Fast Computation of Weighted Distance Functions and Geodesics on Implicit Hyper-Surfaces

Facundo Mémoli
Instituto de Ingeniería Eléctrica
Universidad de la República
Montevideo, Uruguay
memoli@iie.edu.uy

Guillermo Sapiro*
Electrical and Computer Engineering
University of Minnesota
Minneapolis, MN 55455
guille@ece.umn.edu

March 2001

Abstract

An algorithm for the computationally optimal construction of intrinsic weighted distance functions on implicit hyper-surfaces is introduced in this paper. The basic idea is to approximate the intrinsic weighted distance by the Euclidean weighted distance computed in a band surrounding the implicit hyper-surface in the embedding space, thereby performing all the computations in a Cartesian grid with classical and computationally optimal numerics. Based on work on geodesics on Riemannian manifolds with boundaries, we bound the error between the two distance functions. We show that this error is of the same order as the theoretical numerical error in computationally optimal, Hamilton-Jacobi based, algorithms for computing distance functions in Cartesian grids. Therefore, we can use these algorithms, modified to deal with spaces with boundaries, and obtain also for the case of intrinsic distance functions on implicit hyper-surfaces a computationally optimal technique. The approach can be extended to solve a more general class of Hamilton-Jacobi equations defined on the implicit surface, following the same idea of approximating their solutions by the solutions in the embedding Euclidean space. The framework here introduced thereby allows to perform the computations on a Cartesian grid with computationally optimal algorithms, in spite of the fact that the distance and Hamilton-Jacobi equations are intrinsic to the implicit hyper-surface. For other surface representations like triangulated or unorganized points ones, the algorithm here introduced can be used after simple pre-processing of the data.

*Corresponding author

Contents

1 Introduction

Computing distance functions has a number of applications in numerous areas including mathematical physics, image processing, computer vision, robotics, computer graphics, computational geometry, optimal control, and brain research. In addition, having the distance function from a seed to a target point, it is straightforward to compute the corresponding geodesic path, since this is given by the gradient direction of the distance function, back propagating from the target toward the seed (see for example [17]). Geodesics are used for example for path planning in robotics [34], brain flattening and brain warping in computational neuroscience [55, 56, 58, 59, 66], crests, valleys, and silhouettes computations in computer graphics and brain research [7, 29, 60], mesh generation [62], and many applications in mathematical physics. Distance functions are also very important in optimal control [57] and computational geometry for computations such a Voronoi diagrams and skeletons [47]. It is then of extreme importance to develop efficient techniques for the accurate and fast computations of distance functions. It is the goal of this paper to present an accurate and computationally optimal technique for the computation of intrinsic weighted distance functions on implicit hyper-surfaces.[1] It is well-know already, and it will be further detailed below, that this weighted distances can be obtained as the solution of simple Hamilton-Jacobi equations. We will also show that the framework here presented can be applied to a larger class of Hamilton-Jacobi equations defined on implicit surfaces. We also discuss the application of our proposed framework to other, non-implicit, surface representations.

1.1 Distance Function Computation and its Hamilton-Jacobi Formulation

Before proceeding, let us first formally define the concept of *intrinsic* weighted distances on *implicit* hyper-surfaces. Let $\mathcal{S}$ be a (codimension 1) hyper-surface in $\mathbb{R}^d$ defined as the zero level set of a function $\psi : \mathbb{R}^d \to \mathbb{R}$. That is, $\mathcal{S}$ is given by $\{x \in \mathbb{R}^d : \psi(x) = 0\}$. We assume from now on that ψ is a signed distance function to the surface $\mathcal{S}$. (This is a not a limitation, since as we will discuss later, both explicit and implicit representations can be transformed into this form.) Our goal is, for a given point $p \in \mathcal{S}$, to compute the intrinsic g-weighted distance function $d^g_{\mathcal{S}}(p, x)$ for all desired points $x \in \mathcal{S}$.[2] Note that we are referring to the intrinsic g-distance, that is, the geodesic distance on the Riemannian manifold $(\mathcal{S}, g^2 I\!I)$ ($I\!I$ stands for the $d \times d$ identity matrix) and not on the embedding Euclidean space. For a given weight g defined on the surface (we are considering only isotropic metrics for now), the g-distance on $\mathcal{S}$ (that coincides with the geodesic distance of the Riemannian manifold $(\mathcal{S}, g^2 I\!I)$) is given by

$$d^g_{\mathcal{S}}(p, x) \stackrel{\triangle}{=} \inf_{\mathcal{C}_{px}[\mathcal{S}]} \{\mathbf{L}_g(\mathcal{C})\} \tag{1}$$

where

$$\mathbf{L}_g\{\mathcal{C}\} \stackrel{\triangle}{=} \int_a^b g\left(\mathcal{C}(l)\right) \, \|\dot{\mathcal{C}}(l)\| \, dl \tag{2}$$

[1] Although all the examples in this paper are going to be reported for 3D surfaces, the theory is valid for general dimension hyper-surfaces, and it will be presented in this generality. A number of applications deal with higher dimensions. For example, for the general theory of harmonic maps, in order to deal with maps onto general open surfaces, it is necessary to have this notion of intrinsic distance [40]. In addition, higher dimensions might appear in motion planning, when explicitly assuming that the robot is not modeled by a point, thereby adding additional constraints to its movements.

[2] This can certainly be extended to any subset of $\mathcal{S}$.

is the weighted *length functional* defined for piecewise C^1 curves $\mathcal{C} : [a, b] \to \mathcal{S}$, and $C_{px}[\mathcal{S}]$ denotes the set of curves piecewise C^1 joining p to x, traveling on $\mathcal{S}$. In general we will consider the definition to be valid for any $\tilde{g}$ defined over the domain that the curve may travel through .

We need to compute this distance when all the concerning objects are represented in discrete form in the computer. Computing minimal weighted distances and paths in graph representations is an old problem that has been optimally solved by Dijkstra [20]. Dijkstra showed an algorithm for computing the path in $O(n \log n)$ operations, where n is the number of nodes in the graph. The weights are given on the edges connecting between the graph nodes, and the algorithm is computationally optimal. In theory, we could use this algorithm to compute the weighted distance and corresponding path on polygonal (not implicit) surfaces, being the vertices the graph nodes and the edges the connections between them (see [32]). The problem is that this algorithm is limited to travel on the graph edges, giving only a first approximation of the true distance. Moreover, Dijkstra's is not a consistent algorithm: it will not converge to the true desired distance when the graph and grid is refined [41, 42]. The solution to this problem, limited to Cartesian grids, was developed in [26, 50, 51, 57] (and recently extended by Osher and Helmsen, see [44]). Tsitsiklis first described an optimal-control type of approach, while independently Sethian and Helmsen both developed techniques based on upwind numerical schemes. The solution presented by these authors is consistent and converges to the true distance [48, 57], while keeping the same optimal complexity of $O(n \log n)$. This work was later extended in [31] for triangulated surfaces (see also [7, 35] for related works on numerics on non-Cartesian grids). We should note that the algorithm developed in [31] is currently developed only for triangulated surfaces with acute triangles. Therefore, before the algorithm can be applied, as an initialization step the surfaces have to be pre-processed to remove all obtuse triangles or other polygons present in the representation [30]. Following [51], we call to these algorithms *fast marching*.

The basic idea behind the computationally optimal techniques for finding accurate weighted distances, fast marching algorithms, is to note that the distance function holds a Hamilton-Jacobi Partial Differential Equation (PDE) in the viscosity sense; see for example [37, 49] for the general topic of distance functions on Riemannian manifolds (and a nice mathematical treatment), and [12, 22, 30, 43, 45, 51] for the planar (and more intuitive) case. This Hamilton-Jacobi is given by

$$\|\nabla_{\mathcal{S}} d^g_{\mathcal{S}}\| = g \tag{3}$$

where $\nabla_{\mathcal{S}}$ is the gradient intrinsic to the surface, and $d^g_{\mathcal{S}}$ is the g-distance from a given seed point to the rest of the manifold.[3]

That is, we can transform the problem of optimal distance computation into the problem of solving a Hamilton-Jacobi equation (recall that g is known, it is the given weight), also known as the Eikonal equation. In order to solve this equation, the current state of knowledge permits to accurately and optimally (in a computational sense) find (weighted) distances on Cartesian grids as well as on particular triangulated surfaces (after some pre-processing, namely the elimination of obtuse triangles, see [6, 31]). The goal of this paper is to extend this to implicit hyper-surfaces. In other words, we will show how to solve the above Eikonal equation for implicit hyper-surfaces $\mathcal{S}$.

Recall that although all the applications in this paper will be presented for 3D surfaces, the theory is valid for any d-dimensional hyper-surfaces, and will then be presented in this generality.

