note that the quadratic form has to be always positive definite, since every displacement has a cost. So $\| \ \|$ is really a norm on the tangent space, and the bilinear form is a scalar (or inner) product. It is this norm, which enters the computation of the line integral yielding the (fictitious) length of a path.

The directions of least and greatest cost will be mutually orthogonal, due to the symmetry of the matrix; they are given by the eigenspaces of A. The eigenvalues represent the extremal costs. The isotropic case is represented by a matrix A with just one eigenvalue λ of multiplicity two, so $A = \lambda I_2$. Of course, in practical situations, things go the other way around: The extremal costs (eigenvalues) and their directions (eigenspaces) are directly measured on the terrain, and matrix A is obtained by conjugation (change of basis).

In practice, the construction of A at each point proceeds as follows. First, at points where resistence to motion — or any other cost — is independent of direction, A is simply the identity matrix I_2 multiplied by the scalar representing cost, e.g, the inverse of maximum speed. Obstacles are represented by sets of points, where this scalar is set at a very high value. At a point where cost is (weakly) anisotropic, we compute the matrix A by first building an auxiliary diagonal matrix D, whose diagonal entries are the maximal and minimal costs, as directly measured on the terrain. If E is matrix of the change of orthonormal basis which takes the E–W direction to the one of maximal cost, and the N–S direction to the one of minimal cost, then $A = E^{-1} \cdot D \cdot E$. Note that, in order to keep within classic Riemannian metrics, A has to be symmetric, and the extremal directions have to be orthogonal; the more general situation of nonorthogonal extremal directions — hence of an asymmetric matrix — is affordable, but will not be considered in this paper.

2.2. *Strong anisotropy*

One important feature of the above mentioned approach is the fact that reversing a vector does not affect its norm. This is why the method, without modifications, cannot take into account an *oriented* direction, but only directions as parallelism classes. So, the following method has been conceived expressly to allow a distinction between a displacement vector and its opposite.

The method simply consists in taking a vector v of Cartesian components (a, b), passing to polar coordinates (ρ, θ), halving the anomaly, passing again to Cartesian components (a', b'), then evaluating the resulting pair by a suitable quadratic form:

$$\begin{aligned} v \mapsto (a, b) \mapsto (\rho, \theta) \mapsto (\rho, \theta/2) \mapsto (a', b') \mapsto \qquad (2) \\ \mapsto \begin{pmatrix} a' & b' \end{pmatrix} \cdot A \cdot \begin{pmatrix} a' \\ b' \end{pmatrix} = f(v). \end{aligned}$$

Of course, the composite function f so defined is not the square of a true norm, and not even a quadratic form so this is no more a Riemaniann metric. Moreover, the quadratic form corresponding to the matrix A need not be positive definite. In fact, in the case of strong anisotropy some displacements might incur null cost, or even yield a gain. Therefore, A may also have nonpositive eigenvalues.

We define the *skew-norm* of v as

$$\|v\| = \mathtt{sign}(f(v))\sqrt{|f(v)|}. \tag{3}$$

The core of the composition is the function which maps (ρ, θ) to $(\rho, \theta/2)$. This is a cross-section of the 2-fold covering projection $(\rho, \eta) \mapsto (\rho, 2\eta)$ of the tangent plane onto itself, branched on the null vector (see p. 292 of Ref. 19). This cross-section transforms opposite vectors into orthogonal ones, so allowing to apply the previous method.

The vectors at greatest and least skew-norm are opposite to each other, due to the symmetry of A, hence to the orthogonality of its eigenspaces. The isotropic case is again represented by means of a matrix with a single eigenvalue λ of multiplicity two: $A = \lambda I_2$.

In the strong anisotropic case, concrete construction of A then proceeds at each point as follows. A diagonal matrix D is built, whose diagonal entries are the maximal and minimal cost. Now, let r be an oriented straight line whose positive direction goes from minimal to maximal cost. Let then s be the oriented line which bisects the angle between the positive half–lines on the x-axis and on r. If E is the matrix of the change of orthonormal basis taking the oriented direction W → E (i.e. of positive x) to the oriented direction of s, then $A = E^{-1} \cdot D \cdot E$.