[3]Note that $\nabla_{\mathcal{S}}$ and $d^g_{\mathcal{S}}$ become the classical gradient and distance respectively for Euclidean spaces.

1.2 Distance Function and Geodesics on Implicit Surfaces

The motivations behind extending the distance map calculation to implicit surfaces are numerous: **a)** in many applications, surfaces are already given in implicit form, e.g., [10, 13, 15, 24, 44, 45, 52, 64, 60], and there is then a need to extend to this important representation the fast techniques previously mentioned. We could of course triangulate the implicit surface, eliminate obtuse triangles, and then use the important algorithm proposed in [31]. This is not a desirable process in general when the original data is in implicit form, since it affects the distance computation accuracy due to errors from the triangulation, and also adds the computational cost of the triangulation itself, triangulation that might not be needed by the specific application. If for example all what it is needed is to compute the distance between a series of points on the surface, the computational cost added by the triangulation is unnecessary. Note that finding a triangulated representation of the implicit surface is of course dimensionality dependent, and adds the errors of the triangulation process. Moreover, accurate triangulations that ensure correctness in the topology are computationally expensive, and once again there is no reason to perform a full triangulation when we might be interested just in the intrinsic distance between a few points on the implicit surface. **b)** it is a general agreement that the work on implicit representations and Cartesian grids is more robust when dealing with differential characteristics of the surface and partial differential equations on it. Numerical analysis on Cartesian grids is much more studied and supported by fundamental results than the work on polygonal surfaces. It is even recognized that there is no consensus about how to compute basic differential quantities over a triangulated surface, see for example [21], although there is quite an agreement for implicit surfaces. Moreover, representing an hyper-surface with structured elements such as triangles is certainly difficult for dimensions other than 2 or 3. **c)** if the computation of the distance function is just a part of a general algorithm for solving a given problem, it is not elegant, accurate, nor computationally efficient to go back and forth from different representations of the surface.

Before proceeding, we should note that although the whole framework and theory is here developed for implicit surfaces, it is valid for other surface representations as well after simple pre-processing. This will be discussed later in the paper (§5). Moreover, we will lately assume that the embedding is a distance function. This is not a limitation, since many algorithms exist to transform a generic embedding function into a distance one; see also §5. Therefore, the framework here presented can be applied both to implicit (naturally) and other surface representations like triangulated ones.

In order to compute intrinsic distances on surfaces, a small but important number of techniques have been reported in the literature. As mentioned before, in a very interesting work Kimmel and Sethian extended the fast marching algorithm to work on triangulated surfaces. In its current version, this approach can only be used when dealing with 3D triangulated surfaces and its extension to deal with higher dimensions seems very involved. Moreover, it can only correctly handle acute triangulations (thereby requiring a pre-processing step). And of course, it doesn't apply to implicit surfaces without some pre-processing (a triangulation).

Another very interesting approach to computing intrinsic distances, this time working with implicit surfaces, was introduced in [15]. This will be further described below, but before that let's make some comments on it. First, this is an evolutionary/iterative approach, whose steady state gives the solution to the corresponding Hamilton-Jacobi. Therefore, this approach it is not computationally optimal for the class of Hamilton-Jacobi equations discussed in this paper.[4] Second, very careful discretization must be done to the equation proposed in [15] do to the presence

[4]The general framework introduced in [15] is applicable beyond the Hamilton-Jacobi equations discussed in this

of intrinsic jump functions that might change the zero level-set (i.e., the surface). On the other hand, the numerical implementation is not necessary done via the utilization of *monotone schemes*, as required by our approach and all the fast marching techniques previously mentioned (thereby having a theoretical error $\Theta(\sqrt{\Delta x})$ [18]), and better accuracy might then be obtained.

In order to compute the intrinsic distance on an implicit surface, we must then solve the corresponding Hamilton-Jacobi equation presented before. In order to do this in a computationally efficient way, we need to extend the fast marching ideas in [26, 44, 50, 51, 57], which assume a Cartesian grid, to work in our case. Since an implicit surface is represented in a Cartesian grid, corresponding to the embedding function, the first and most intuitive idea is then to attempt to solve the *intrinsic Eikonal* using the *fast marching* technique. The first step towards our goal is to express all the quantities in the intrinsic Eikonal by its *implicit-extended* representations. What we mean is that the *intrinsic* problem (we consider $g = 1$ for simplicity of exposition)

$$\begin{cases} \|\nabla_{\mathcal{S}}\, d_{\mathcal{S}}(p)\| = 1 \ \text{ for } p \in \mathcal{S} \\ d_{\mathcal{S}}(p) = 0. \end{cases} \tag{4}$$

with $p \in \mathcal{S}$ the seed point, is to be extended to all $\mathbb{R}^d$ (or at least to a band surrounding $\mathcal{S}$), and the derivatives are to be taken tangentially to $\{\psi = 0\}$. Considering then the projection of the Euclidean gradient onto the tangent space of $\mathcal{S}$ to obtain the intrinsic one, and denoting by $\hat{d}$ the Euclidean extension to the intrinsic distance $d_{\mathcal{S}}$, we have to numerically solve, in the embedding Cartesian grid, the equation

$$\begin{cases} \|\nabla \hat{d}(x)\|^2 - |\nabla \hat{d}(x) \cdot \nabla \psi(x)|^2 = 1 \ \text{ for } x \in \mathbb{R}^d \\ \hat{d}(l(p)) = 0. \end{cases} \tag{5}$$

where $l(p)$ is the ray through p normal to the level sets of ψ.

This is exactly the approach introduced in [15], as discussed above, to build-up the evolutionary approach, given by the following PDE:

$$\phi_t + sgn(\phi_0)\left(\sqrt{\|\nabla\phi\|^2 - |\nabla\phi \cdot \nabla\psi|^2} - 1\right) = 0 \tag{6}$$

where $\phi_0(x) = \phi(x, 0)$ is the initial value of the evolving function, generally a step-like function that tells inside from outside of the zero level-set. One then finds $\hat{d}(\cdot) = \phi(\cdot, \infty)$.

Of course, in order to obtain a computationally optimal approach, we want to solve the stationary problem (5), and not its iterative counterpart (6). It turns out that the basic requirements for the construction of a fast marching method, even with the recent extensions in [44], do not hold for this equation. This can be explicitly shown, and has also been hinted by Kimmel and Sethian in their work on geodesics on surfaces given as graphs of functions.[5]

To recap, the fast marching approach cannot be directly applied to the computation of intrinsic distances on implicit surfaces defined on a Cartesian grid (equation (5)), and the state of the art in numerical analysis for this problem says that in order to compute intrinsic distances one has either to work with triangulated surfaces or has to use the iterative approach mentioned above. The problems with both techniques were reported before, and it is the goal of this paper to present a third approach that addresses all these problems.

paper (see also [8, 16]). Here we limit the comparison between the techniques to the equations were both approaches are applicable.

[5] We have also benefited from private conversations with Stan Osher and Ron Kimmel to confirm this claim.

1.3 Our Contribution

The basic idea here presented is conceptually very simple. We first consider a small h offset of $\mathcal{S}$. That is, since the embedding function ψ is a distance function, with $\mathcal{S}$ as its zero level set, we consider all points x in $\mathbb{R}^3$ for which $|\psi(x)| \leq h$. This gives a region in $\mathbb{R}^d$ with boundaries. We then modify the (Cartesian) fast marching algorithm mentioned above for computing the distance transform inside this h-band surrounding $\mathcal{S}$. Note that here, all the computations are as in the works in [26, 50, 51, 57], in a Cartesian grid. We then use this Euclidean distance function as an approximation of the intrinsic distance on $\mathcal{S}$. In §2 we show that the error between these two distances, under reasonable assumptions on the surface $\mathcal{S}$, is of the same order as the numerical error introduced by the fast marching algorithms in [26, 50, 51, 57].[6] Therefore, when adapting these algorithms to work on Euclidean spaces with boundary, adaptation described in §3, we obtain an algorithm for the computation of intrinsic distances on implicit surfaces with the same simplicity, computational complexity, and accuracy than the optimal fast marching techniques for computing Euclidean distances on Cartesian grids.[7] In §3 we also explicitly discuss the numerical error of our proposed technique. Examples of the algorithm here proposed are given in §4. Since Osher and Helmsen have recently shown that the fast marching algorithm can be used to solve additional Hamilton-Jacobi equations, we show that the framework here proposed can be applied to equations from that class as well, this is done in §5. This section also discusses the use of the framework here presented for non-implicit surfaces. Finally, some concluding remarks are given in §6.

2 Distance Functions: Intrinsic vs. Extrinsic

The goal of this section is to present the connection between the intrinsic distance function and the Euclidean one computed inside a band surrounding the (implicit) surface. We will completely characterize the difference between these two functions, mainly based on results on shortest paths on manifolds with boundary. The results here presented will justify the use of the Cartesian fast marching algorithms also for the computation of intrinsic weighted distances on implicit surfaces.

Recall that we are dealing with a closed hyper-surface $\mathcal{S}$ in $\mathbb{R}^d$ represented as the zero level-set of a distance function $\psi : \mathbb{R}^d \to \mathbb{R}$. That is, $\mathcal{S} = \{\psi = 0\}$. Our goal is to compute a g-weighted distance map on this surface from a seed point $q \in \mathcal{S}$. Let

$$\Omega_h \stackrel{\triangle}{=} \bigcup_{x \in \mathcal{S}} B(x,h) \;=\; \{x \in \mathbb{R}^d : |\psi(x)| \leq h\}$$

be the *h-offset* of $\mathcal{S}$ (here $B(x,h)$ is the ball centered at x with radius h). It is well known that for a smooth $\mathcal{S}$, $\partial\Omega_h$ is also smooth if h is sufficiently small, see Appendix A for references. Ω_h is then a *manifold with smooth boundary.*

Our computational approach is based on approximating the solution of the *intrinsic* problem ($d^g_{\mathcal{S}}(p)$ is the intrinsic g-weighted distance on $\mathcal{S}$).

[6] In contrast with works such as [1, 46], where an offset of this form is just used to improve the complexity of the level-sets method, in our case the offset is needed to obtain a small error between the computed distance transform and the real intrinsic distance function, see next Section.

[7] Although in this paper we deal with the fast marching techniques, other techniques for computing distance functions on Cartesian grids, e.g., the fast technique reported in [11] for uniform weights, could be used as well, since the basis of our approach is the approximation of the intrinsic distance by an extrinsic one.

$$\begin{cases} \|\nabla_{\mathcal{S}} d^g_{\mathcal{S}}(p)\| = g \ \text{ for } p \in \mathcal{S} \\ d^g_{\mathcal{S}}(q) = 0. \end{cases} \tag{7}$$

by that of the *Euclidean* (or *extrinsic*) one:

$$\begin{cases} \|\nabla d^{\tilde{g}}_{\Omega_h}(p)\| = \tilde{g} \ \text{ for } p \in \Omega_h \\ d^{\tilde{g}}_{\Omega_h}(q) = 0. \end{cases} \tag{8}$$

where $\tilde{g}$ is a smooth *extension* of g in a domain containing Ω_h, and $d^{\tilde{g}}_{\Omega_h}(p)$ is the Euclidean $\tilde{g}$-weighted distance in Ω_h. Our goal is to be able to control $\|d^g_{\mathcal{S}} - d^{\tilde{g}}_{\Omega_h}\|_{L_\infty(\mathcal{S})}$ with h. Note that we have replaced the intrinsic gradient $\nabla_{\mathcal{S}}$ by the Euclidean one and the intrinsic distance $d^g_{\mathcal{S}}(p)$ on the surface by the Euclidean distance $d^{\tilde{g}}_{\Omega_h}(p)$ in Ω_h. We have then transformed the problem of computing an intrinsic distance into the problem of computing a distance in an Euclidean manifold with boundary.

We will show that under suitable (and likely) geometric conditions on $\mathcal{S}$ we can indeed control $\|d^g_{\mathcal{S}} - d^{\tilde{g}}_{\Omega_h}\|_{L_\infty(\mathcal{S})}$ with h. In order to materialize this, we first need to briefly discuss the extension $\tilde{g}$ and to review some basic background material on Riemannian manifolds with boundary.

2.1 The Extension of the Weight g

We require that $\tilde{g}|_{\mathcal{S}} = g$, and that $\tilde{g}$ is *smooth* within Ω_h. There are situations when one has a readily available extension, other where the extension has to be "invented." We call the former *natural extension* and the latter *general extension.* Both cases, as argued below, will provide smooth functions $\tilde{g}$.

In many applications the weight $g : \mathcal{S} \to I\!R$ depends on the curvature structure of the hyper-surface. Denoting $\mathbf{B}_{\mathcal{S}}(\cdot) : \mathcal{S} \to I\!R^{d\times d}$ the second fundamental form of $\mathcal{S}$, and $\Lambda\left(\mathbf{B}_{\mathcal{S}}(x)\right)$ the set of its eigenvalues, this means that

$$g(x) = F(\Lambda\left(\mathbf{B}_{\mathcal{S}}(x)\right)$$

where F is a given function. In this case it is utterly *natural* to take advantage of the implicit representation by noting that $\mathbf{B}_{\mathcal{S}}(x) = \mathbf{H}_{\psi|_{T_x\mathcal{S}}}(x)$ for $x \in \mathcal{S}$, where $\mathbf{H}_\psi$ is the Hessian of ψ and $T_x\mathcal{S}$ is the tangent space to $\mathcal{S}$ at x (see [36]). The *natural* extension then becomes

$$\tilde{g}(x) = F(\Lambda(\mathbf{H}_{\psi|_{T_x\mathcal{S}}}(x))), \ \ x \in \Omega_h \tag{9}$$

This extension is valid for $\{x \in I\!R^d : |\psi(x)| < 1/\mathcal{M}_{\mathcal{S}}\}$, where $\mathcal{M}_{\mathcal{S}}$ absolutely bounds all principal curvatures of $\mathcal{S}$, see Appendix A.

When the weight g cannot be directly extended to be valid for a tubular neighborhood of the hyper-surface, one has to do that in a pedestrian way. One such extension comes from propagating the values of g along the normals of $\mathcal{S}$ in a constant fashion, i.e.:

$$\tilde{g}(x) = g(\Pi_{\mathcal{S}}(x)), \ \ x \in \Omega_h \tag{10}$$

where $\Pi_{\mathcal{S}}(\cdot) : I\!R^d \to \mathcal{S}$ stands for the normal projection onto $\mathcal{S}$. This extension is well defined and smooth as long as there is a unique *foot* in $\mathcal{S}$ for every x in the domain of the extension Ω. Taking h sufficiently small we can guarantee that $\Omega \supset \Omega_h$ if $\mathcal{S}$ is smooth. See Appendix A for some details.

In practice this extension can be accomplished solving the equation [14]

$$\phi_t + sgn(\psi)\ \nabla\psi \cdot \nabla\phi = 0$$

with initial conditions given by any $\phi(\cdot, 0)$ such that $\phi(\cdot, 0)|_{\mathcal{S}} = g$. Then $\tilde{g}(\cdot) \stackrel{\triangle}{=} \phi(\cdot, \infty)$.

2.2 Shortest Paths and Distance Functions in Manifolds with Boundary

Since we want to approximate the problem of intrinsic distance functions by a problem of distance functions in manifolds with boundary, and to prove that the latter converges to the former, we need to review basic concepts on this subject. We will mainly include results from [2, 3, 61]. We are interested in the existence and smoothness of the geodesic curves on manifolds with boundary, since our convergence arguments below depend on these properties. We will assume throughout this section that $(\mathcal{M}, m)$ is a *connected* and *complete* Riemannian manifold with boundary (this will later become the h-offset Ω_h with the metric $\tilde{g}^2 I\!I$).

Deffinition 1 *Let $p, q \in \mathcal{M}$, then if $d_{\mathcal{M}}(\cdot,\cdot) : \mathcal{M} \times \mathcal{M} \to I\!R$ is the* distance function *in $\mathcal{M}$ (with its metric m), a* shortest path *between p and q is a path joining them such that its Riemannian length equals $d_{\mathcal{M}}(p, q)$.*

Now, since $\mathcal{M}$ is *complete*, for every pair of points p and q there exist a *shortest path* joining them, see [2]. The following results deals with the regularity of this shortest path.

Theorem 1 *Let $(\mathcal{M}, m)$ be a C^3 manifold with C^1 boundary B. Then any* shortest path *of $\mathcal{M}$ is C^1.*

When $(\mathcal{M}, m)$ is a flat manifold (i.e. $\mathcal{M}$ is a codimension 0 subset of $I\!R^d$ and the metric m is isotropic and constant), it is easy to see that any *shortest path* must be a straight line whenever it is in the interior of $\mathcal{M}$, and a shortest path of the boundary B when it is there. That will be the situation for us from now on.