Note that our representation of (weak and strong) anisotropy is still rather rigid, although effective. The rigidity comes from the fact that the only "degrees of freedom" are a direction (the other is compelled to be orthogonal, in the weak case), the value along that direction, the value along the orthogonal direction.

In the strong case, just one direction is directly considered, with the two possible orientations. By the way, our method allows to treat ploughed fields and hill sides, but not the two together. To solve the problem of ploughed fields on a hill side, we should be compelled to mix the values coming from the two methods vector by vector.

Of course, the method proposed in this paper is just one of many possible solutions to the problem of distinguishing orientation. We have chosen it as the slightest possible variation from the weak anisotropy model and perhaps the simplest one allowing orientation.

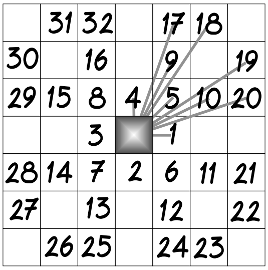
3. The Discrete Model

We have chosen to treat the variational problem on a directed graph. Vertices represent equally spaced ground points of a rectangular grid, arcs connect each vertex to the ones of a suitable neighbourhood. So each arc starting at a vertex P corresponds to a tangent vector based at P; we label it by the norm of the vector, computed according to the metric tensor at P itself. This rough discretization is introduced just for explaining the method by simple simulations; of course, a real application should directly deal with tangent vectors.

We can assume that the ground patch of interest fits into a rectangle, and that we draw the rectangular grid of vertices with rows and columns parallel to its sides. Recall that orthogonality and length are here the physical ones. For the topographic map of an undulating ground, the distances are actually those of the projection on a horizontal plane. Units can be so adjusted, that the nearest neighbors of each vertex are at distance one.

For the sake of description simplicity, we can assume that one side of the rectangle, and the x axis at each vertex, are in East–West direction, and the other, together with all y axes, in North–South direction. We progressively number the vertices, from left to right, from top to bottom.

The vehicle is supposed to be endowed with a limited number of possible displacements: With the obvious exceptions of the vertices at the border of the rectangle, each vertex P is joined to 32 vertices (as, e.g., in Ref. 20). These correspond to the 32 vectors of integer components (m, n) with $1 \leq |m| + |n| \leq 5$ which are not multiples of vectors with lesser $|m+n|$ (see Fig. 1). Of course, a finer or coarser directional resolution can be chosen by the actual programmer.



	31	32		17	18	
30		16		9		19
29	15	8	4	5	10	20
		3		1		
28	14	7	2	6	11	21
27		13		12		22
	26	25		24	23	

Fig. 1. The 32 vertices reached by elementary displacements.

In other words, the vertices adjacent to P are the ones which can be reached from P by a "tower–like" path of no more than five steps, in which at least one change of direction is necessary; the vertices left out are those which can be reached by repeated, shorter displacements in the same direction.

This setting has a problem: the robot tends to cut through the edges of obstacles (i.e. ground units with high eigenvalues). This happens because at P the robot may test a displacement vector to a point Q and find it convenient, even if the line PQ crosses an obstacle (what it cannot know). A possible solution to this problem (which we have tested in Ref. 2) is smoothing the boundaries between all pairs of different ground units: Each ground unit is surrounded by a belt of vertices at which the metric is a mean of the nearby metrics. We shall not use this expedient in the experiments of the present paper, as it might slightly confuse the evaluation of the main method; here, our "robots" do cut through the edges.

As an algorithm for finding paths with minimal weight sum, we (as also Ref. 18) have chosen the one of Dijkstra,[3] which has an $O(m + n \log n)$ computational complexity on a graph with m arcs and n vertices. The algorithm implemented in our program runs — for small experiments as the ones presented here — in real time. For a detailed survey on shortest path algorithms see Ref. 16.

3.1. *Weak anisotropy*

As already said, at each vertex a matrix (g_{ij}) represents the metric tensor with respect to the fixed base. Then the squared norm of a displacement vector is evaluated as in Formula 1 of Sec. 2.1. E.g., the vector v of components $(1, 2)$ is multiplied as follows: $\begin{pmatrix} 1 & 2 \end{pmatrix} \cdot \begin{pmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \end{pmatrix}$ whence the norm $\sqrt{g_{11} + 2g_{12} + 2g_{21} + 4g_{22}}$.