It might seem a bit awkward that one cannot achieve higher regularity class than C^1 for the shortest paths, even by increasing the regularity of $\mathcal{M} \cup B$, but a simple counterexample will convince the reader. Think of $\mathcal{M}$ as $I\!R^2$ with the open unit disc removed, see Figure 1, and its Euclidean metric. The acceleration in all the open segment (AP) is $\vec{0}$, and in all the open arc (PQ) is $-\vec{e}_r$, that is, it points inwards, and has modulus 1. That is, even in most simple examples, C^2 regularity is not achievable. It is, however, very easy to check that in this case $\dot{\gamma}$ is actually *Lipschitz*.

For the general situation, in [3, 38] the authors proved that shortest paths do have *Lipschitz continuous* first derivative, what means that in fact shortest paths are twice differentiable *almost everywhere* by Rademacher's Theorem. This fact will be of great importance below.

For a more comprehensive understanding of the theory of shortest paths and distance functions in Riemannian manifolds with boundary, see [2, 3, 38, 61] and references therein.

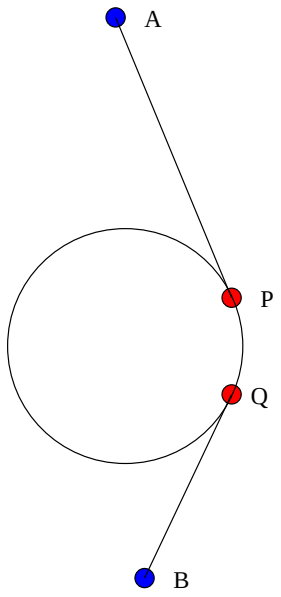

Figure 1: The minimal path is C^1, but not C^2.

2.3 Convergence Result for the Extrinsic Distance Function

We now show the relation between the Euclidean distance in the band Ω_h and the intrinsic distance in the surface $\mathcal{S}$. Below we will denote $d_{\mathcal{S}} \stackrel{\triangle}{=} d^1_{\mathcal{S}}$, and $d_{\Omega_h} \stackrel{\triangle}{=} d^1_{\Omega_h}$.

Observation 1 *Since we assume the implicit surface $\mathcal{S}$ to be compact, the continuous function $d_{\mathcal{S}} : \mathcal{S} \times \mathcal{S} \to I\!R$ attains its maximum. Therefore we can define the diameter of the set as*

$$diam(\mathcal{S}) \stackrel{\triangle}{=} \sup_{p,q \in \mathcal{S}} d_{\mathcal{S}}(p,q) < \infty$$

Observation 2 *Since $\mathcal{S} \subset \Omega_h$ we have that for every pair of points p and q in $\mathcal{S}$, $d_{\Omega_h}(p,q) \leq d_{\mathcal{S}}(p,q)$, so in view of the previous observation we have*

$$d_{\Omega_h}(p,q) \leq diam(\mathcal{S}) \;\; \forall\, p,q \in \mathcal{S}$$

Observation 3 *Since we are assuming $\tilde{g}$ to be a* smooth *extension of g to all $\Omega \supset \Omega_h$ (we stress the fact that the extension does not depend on h), $\tilde{g}$ will be Lipschitz in Ω, and we call $K_{\tilde{g}}$ its associated constant. Further, we will denote $M_g \stackrel{\triangle}{=} \max_{\{x \in \mathcal{S}\}} g(x)$ and $M_{\tilde{g}} \stackrel{\triangle}{=} \sup_{\{x \in \Omega\}} \tilde{g}(x)$.*

We need the following *Lemma* whose simple proof we omit (see for example [17]).

Lemma 1 *When a $\tilde{g}$-shortest path travels through an interior region, its curvature is absolutely bounded by*

$$B_{\tilde{g}} \stackrel{\triangle}{=} \sup_{\{x \in \Omega\}} \left(\frac{\|\nabla \tilde{g}(x)\|}{\tilde{g}(x)} \right)$$

The following Lemma will be needed in the proof of the Theorem below. Its proof can be found in Appendix B.

Lemma 2 *Let $f : [a,b] \to I\!R$ be a $C^1([a,b])$ function such that f' is Lipschitz. Let $\varphi \in L^\infty([a,b])$ denote (one of) f''s weak derivative(s). Then*

$$\int_a^b f'^2(x)\,dx = f\,f'|_a^b - \int_a^b f(x)\varphi(x)\,dx$$

We are now ready to present one of the main results of this Section. We bound the error between the intrinsic distance on $\mathcal{S}$ and the Euclidean one in the offset Ω_h. As we will see below, in the most general case, the error is of the order $h^{1/2}$ (h being half the offset width). We will later discuss that this is also the order of the theoretical error for the numerical approximation in fast marching methods. That will lead us to conclude that our algorithm does keep the convergence rate within the theoretically proven order for fast marching methods numerical approximation. However, for all practical purposes, the order of convergence in the numerical schemes used by fast marching methods is that of h, see [48]. We will also argue that for all practical purposes we can guarantee no decay in the overall rate of convergence. We defer the detailed discussion on this to after the presentation of the general bound below.

Theorem 2 *Let A and B be two points on the* smooth hypersurface *$\mathcal{S}$. Let $d_h^{\tilde{g}} = d_{\Omega_h}^{\tilde{g}}(A,B)$ and $d_{\mathcal{S}}^{g} = d_{\mathcal{S}}^{g}(A,B)$. Then we have that for sufficiently small h*

$$\left| d_{\mathcal{S}}^{g} - d_h^{\tilde{g}} \right| \leq h^{\frac{1}{2}}\; C(h)\; diam(S)$$

where $C(h)$ depends on the global curvature structure of $\mathcal{S}$ and on $\tilde{g}$, and approaches a constant when $h \downarrow 0$ (it does not depend on A nor B, we give a precise form of $C(h)$ in the proof).

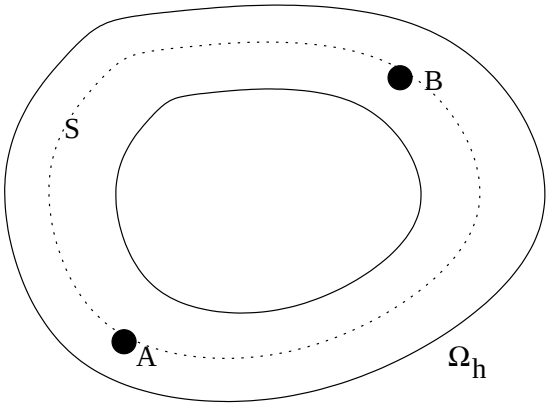

Figure 2: Tubular neighborhood.

Proof:
Let $d_h = d_{\Omega_h}(A,B)$; $d_{\mathcal{S}} = d_{\mathcal{S}}(A,B)$ and let $\gamma : [0, d_h]$ denote a Ω_h $\tilde{g}$-distance minimizing arc-length parameterized path between $A = \gamma(0)$ and $B = \gamma(d_h)$, such that $\|\dot{\gamma}\| = 1$. Let $\delta = \Pi_\psi(\gamma) = \gamma - \psi(\gamma)\nabla\psi(\gamma)$ be the orthogonal projection of γ onto $\mathcal{S}$. This will be as smooth as γ for small enough h, see Appendix A. For sufficiently small h, the boundary of Ω_h will be smooth, since $\mathcal{S}$ is smooth and no shocks *will be generated (see next Section and Appendix A). So we can assume that γ is C^1 and that $\dot{\gamma}$ is Lipschitz, since it is a* shortest path *within a smooth Riemannian manifold with boundary, see §2.2 above.*

It is evident that (this is a simple but key observation)

$$\mathbf{L}_{\tilde{g}}\{\gamma\} = d_h^{\tilde{g}} \overset{(1)}{\leq} d_{\mathcal{S}}^{g} \overset{(2)}{\leq} \mathbf{L}_g\{\delta\}$$

since

(1) $\mathcal{S} \subset \Omega_h$ *and* $\tilde{g}|_{\mathcal{S}} = g$

(2) δ *need not to be a* g*-shortest path between* A *and* B *on* $\mathcal{S}$.

We then have

$$\begin{aligned}
|\, d_{\mathcal{S}}^{g} - d_h^{\tilde{g}} \,| &\leq |\, \mathbf{L}_g\{\delta\} - \mathbf{L}_{\tilde{g}}\{\gamma\} \,| \;=\; |\, \mathbf{L}_{\tilde{g}}\{\delta\} - \mathbf{L}_{\tilde{g}}\{\gamma\} \,| \\
&\leq \int_0^{d_h} \Big|\; \tilde{g}(\delta)\|\dot{\delta}\| - \tilde{g}(\gamma)\|\dot{\gamma}\| \;\Big|\; dt \\
&\leq \int_0^{d_h} \Big|\tilde{g}(\delta)\|\dot{\delta}\| - \tilde{g}(\delta)\|\dot{\gamma}\|\Big|\; dt \;+\; \int_0^{d_h} |\, \tilde{g}(\delta)\|\dot{\gamma}\| - \tilde{g}(\gamma)\|\dot{\gamma}\| \,|\; dt \\
&= \int_0^{d_h} g(\delta) \Big|\|\dot{\delta}\| - \|\dot{\gamma}\|\Big|\; dt \;+\; \int_0^{d_h} |\tilde{g}(\delta) - \tilde{g}(\gamma)|\; dt \\
&\leq M_g \int_0^{d_h} \|\dot{\gamma} - \dot{\delta}\|\; dt \;+\; K_{\tilde{g}} \int_0^{d_h} \|\gamma - \delta\|\; dt \\
&= M_g \int_0^{d_h} \|\, \nabla\psi(\gamma)\cdot\dot{\gamma}\, \nabla\psi(\gamma) \;+\; \psi(\gamma)\, \mathbf{H}_{\psi}(\gamma)\dot{\gamma} \,\|\; dt \;+\; K_{\tilde{g}} \int_0^{d_h} \|\psi(\gamma)\nabla\psi(\gamma)\|\; dt \\
&\leq M_g \int_0^{d_h} |\, \nabla\psi(\gamma)\cdot\dot{\gamma} \,|\; dt \;+\; h\, M_g \int_0^{d_h} \|\, \mathbf{H}_{\psi}(\gamma)\dot{\gamma} \,\|\; dt \;+\; K_{\tilde{g}}\, h\, d_h
\end{aligned}$$

We now bound the first two terms at the end of the preceding expression.

1. *We first bound the second term in the preceding expression, this will be an ingredient to the bounding of the first term as well.*

   We have:

$$\|\, \mathbf{H}_{\psi}(\gamma)\dot{\gamma} \,\| \;\leq \sup_{\{v:\, \|v\|=1\;;\; p:\, d(p,\mathcal{S})\leq h\}} \|\, \mathbf{H}_{\psi}(p)v \,\| = \sup_{\{p:\, d(p,\mathcal{S})\leq h\}} \max\left(|\lambda(p)|,\, |\mu(p)|\right)$$

   where $\lambda(p)$ *and* $\mu(p)$ *denote the largest and the smallest eigenvalue of* $\mathbf{H}_{\psi}(p)$*, respectively.*

   Now, as we know from Appendix A the maximum absolute eigenvalue of $\mathbf{H}_{\psi}(p)$*,* $K(p)$*, is bounded by*

$$K(p) \leq \frac{\mathcal{M}_{\mathcal{S}}}{1 - |\psi(p)|\mathcal{M}_{\mathcal{S}}}$$

   where $\mathcal{M}_{\mathcal{S}}$ *is the maximum absolute eigenvalue of* $\mathbf{H}_{\psi}|_{\mathcal{S}}$*, that is*

$$\mathcal{M}_{\mathcal{S}} = \sup_{\{x\in\mathcal{S}\}} \max_{\{1\leq i\leq d\}} |\, \lambda_i\left(\mathbf{H}_{\psi}(x)\right) \,|$$

where $\lambda_i(\cdot)$ stands for the i-th eigenvalue of a symmetric matrix.

Then

$$\int_0^{d_h} \| \mathbf{H}_\psi(\gamma(s))\dot{\gamma}(s) \| ds \leq d_h \frac{\mathcal{M}_\mathcal{S}}{1 - h\mathcal{M}_\mathcal{S}}$$

2. *Let us define the function $f : [0, d_h] \to \mathbb{R}$, $f(t) = \psi(\gamma(t))$. Then formally $\dot{f}(t) = \nabla\psi(\gamma(t)) \cdot \dot{\gamma}(t)$ and $\ddot{f}(t) = \mathbf{H}_\psi(\gamma(t))[\dot{\gamma}(t), \dot{\gamma}(t)] + \nabla\psi(\gamma(t)) \cdot \ddot{\gamma}(t)$. Since $\dot{\gamma}(\cdot)$ is Lipschitz, and ψ is regular we can guarantee that $\dot{f}(\cdot)$ is also Lipschitz, so $\ddot{f}(\cdot)$ exists almost everywhere. We want to bound*

$$\int_0^{d_h} \left| \dot{f}(t) \right| \, dt$$

We note first that $f(0) = f(d_h) = 0$, and $|f(t)| \leq h$, $|\ddot{f}(t)| \leq \frac{\mathcal{M}_\mathcal{S}}{1-h\mathcal{M}_\mathcal{S}} + B_{\tilde{g}}$ for almost every $t \in [0, d_h]$. In fact, we have that for those sub-intervals of $[0, d_h]$ in which the shortest path travels through $\partial\Omega_h$, either $f(t) = h$, or $f(t) = -h$ for the whole subinterval, and therefore $f(t)$ is constant for each subinterval, so $\ddot{f}(t) = 0$ there. On the other hand, when γ is in the interior of Ω_h, it is a $\tilde{g}$-geodesic, so its acceleration is bounded by $B_{\tilde{g}}$, as we have seen in Lemma 1. Therefore, we conclude that $|\ddot{f}(t)| \leq |\mathbf{H}_{\psi(\gamma(t))}[\dot{\gamma}(t), \dot{\gamma}(t)]| + B_{\tilde{g}}$. Combining all this we have that for almost every $t \in [0, d_h]$,

$$|\ddot{f}(t)| - B_{\tilde{g}} \leq \sup_{\{\, v : \|v\| = 1;\, d(p,\mathcal{S}) \leq h\}} | \mathbf{H}_\psi(p)[v, v] | \leq \sup_{\{p :\, d(p,\mathcal{S}) \leq h\}} \max\left(|\lambda(p)|,\, |\mu(p)|\right)$$

and the given bound follows as before.

Applying Cauchy-Schwartz *inequality we obtain:*

$$\int_0^{d_h} | \dot{f}(t) | \, dt \leq \sqrt{(d_h) \int_0^{d_h} \dot{f}^2(t) \, dt}$$

Now using Lemma 2:

$$\int_0^{d_h} \dot{f}^2(t) \, dt = \dot{f} f \Big|_0^{d_h} - \int_0^{d_h} f \ddot{f} \, dt = - \int_0^{d_h} f \ddot{f} \, dt$$

$$\leq \int_0^{d_h} | f | \left| \ddot{f} \right| \, dt \leq (d_h)\, h \left(\frac{\mathcal{M}_\mathcal{S}}{1 - h\mathcal{M}_\mathcal{S}} + B_{\tilde{g}} \right)$$

Finally,

$$\int_0^{d_h} | \nabla\psi(\gamma) \cdot \dot{\gamma} | \, dt \leq (d_h) \sqrt{h \left(\frac{\mathcal{M}_\mathcal{S}}{1 - h\mathcal{M}_\mathcal{S}} + B_{\tilde{g}} \right)}$$

Using both computed bounds, we find that

$$| d_\mathcal{S}^g - d_h^{\tilde{g}} | \leq diam(\mathcal{S}) \sqrt{h} \left[M_g \sqrt{\frac{\mathcal{M}_\mathcal{S}}{1 - h\mathcal{M}_\mathcal{S}} + B_{\tilde{g}}} + M_g \sqrt{h} \frac{\mathcal{M}_\mathcal{S}}{1 - h\mathcal{M}_\mathcal{S}} + K_{\tilde{g}} \sqrt{h} \right] \tag{11}$$

■

From the preceding *Lemma* we obtain:

Corollary 1 *For a given point $q \in \mathcal{S}$*

$$d^{\tilde{g}}_{\Omega_h}\Big|_{\mathcal{S}}(q,\cdot) \overset{h\downarrow 0}{\rightrightarrows} d^{g}_{\mathcal{S}}(q,\cdot) \quad \text{in } \mathcal{S}$$

Remark 1 *The rate of convergence obtained with the techniques shown above is of order $\sqrt{h}$. A quick look over the proof of convergence shows that the term responsible for the $h^{1/2}$ rate is $\int_0^{d_h} \left| \dot{f}(t) \right| dt$. All other terms have the higher rate of h. Suppose we can find a* finite *collection of (disjoint) intervals $I_i = (a_i, b_i)$ such that $sgn(\dot{f})$ is constant (f is monotonic) within each I_i, $\cup_{i=1}^{N} I_i \subseteq [0, d_h]$ where N is the cardinality of that collection of intervals, and $\dot{f}(t) = 0$ for $t \in [0, d_h] \setminus \cup_{i=1}^{N} I_i$. Then, we could write:*

$$\begin{aligned}
\int_0^{d_h} \left| \dot{f}(t) \right| dt &= \sum_{i=1}^{N} sgn(\dot{f})\Big|_{(a_i,b_i)} \int_{a_i}^{b_i} \dot{f}(t)\, dt \\
&= \sum_{i=1}^{N} sgn(\dot{f})\Big|_{(a_i,b_i)} (f(b_i) - f(a_i)) = \sum_{i=1}^{N} |f(b_i) - f(a_i)| \\
&\leq \sum_{i=1}^{N} (|f(b_i)| + |f(a_i)|) \\
&\leq 2Nh \quad \textit{since} \quad f(t) = \psi(\gamma(t)) \ \textit{and} \ \gamma(\cdot) \ \textit{travels through} \ \Omega_h,
\end{aligned}$$

obtaining a higher rate of convergence, h. It is quite convincing that cases where $N = \infty$ can be considered pathological. We then argue that for all practical purposes the rate of convergence achieved is h. Notwithstanding, we are currently studying the space of surfaces and metrics g for which we can guarantee that $N < \infty$, and advances in this subject will be reported elsewhere.