3.2. *Strong anisotropy*

The only difference from the case of weak anisotropy, is that the weights are not directly given by matrix multiplication. Before this operation, conversion to polar, halving of the anomaly (the cross section of the described covering projection), and conversion back to Cartesian are performed, according to Formula 2 and Formula 3 of Sec. 2.2.

The same vector v of components $(1, 2)$ will now be processed as follows.

$$\begin{aligned} v \mapsto (1,2) &\mapsto \left(\sqrt{5}, \arctan(2)\right) \mapsto \left(\sqrt{5}, \frac{\arctan(2)}{2}\right) \qquad (4) \\ &\mapsto \sqrt{5}\left(\sqrt{\frac{\sqrt{5}+1}{2\sqrt{5}}}, \sqrt{\frac{\sqrt{5}-1}{2\sqrt{5}}}\right) \\ &\mapsto \frac{5+\sqrt{5}}{2} g_{11} + \sqrt{5} g_{12} + \sqrt{5} g_{21} + \frac{5-\sqrt{5}}{2} g_{22} \end{aligned}$$

which is the value of f on the displacement vector, and from which, depending on the sign, we get the skew-norm by Formula 3.

For practical purposes, we assume that the system either already has a database formed by the matrices A for each relevant point, or builds it dynamically while exploring the terrain. At each point, in either case, the following algorithm computes the costs of the elementary displacements v[i] (32 in the case of Fig. 1) by means of the entries g11, g12, g22 of the symmetric A at that point, so getting as output the cost values for each v[i]; these become the input data for the optimal (or suboptimal) path search.

```
for (i = 0; i < 32; i++)
{ a = v[0] [i]; b = v[1] [i];  /*loading 32 displacements*/
  ro = sqrt ((a^2) + (b^2)); /*from Cartesian to polar and...*/
  theta = atan (b / a);
  aprime = ro * cos (theta /2); /*...from "half"polar back to Cartesian*/
  bprime = ro * sin (theta / 2);
  f[i] = (aprime^2) * g11 +     /*local cost computation*/
                2 * aprime * bprime * g12 +
                (bprime^2) * g22;
}
```

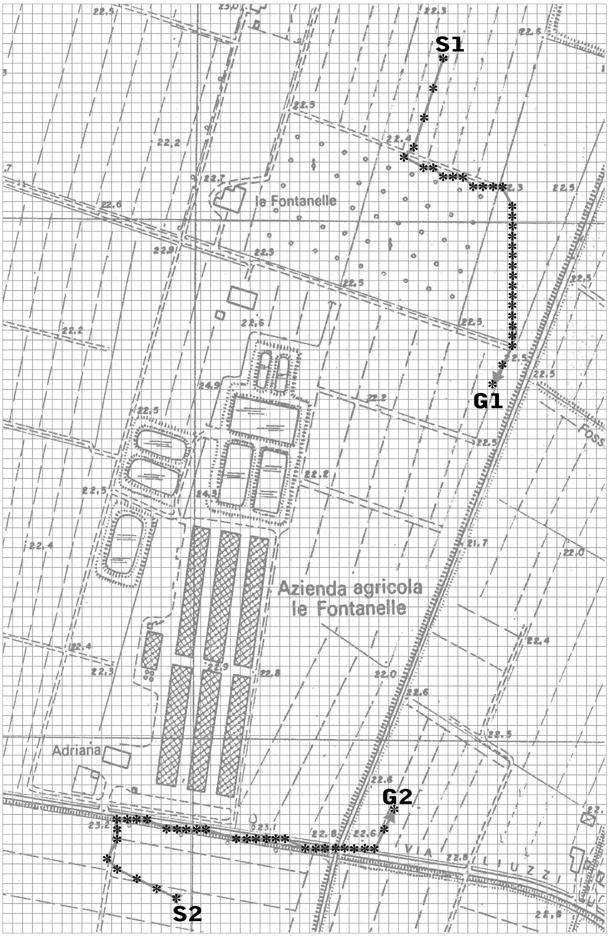


Fig. 2. Two simulations in weak anisotropy.

4. Simulations

4.1. *Weak anisotropy*

Besides some isotropic situations, we have experimented with the program on a true countryside topographic map (Fig. 2). The ground patch considered has different features, the most intriguing one being the presence of ploughed fields, with furrows of several directions. $S1, S2$ are starting points, and $G1, G2$ are, respectively, goals for the moving robot.