This shows that we can approximate the intrinsic distance with the Euclidean one on the offset band Ω_h. Moreover, as we will detail below, the approximation error is of the same order as the theoretical numerical error in fast marching algorithms. Thereby, we can use fast algorithms in Cartesian grids to compute intrinsic distances (on implicit/implicitized surfaces), enjoying their computational complexity without affecting the convergence rate given by the underlying numerical approximation scheme.

3 Numerical Implementation and its Theoretical Error

In this section we first discuss the simple modification that needs to be incorporated into the (Cartesian) fast marching algorithm in order to deal with Euclidean spaces with manifolds. We then propose a way of estimating the (now discrete) offset h, and bound the total numerical error of our algorithm, thereby showing our assertion that the error with our algorithm is of the same order than the one obtained with the fast marching algorithm for Cartesian grids (or triangulated 3D surfaces).

As stated before, we are dealing with the numerical implementation of the Eikonal equation inside an open, bounded and connected domain Ω (this will later become the offset Ω_h). The general equation, when $P(x)$ is the weight (it becomes $\tilde{g}$ for our particular case), is given by

$$\begin{cases} \|\nabla f(x)\| = P(x) \quad \forall\, x \in \Omega \\ f(r) = 0 \end{cases} \tag{12}$$

being r the seed point. Note that following the results in the previous section, we are now dealing with the Eikonal equation in Euclidean space, and then the Euclidean gradient is used above.

The upwind numerical scheme to be used for this equation is of the form ($\Delta x_1 = \Delta x_2 = \ldots = \Delta x_d \;=\; \Delta x$) [48]:

$$\begin{cases} \sum_{j=1}^{d} max^2(\widehat{f}(p) - m_j, 0) = (\Delta x)^2\, P^2(p) \\ m_j = min(\widehat{f}(p + \Delta x \vec{e}_j), \widehat{f}(p - \Delta x \vec{e}_j)) \end{cases} \tag{13}$$

where $\widehat{f}$ is the numerically computed value of f for every point p in the discrete domain

$$\mathcal{D}(\Omega, \Delta x) \stackrel{\triangle}{=} \Omega \cap (\mathbb{Z}\Delta x)$$

Here, $\vec{e}_j$ with $j = 1, 2, ..., d$, are the elements of the canonical basis of $\mathbb{R}^d$.

We now describe the fast marching algorithm for solving the above equation. For this we follow the presentation in [51]. For clarity we write down the algorithm in *pseudo-code* form. Details on the original fast marching method on Cartesian grids can be found in the mentioned references.

At all times there are 3 kinds of points under consideration:

- **NarrowBand**. These points have to them associated an already guessed value for $\widehat{f}$, and are immediate neighbors to those points whose value has already been "frozen."
- **Alive**. These are the points whose $\widehat{f}$ value has already been frozen.
- **Far Away**. These are points that haven't been processed yet, so no tentative value has been associated to them. For that reason they have $\widehat{f} = \infty$.

The steps of the algorithm are:

- *Initialization*:
  1. Set $\widehat{f} = 0$ for every point belonging to the set **Alive** (these are the seed/s point/s).
  2. Find a tentative value of $\widehat{f}$ for every **Neighbor** of an **Alive** point and tag them **NarrowBand**.
  3. Set $\widehat{f} = \infty$ for all the remaining points in the discrete domain.
- *Advance*:
  1. Beginning of loop: Let (p_{min}) be the point $\in$ [**NarrowBand**] which takes the least value of $\widehat{f}$.
  2. Insert the point p_{min} to the set [**Alive**] and remove it from [**NarrowBand**].

3. Tag as **Neighbors** all those points in the discrete domain that can be written in the form $p_{min} \pm \Delta x\, \vec{e}_j$, and belonging to [**NarrowBand**] $\cup$ [**FarAway**]. If a **Neighbor** is in [**FarAway**], remove it from that set and insert it to [**NarrowBand**].
4. Recalculate $\widehat{f}$ for all **Neighbors** using equation (13)
5. Back to the beginning.

The boundary conditions are taken such that points beyond the discrete domain have $\widehat{f} = \infty$.

The condition that is checked all the time, and that really defines the domain the algorithm is working within, is the one that determines if a certain point q is **Neighbor** of a given point p that belongs to the domain. The only thing one has to do in order to make the algorithm work in the domain Ω_h specified by $\{x \in I\!R^d : |\psi(x)| \leq h\}$ is change the way the **Neighbor** checking is done. More precisely, we should check

$$q \in \textbf{Neighbor}(p) \text{ iff } \{(|\psi(q)| \leq h) \;\&\&\; (q \text{ can be written like } p \pm \Delta x\, \vec{e}_j\ )\}$$

the emphasis here being on the test "$|\psi(q)| \leq h$." We could also achieve the same effect by giving an infinite weight to all points outside Ω_h, that is, we treat the outside of Ω_h as an obstacle. That is, with an extremely simple modification to the fast marching algorithm, we make it work as well for distances on manifolds with boundary, and therefore, for intrinsic distances on implicit surfaces. This is of course supported by the convergence results in the previous section and the analysis on the numerical error presented below.

3.1 Bounding the Offset h

We now present a technique to estimate h, the size of the offset of the hypersurface $\mathcal{S}$ that actually defines the computational domain Ω_h. The bounds on h are very simple. On one hand, we need h to be large enough so that the upwind scheme can be implemented, meaning that h has to be large enough to include the stencil used in the numerical implementation. On the other hand, h has to be small enough to guarantee that Ω_h remains simply connected with smooth boundaries and that $\tilde{g}$ remains smooth inside Ω_h.

Let $\mathcal{M}_{\mathcal{S}}$ be as before a bound for the absolute sectional curvature of $\mathcal{S}$, and let Δx be the grid size. In addition, let W be the maximal offsetting of the surface $\mathcal{S}$ that guarantees that the resulting set remains connected and different parts of the boundary of that set do not touch each other. We show below that a suitable bounding of h is

$$\Delta x \sqrt{d} \;<\; h \;<\; \min\left\{\frac{1}{\mathcal{M}_{\mathcal{S}}}, W\right\}. \tag{14}$$

Let us introduce some additional notation. We denote by *cell* to the unit cell of the computational grid. Let x be a point in Ω_h, we denote by $n(x)$ the number of cells $C_1(x), \ldots, C_{n(x)}$ that contain x. It is clear that if $x \in \mathcal{D}(\Omega_h, \Delta x)$ (it is a grid point), then x is contained in 2^d cells having x as a vertex. It is also clear that $n(x) \leq 2^d$. For a given cell C we call $P(C)$ the set of points of $\mathcal{D}(\Omega_h, \Delta x)$ that compose C (i.e., its vertices). We will denote by $\mathbf{C}(x)$ the set $\bigcup_{i=1}^{n(x)} C_i(x)$, and by $\mathbf{P}(x)$ the set $\left(\bigcup_{i=1}^{n(x)} P(C_i(x))\right)^*$ where the "*" means that we remove repeated elements (points).