In the upper part of the domain, near $S1$ and $G1$ there are furrows forming an angle of about 18° with the N–S direction. The matrix E of change of basis is then approximately $\begin{pmatrix} 0.95 & -0.3 \\ 0.3 & 0.95 \end{pmatrix}$.

If we assume to have measured (with respect to suitable units) a resistance to motion of 10 across the furrows and of 5 along, then the diagonal matrix D is $\begin{pmatrix} 10 & 0 \\ 0 & 5 \end{pmatrix}$ and the final matrix A is approximately

$$\begin{pmatrix} 9.5 & -1.4 \\ -1.4 & 5.4 \end{pmatrix} = \begin{pmatrix} 0.95 & 0.3 \\ -0.3 & 0.95 \end{pmatrix} \cdot \begin{pmatrix} 10 & 0 \\ 0 & 5 \end{pmatrix} \cdot \begin{pmatrix} 0.95 & -0.3 \\ 0.3 & 0.95 \end{pmatrix}$$

The orchard (represented as a grid of dots) is an isotropic obstacle, and we set its resistence to motion at a high 50: $\begin{pmatrix} 50 & 0 \\ 0 & 50 \end{pmatrix}$ whereas the meadow just north of $G1$ is isotropic but practicable; assume we have measured its resistance to motion to be 5: $\begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix}$. There is a narrow track at the upper margin of the orchard; we assume its metric tensor to be $\begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$.

The result of the search for an optimal path (computed as "shortest" by the system according to the displacement costs deriving from these matrices) is shown by the dotted lines: from $S1$ the robot first moves along the furrows, then it turns eastwards when it hits the boundary of the orchard and goes along the track up to the corner. There, it finds the meadow which it crosses diagonally, pointing not exactly towards the goal, but almost, in order to reach the endpoint of a furrow, along which it goes up to $G1$.

More interesting yet is the bottom path, starting at point $S2$. The robot points away from the goal, because of the presence of a drain (isotropic with value 100); it chooses to reach a track going along the furrows; then it reaches the junction with a road (isotropic with value 1), goes right over the bridge, up to the furrow which leads it to $G2$.

Note that the furrow starting at $S2$ is not followed exactly: this is because the angle with the x axis is not one of the 32 allowed. However, the solution looks acceptable.

4.2. *Strong anisotropy*

For a simulation under strong anisotropy, we have worked with totally synthetic data (Figs. 3–6). Letter S stands for Start and G for Goal. The figures are bird's eyeviews. The sign in the lower right corner of each figure indicates (in perspective) the direction of maximum slope, with the corresponding cost values.

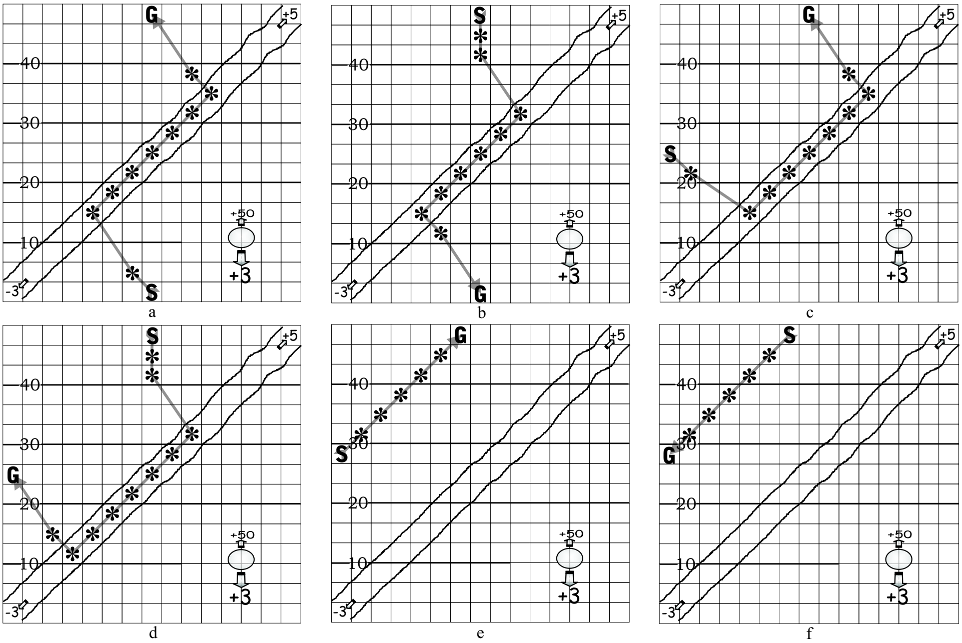


Fig. 3. A road crossing a hillside.

The situation of Figs. 3 and 4 represents a steep side of a hill (with maximum slope from South to North), on which a road climbs from South–West to North–East. In Fig. 3 the road has been given eigenvalues 5 uphill and -3 downhill (along its own direction), so the corresponding diagonal matrix is $\begin{pmatrix} 5 & 0 \\ 0 & -3 \end{pmatrix}$.

Recalling that the angles with the x axis have to be halved before computing the matrix, the matrix E of change of basis is the one of a $\pi/8$ rad rotation:

$$\begin{pmatrix} \frac{\sqrt{\sqrt{2}+2}}{2} & \frac{\sqrt{2-\sqrt{2}}}{2} \\ -\frac{\sqrt{2-\sqrt{2}}}{2} & \frac{\sqrt{\sqrt{2}+2}}{2} \end{pmatrix}$$

and the resulting matrix A is

$$\begin{pmatrix} 2\sqrt{2}+1 & 2\sqrt{2} \\ 2\sqrt{2} & 1-2\sqrt{2} \end{pmatrix} = \begin{pmatrix} \frac{\sqrt{\sqrt{2}+2}}{2} & -\frac{\sqrt{2-\sqrt{2}}}{2} \\ \frac{\sqrt{2-\sqrt{2}}}{2} & \frac{\sqrt{\sqrt{2}+2}}{2} \end{pmatrix} \cdot \begin{pmatrix} 5 & 0 \\ 0 & -3 \end{pmatrix} \cdot \begin{pmatrix} \frac{\sqrt{\sqrt{2}+2}}{2} & \frac{\sqrt{2-\sqrt{2}}}{2} \\ -\frac{\sqrt{2-\sqrt{2}}}{2} & \frac{\sqrt{\sqrt{2}+2}}{2} \end{pmatrix}$$

The meadows have eigenvalues 50 uphill (i.e. towards the top of the picture) and 3 downhill, with matrix

$$\begin{pmatrix} 53/2 & 47/2 \\ 47/2 & 53/2 \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix} \cdot \begin{pmatrix} 50 & 0 \\ 0 & 3 \end{pmatrix} \cdot \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix}$$

The setting of Figure 4 differs in that the meadows are given eigenvalues 50 and -1, i.e. with a slight gain in descent, so that the matrix is

$$\begin{pmatrix} 49/2 & 51/2 \\ 51/2 & 49/2 \end{pmatrix} = \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix} \cdot \begin{pmatrix} 50 & 0 \\ 0 & -1 \end{pmatrix} \cdot \begin{pmatrix} \sqrt{2}/2 & -\sqrt{2}/2 \\ \sqrt{2}/2 & \sqrt{2}/2 \end{pmatrix}$$