The lower bound comes from forcing that for every $x \in \mathcal{S}$, all points in $\mathbf{C}(x)$ lie within Ω_h (note of course that we want h as small as possible). That is:

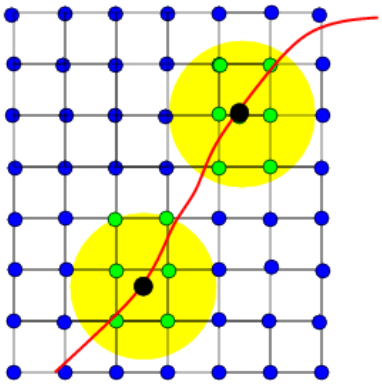

Figure 3: We depict the situation that leads to the lower bound for h in the $2D$ case. In red: the curve. In black: the centers of $\overline{B}(x \in \mathcal{S}, d^{1/2}\Delta x)$. In green: the points of $\mathcal{D}(\Omega_h, \Delta x)$ that fall inside $\overline{B}(x, d^{1/2}\Delta x)$ for some $x \in \mathcal{S}$, and in blue those that don't.

$$\bigcup_{x \in \mathcal{S}} \mathbf{C}(x) \subset \Omega_h$$

Once again, this constraint comes in order to guarantee that there are "enough" points to make the discrete calculations. We try to make $\mathbf{C}(x) \subset \overline{C}(x, l)$, where $C(x, l)$ stands for the hypercube centered in x, with side length $2l$, and sides parallel to the gridding directions. The worst scenario is when x is a point in the discrete domain, and we must have $l \geq \Delta x$. Finally, we observe that $C(x, l) \subset B(x, l\sqrt{d})$. The condition then becomes

$$\bigcup_{x \in \mathcal{S}} B(x, \Delta x\sqrt{d}) \subset \Omega_h = \bigcup_{x \in \mathcal{S}} B(x, h)$$

which provides the lower bound, $h > \Delta x\sqrt{d}$.

The upper bound includes two parts. First, we shouldn't go beyond W, since if we do so, different parts of the offset surface might touch each other, situation which can even create a non-simple band Ω_h. The second part of the upper bound comes from seeking that when traveling on a characteristic line of ψ at a point p of $\mathcal{S}$, no shocks occur inside Ω_h. It is a simple fact that this won't happen if $h < \frac{1}{\mathcal{M}_{\mathcal{S}}}$, see Appendix A. It is extremely important to guarantee this both to obtain smooth boundaries for Ω_h and to obtain smooth extensions of the metric g ($\tilde{g}$).

Note of course that in general, h and also Δx can be position dependent. We can use an adaptive grid, and in places where the curvature of $\mathcal{S}$ is high, or places where high accuracy is desired, we can make Δx small.

3.2 The Numerical Error

It is time now to explicitly bound the numerical error of our proposed method. As stated above, it is our goal to formally show that we are within the same order that the computationally optimal (fast marching) algorithms for computing distance functions on Cartesian grids. Note that the numerical error for the fast marching algorithm on triangulated surfaces has not been reported, although it is of course bounded by the Cartesian one (since this provides a particular "triangulation").

3.2.1 Numerical Error Bound of the Cartesian Fast Marching

The aim of this section is to bound a quantity that measures the difference between the numerically computed value $\widehat{d^g_{\mathcal{S}}}(p,\cdot)$ and the real value $d^g_{\mathcal{S}}(p,\cdot)$. Any such quantity will be comparing both functions on $\mathcal{S}$, but in principle the numerically computed value will not be defined all over the hypersurface. So we will be dealing with an interpolation stage, that we comment further below in §3.2.2.

Let us fix a point $p \in \mathcal{S}$, and let $\widehat{f}(\cdot)$ be the numerically computed solution (according to (13)), and $f(\cdot)$ the *real* viscosity solution of the problem (12). The approximation error is then bounded by (see [48])

$$\max_{p\in\mathcal{D}(\Omega,\Delta x)} |\widehat{f}(p) - f(p)| \leq C_L(\Delta x)^{\frac{1}{2}} \tag{15}$$

where C_L is a constant. In practice, however, the authors of [48] observed first order accuracy. As we have seen, we have also find an error of order $h^{1/2}$ for the general approximation of the weighted intrinsic distance on $\mathcal{S}$ with the distance in the band Ω_h, and a practical order of h (see Remark 1 and Theorem 2).

Before proceeding with the presentation of the whole numerical error of our proposed algorithm, we need the following simple Lemma whose proof we omit.

Lemma 3 *For a convex set $D \subset \Omega$, and $y, z \in D$, f satisfies*

$$|f(z) - f(y)| \leq \|P\|_{L^\infty(\Omega)}\|z - y\|$$

Remark 2 *Using the preceding Lemma and (15), it is easy to see that for x such that $\mathbf{C}(x) \subset \Omega$:*

$$|\widehat{f}(p) - \widehat{f}(q)| \leq 2C_L(\Delta x)^{\frac{1}{2}} + \|P\|_{L^\infty(\Omega)}\sqrt{d}\Delta x, \quad \forall p, q \in \mathbf{P}(x) \tag{16}$$

a relation we will shortly use.

3.2.2 The Interpolation Error

Since following our approach we are now computing the distance function in the band Ω_h, in the corresponding discrete Cartesian grid, we have to interpolate this to obtain the distance on the zero level-set $\mathcal{S}$. This interpolation produces a numerical error which we now proceed to bound.

Given the function $\zeta : \mathcal{D}(\Omega, \Delta x) \to I\!R$, we define the function $\mathcal{I}(\zeta) : \Omega \to I\!R$ through an interpolation scheme. We will assume that the interpolation error is bounded in the following way:[8]

$$\sup_{y\in\mathbf{P}(x)} |\zeta(y) - \mathcal{I}(\zeta)(x)| \leq \max_{z\in\mathbf{P}(x)} \zeta(z) - \min_{z\in\mathbf{P}(x)} \zeta(z)$$

for every $x \in \Omega$

[8] One may imagine several interpolation schemes satisfying this not-stringent-at-all condition.

3.2.3 The Total Error

We now present the complete error (numerical plus interpolation) introduced by our algorithm, without considering the possible error in the computation of $\tilde{g}$ (or in other words, we assume that the weight was already given in the whole band Ω_h).

Let p be a point in $\mathcal{S}$. We denote by

- $d^g_{\mathcal{S}}(p,\cdot) : \mathcal{S} \to I\!R$ the *intrinsic* g-distance function from p to any point in $\mathcal{S}$.
- $d^{\tilde{g}}_h(p,\cdot) : \Omega_h \to I\!R$ the $\tilde{g}$-distance function from p to any other point in Ω_h.
- $\widehat{d^{\tilde{g}}_h}(p,\cdot) : \mathcal{D}(\Omega_h, \Delta x) \to I\!R$ the *numerically computed* value of $d^{\tilde{g}}_h(p,\cdot)$ to any point in the discrete domain.
- $\mathcal{I}\left(\widehat{d^{\tilde{g}}_h}\right)(p,\cdot) : \mathcal{S} \to I\!R$ the result of interpolating $\widehat{d^{\tilde{g}}_h}$ (that's only specified for points in $\mathcal{D}(\Omega_h, \Delta x)$) to points in $I\!R^d \supset \mathcal{S}$.

The goal is then to bound $\left\| d^g_{\mathcal{S}}(p,\cdot) - \mathcal{I}\left(\widehat{d^{\tilde{g}}_h}\right)(p,\cdot) \right\|_{L^\infty(\mathcal{S})}$, and we proceed to do so now.

Let x be in $\mathcal{S}$ and y in $\mathbf{P}(x)$, then:

$$\left| d^g_{\mathcal{S}}(p,x) - \mathcal{I}\left(\widehat{d^{\tilde{g}}_h}\right)(p,x) \right| \le |d^g_{\mathcal{S}}(p,x) - d^{\tilde{g}}_h(p,x)| + |d^{\tilde{g}}_h(p,x) - d^{\tilde{g}}_h(p,y)| \tag{17}$$
$$+ \left| d^{\tilde{g}}_h(p,y) - \widehat{d^{\tilde{g}}_h}(p,y) \right| + \left| \widehat{d^{\tilde{g}}_h}(p,y) - \mathcal{I}\left(\widehat{d^{\tilde{g}}_h}\right)(p,x) \right|$$

and using Proposition 2, Lemma 3, (15) and simple manipulations (in that order) we obtain:

$$\left| d^g_{\mathcal{S}}(p,x) - \mathcal{I}\left(\widehat{d^{\tilde{g}}_h}\right)(p,x) \right| \le C(h)\, diam(\mathcal{S})\, h^{\frac{1}{2}} + M_{\tilde{g}}\|x-y\| + C_L(\Delta x)^{\frac{1}{2}}$$
$$+ \left(\max_{y \in \mathbf{P}(x)} \widehat{d^{\tilde{g}}_h}(p,y) - \min_{y \in \mathbf{P}(x)} \widehat{d^{\tilde{g}}_h}(p,y) \right)$$

The last term can be dealt with using (16). Taking into account that we will always choose $h = C_x^2\, \Delta x \sqrt{d}$ for some constant $C_x > 1$ (as we saw in §3.1), we conclude:

$$\left\| d^g_{\mathcal{S}}(p,\cdot) - \mathcal{I}\left(\widehat{d^{\tilde{g}}_h}\right)(p,\cdot) \right\|_{L^\infty(\mathcal{S})} \le (\Delta x)^{\frac{1}{2}}\, \mathcal{C}(\Delta x; \mathcal{S}) \tag{18}$$

where $\mathcal{C}(\Delta x; \mathcal{S})$ goes to a constant (that depends on $\mathcal{S}$) as $\Delta x \downarrow 0$, and this provides the desired bound. Note that as announced below, the error of our algorithm is of the same (theoretical) order as the error of the fast marching on Cartesian grids, thereby justifying our approach.