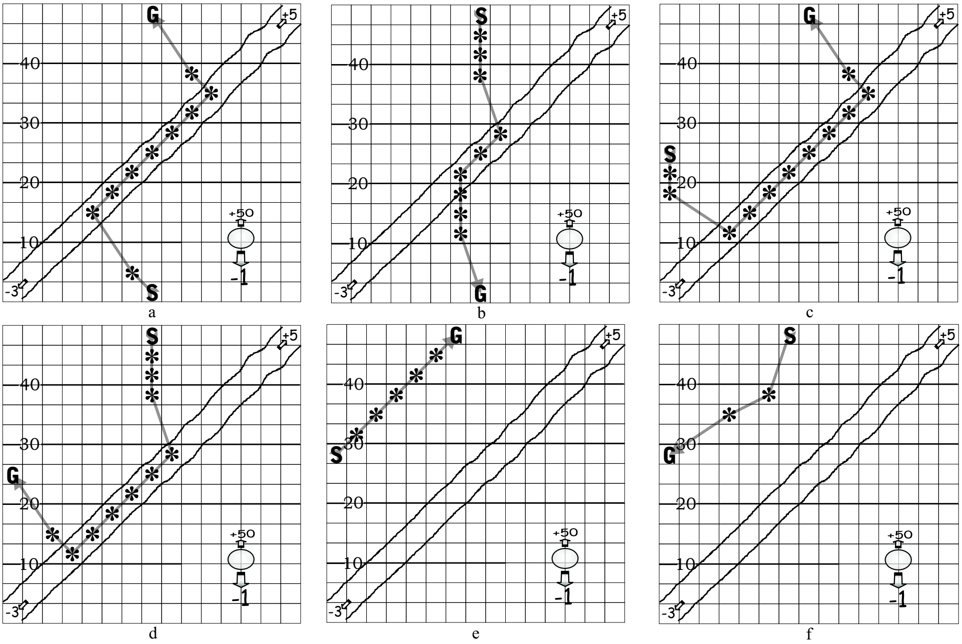
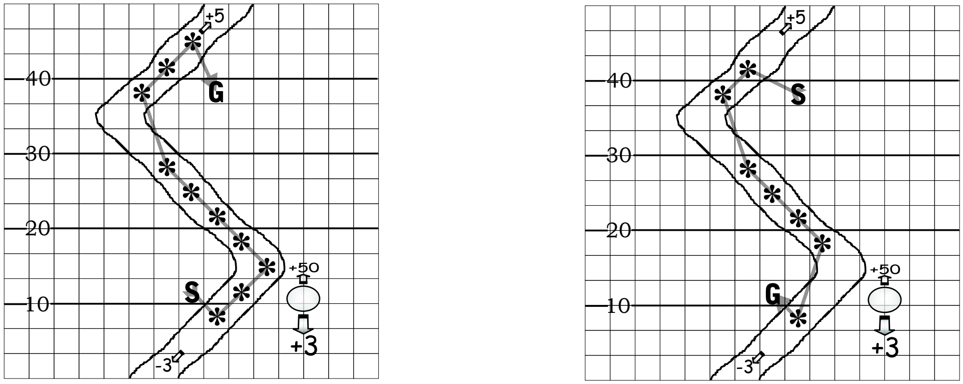


Fig. 4. A road crossing a hillside, with a gain when going downhill.

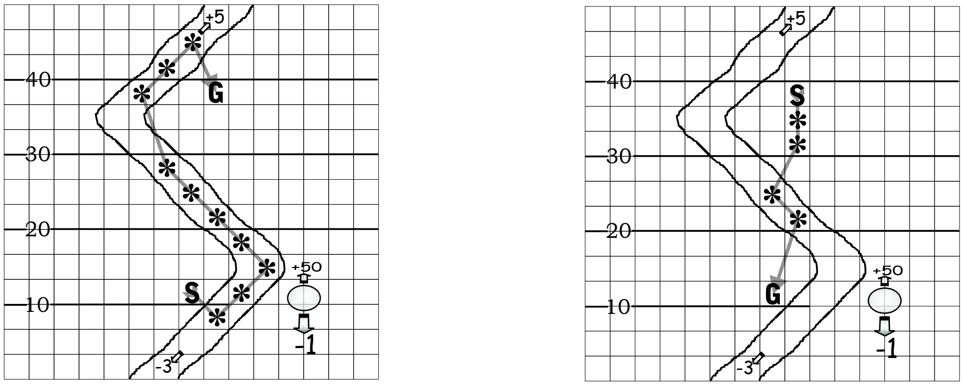
As the pairs of starts and goals are the same in both situations, it is possible and interesting to compare the robot behaviours. Of course there is much more symmetry between ascent and descent in Fig. 3; in Figs. 3(a) to 3(d) the robot points straight to the road and goes along it — be it uphill or downhill. Only in Figs. 3(e) and 3(f) are start and goal too close, and the robot does not make use of the road. Figure 4(b) is very different from the homologous Fig. 3(b): the robot covers just a short path on the road, as it finds it convenient to cross the meadows downhill. There is a difference also between Figs. 3(f) and 4(f): whereas Fig. 3(f) is symmetrical to Fig. 3(e), this is not so for Fig. 4(f). In fact the robot tries to get as much as possible from the gain that comes from going straight downhill.

A more complex setting is shown in Figs. 5 and 6: Here the road goes zig–zag. The eigenvalues for the road are again 5 and –3. Figure 5 is relative to eigenvalues 50 and 3 for the meadows, whereas Fig. 6 has 50 and –1 instead. Also in these cases one can appreciate the asymmetry between ascent [Figs. 5(a) and 6(a)] and descent [Figs. 5(b) and 6(b)], and between the two different solutions in descent [Figs. 5(b) and 6(b)], while ascent is the same.



a b

Fig. 5. A zig-zag road crossing a hillside.



a b

Fig. 6. A zig-zag road crossing a hillside, with a gain when going downhill.

Of course, the discretization used here yields awkward polygonals. It would be interesting to implement a more faithful simulation in a continuous domain.

5. Why This Method

A reasonable objection to our method has been: "Why not directly assign weights to edges?". The answer is in the number of decisions or measurements to be made; our method just requires to evaluate the field characteristic in the form of a symmetric 2×2 matrix, instead of the usual single parameter. Then the method takes care of deciding the weights of the vectors corresponding to the various directions, whatever resolution (i.e. whatever the number of directions) the user wants to adopt. So the user has just to find the direction of maximum-minimum resistance to motion, and to measure the values of resistance in the two opposite orientations.

Speculations on theoretical issues are not new to the area of motion planning: Ref. 1 is concerned with different metrics; Ref. 25 abstracts potential field theory (as many others) in order to re–apply it to motion planning with the support of fuzzy logic; Ref. 23 is closer to our work, in that it introduces a fuzzy rule–based Traversability Index. As we mentioned earlier, direction–dependent path planning under the effect of friction and gravity has been treated in an extremely detailed way by Rowe and Ross in Ref. 21 and more recently by Sun and Reif in Ref. 24. These papers, moreover, treat exhaustively the problem of impermissibility.

A comparison between our method and the results we have just quoted is not possible, since the aims are different. Those papers study properties of optimal paths, or deal with computational issues, within a specific model of anisotropy. Instead, we are proposing a different way of modelling anisotropy: a new model to which those same results might apply. In fact, the only possible, true comparison is between our construction and Leuthäusser's own applications of his idea. There, there is no doubt that ours is a definite progress.

We think that our contribution is potentially meaningful, as it offers the possibility of mixing the advantages of the methods of Rowe, Ross, Sun, Reif to the great generality of Leuthäusser's idea; moreover it offers a rather new mathematical viewpoint on the problem.

As mathematicians, we are particularly interested in unfolding the present method — introduced in an industrial environment — in its full theoretical capability. Leuthäusser's method of altered Riemaniann metric seems to be little present in the literature, maybe because of its limited exploitation by that Author. Its great simplicity and generality suggest that it could — if not substitute — at least integrate alternative path planning methods in anisotropic environments. Its major drawback, i.e. insensitivity to direction but not to orientation, is overcome in the present paper.

As *applied* mathematicians, we care for the effectiveness of a theoretical issue. Our simulations show that this tool produces plausible reactions to various simple strongly anisotropic conditions. It could be interesting to collaborate with an engineering team in exploiting integrations and applications of this tool.

6. Conclusions

We have shown how to use a varying metric tensor to direct the choice of a path for a moving robot, in anisotropic situations.

In the case of weak anisotropy (indifference to reversal of displacement) the method is simply the one suggested by U. Leuthäusser and plainly imported from Riemannian geometry. Under strong anisotropy, where opposite displacements have different costs, the use of a covering projection enhances the method.

Some simulations show the applicability of both techniques.

References

1. N. M. Amato, O. B. Bayazit, L. K. Dale, C. Jones and D. Vallejo, Choosing good distance metrics and local planners for probabilistic roadmap methods, *IEEE Trans. Robot. Autom.* **16**(4) (2000) 442–447.
2. A. Anghinolfi and M. Ferri, Navigazione robotica in ambiente anisotropo, 37^{TH} *ANIPLA Annual Conference Milano* (Nov. 23–25, 1993) pp. 695–703.
3. M. L. Fredman and R. E. Tarjan, Fibonacci heaps and their uses in improved network optimization algorithms, *JACM* **34**(3) (1987) 596–615.
4. T. Fukao, H. Nakagawa and N. Adachi, Adaptive tracking control of a nonholonomic mobile robot, *IEEE Trans. Robot. Autom.* **16**(5) (2000) 609–615.
5. G. Gallo and S. Pallottino, Short path algorithms, *Ann. Operat. Res.* **13** (1988) 3–79.
6. A. Haït, T. Siméon and M. Taïx, Algorithms for rough terrain trajectory planning, *Adv. Robot.* **16**(8) (2002) 673–699.
7. K. Iagnemma, H. Shibly, A. Rzepniewski and S. Dubowsky, Planning and control algorithms for enhanced rough–terrain rover mobility, *Proc. Sixth Int. Symp. Artificial Intelligence, Robotics and Automation in Space, i-SAIRAS* (2001).
8. D. B. Johnson, A note on Dijkstra's shortest path algorithm, *JACM* **20**(3) (1973) 385–388.
9. F. Lamiraux and J.-P. Lammond, Smooth motion planning for car-like vehicles, *IEEE Trans. Robot. Autom.* **17**(4) (2001) 498–501.
10. D. Laugwitz, Differentialgeometrie, Teubner, Stuttgart, 1977.
11. U. Leuthäusser, Robot path planning based on variational methods, *Mobile Robots IV, Proc. SPIE* (Boston 14-15 Nov. 1991), Vol. 1613, eds. W. J. Wolfe and W. H. Chun, pp. 146–154.
12. U. Leuthäusser, On navigation functions for shortest path problems, *IASTED Proceedings "Robotics and Manufacturing"* (Oxford, 1993).
13. U. Leuthäusser, Path planning for autonomous robots in digital maps, *EURNAV 92 Proc. "Digital Mapping and Navigation"* (London, 1992).
14. R. Malik and S. Prasad, Robot mapping in unstructured environments, *Mobile Robots IV, Proc. SPIE* (Boston 14-15 Nov. 1991), Vol. 1613, eds. W. J. Wolfe and W. H. Chun, pp. 181–189.
15. K. J. Melzer, Analytical methods and modelling state of the art report, *J. Terramechanics* (Pergamon Press, N.Y., 1982).
16. J. S. B. Mitchell, Geometric shortest paths and network optimization, *Handbook of Computational Geometry*, eds. J.-R. Sack and J. Urrutia (Elsevier Science Publ. B.V. North-Holland, Amsterdam, 2000), pp. 633–701.
17. G. Oriolo, The reactive vortex fields method for robot motion planning with uncertainty, *Automation 1992 Proc.* 36^{TH} *ANIPLA Ann. Conf.* (Genoa Nov. 16-18 1992), Vol. 2, eds. G. Bottaro and R. Zoppoli, pp. 584–597.

18. K. Pai and L.-M. Russell, Multiresolution rough terrain motion planning, *IEEE Trans. Robot. Autom.* **14**(1) (1998) 19–33.
19. D. Rolfsen, *Knots and Links* (Publish or Perish, 1976).
20. N. C. Rowe and Y. Kanayama, “Minimum-energy paths on a vertical-axis cone with anisotropic friction and gravity effects, *Int. J. Robot. Res.* **13**(5) (1994) 408–432.
21. N. C. Rowe and R. S. Ross, Optimal grid-free path planning across arbitrarily contoured terrain with anisotropic friction and gravity effects, *IEEE Trans. Robot. Autom.* **6**(5) (1990) 540–553.
22. K. Sabe, M. Fukuchi, J.-S. Gutmann, T. Ohashi, K. Kawamoto and T. Yoshigahara, Obstacle avoidance and path planning for humanoid robots using stereo vision, *Int. Conf. Robotics and Automation (ICRA)* (New Orleans, 2004), pp. 592–597.
23. H. Seraji and A. Howard, Behavior–based robot navigation on challenging terrain: A fuzzy logic approach, *IEEE Trans. Robot. Autom.* **18**(3) (2002) 308–321.
24. Zh. Sun and J. H. Reif, On finding approximate optimal paths in weighted regions, *J. Algorithms* **58** (2006) 1–32.
25. N. Tsourveloudis, L. Doitsidis and K. Valavanis, Autonomous navigation of unmanned vehicles: A fuzzy logic perspective, *Cutting Edge Robotics* (2005), pp. 291–310.