We have then obtained that the use of an Euclidean approximation in the band Ω_h to the intrinsic distance function on the level-set $\mathcal{S}$ doesn't change the order of the whole numerical approximation. In the most general case, the theoretical bound is of order $h^{1/2}$, while for all practical purposes is of order h (when we replace in the computation above the practical bounds for the fast marching algorithm and for the distance approximation).

To conclude, let's point out that since we are working within a narrow band (Ω_h) of the surface $\mathcal{S}$, we are actually not increasing the dimensionality of the problem. We can then work with a Cartesian grid while keeping the same dimensionality as if we were working on the surface.

4 Experiments

We now present a number of 3D examples of our algorithm. Recall that although all the examples are given in 3D, the theory presented above is valid for any dimension $d \geq 3$.

Two classes of experimental results are presented. We show a number of intrinsic distance functions for implicit surfaces, as well as geodesics computed using these functions. In order to compute interesting geodesics, we use also non-uniform weights, permitting the computation of crest/valleys, and optimal paths with obstacles. We also experimentally compare our results with those obtained on triangulated surfaces using the fast marching technique developed by Kimmel and Sethian (for this we use the results and software reported in [6]).

Figures 4, 5, and 6 show the intrinsic distance function for implicit surfaces computed with the method here proposed ($g = 1$). An arbitrary seed point on the implicit surface has been chosen, and pseudo colors are used to improve the visualization. Red corresponds to low values of the distance and blue to the high ones. We observe that, as expected, the distance (colors) vary smoothly, and that close points have similar colors and far points have very different colors (close and far measured on the surface of course).

In Figure 7 we compare the result of our approach with that of fast marching on a triangulated surface (this later computed with the package reported in [6]), while in Figure 8 we show level lines of the intrinsic distance function computed with the technique here proposed.

Before concluding this part of the experiments, let's give some technical details on the implementation. The code for the examples in this paper was written in `C++`. For visualization purposes, VTK was used. Most of the "hard code" was done taking advantage of Blitz++'s double templatized arrays and related routines, see [9]. The implicit models used in this paper were obtained from [65] (other techniques, e.g., [39], could be used as well). All the code was compiled and run in a $450\,Mhz$ Pentium *III*, with $256\,Mb$ of RAM, working under Linux (RedHat 6.2). The compiler used was `egcs-2.91.66` and the level of optimization was 3. In the table below we show running times of the intrinsic distance map algorithm for some of the implicit models we used, along with the corresponding *offset*-value (h) and size and number of grid points in Ω_h for each model.

Model	**Size**	$\#\mathcal{D}(\Omega_h, \Delta x)$	**h**	**Running Time** (secs)
Brain	$122 \times 142 \times 124$	$168,603$	1.75	9.4
Bunny	$81 \times 80 \times 65$	$38,107$	1.75	1.99
Knot	$80 \times 81 \times 44$	$16,095$	1.0	0.76
Sphere	$70 \times 70 \times 70$	$11,800$	1.75	0.65
Torus	$64 \times 64 \times 64$	$21,704$	1.75	1.16
Teapot	$80 \times 55 \times 46$	$24,325$	1.75	1.22

4.1 Geodesics on Implicit Surfaces

To find geodesic curves on the implicit surface, we back track starting from a specified target point toward the seed point, while traveling on the surface in the direction given by the (negative) intrinsic-distance gradient. This means that after we have computed the intrinsic distance function as explained above, we have to solve the following **ODE** (which obviously keeps the curve on $\mathcal{S}$):

$$\begin{cases} \dot{\gamma} = -\nabla_{\psi} d^{\tilde{g}}_{\Omega_h}(\gamma) \\ \gamma(0) = p \in \mathcal{S} \end{cases}$$

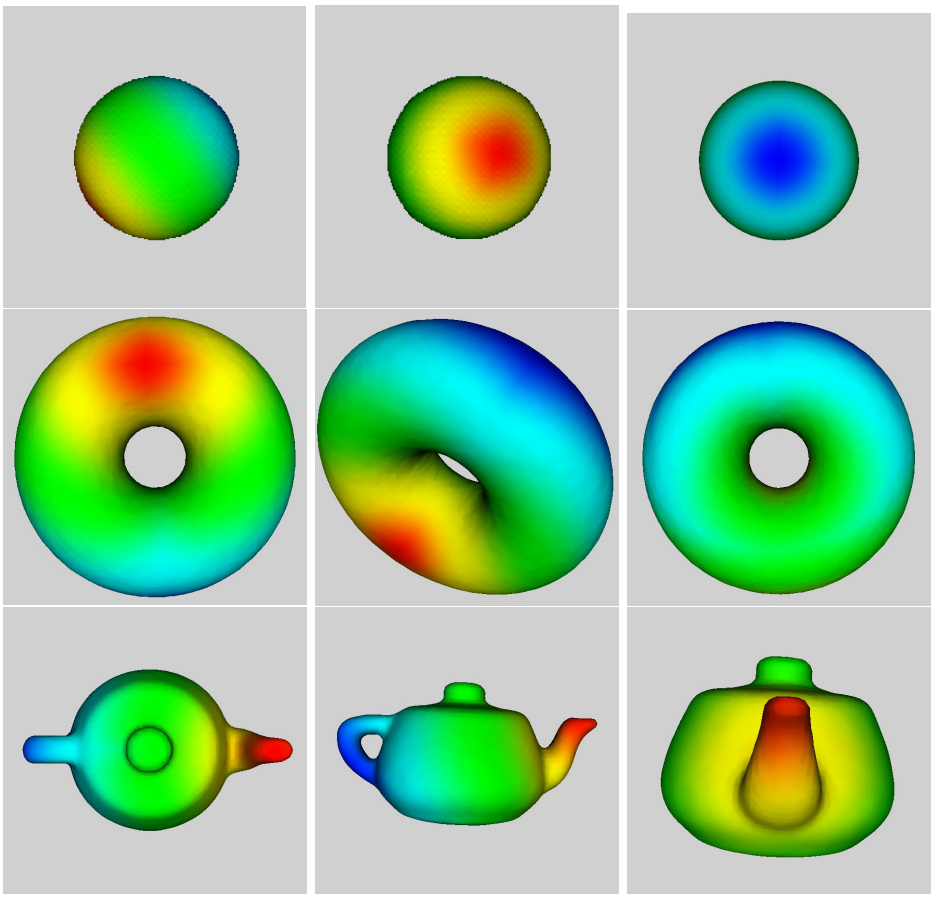

Figure 4: Distance maps from a point on the sphere, torus and teapot (three views are presented for each model).

where $\nabla_\psi d^{\tilde{g}}_{\Omega_h}(p) \triangleq \nabla d^{\tilde{g}}_{\Omega_h}(p) - \left(\nabla d^{\tilde{g}}_{\Omega_h}(p) \cdot \nabla\psi(p)\right) \nabla\psi(p)$ is the gradient of $d^{\tilde{g}}_{\Omega_h}$ at $p \in \mathcal{S}$ projected onto the tangent space to $\mathcal{S} = \{\psi = 0\}$ at p. Since we must discretize the above equation, one can no longer assume that at every instant the geodesic path γ will lie on the surface, so a projection step must be added. In addition, since all quantities are known only at grid points, an interpolation scheme must be used to perform all evaluations at positions given by γ. We have used a simple Runge-Kutta integration procedure, with adaptive step, namely an ODE23 procedure.

Before presenting examples of geodesic curves, we should note that we are assuming that $\nabla_\psi d^{\tilde{g}}_{\Omega_h}$ is a good approximation of $\nabla_{\mathcal{S}} d^g_{\mathcal{S}}$ (and not just $d^{\tilde{g}}_{\Omega_h}$ a good approximation of $d^g_{\mathcal{S}}$ as we have previously proved). Bounding the error between these two gradients, e.g., using the framework of viscosity solutions, is the subject of current work.

The figures described next illustrate the computation of geodesic curves on implicit surfaces for different weights g. In all the figures the geodesic curve is drawn on top of the surface, which is colored as before, colors indicating the intrinsic weighted distance.

In Figure 9 we present both the geodesic curve computed with our technique and the one computed with the fast marching algorithm on triangulated surfaces following the implementation reported in [6].

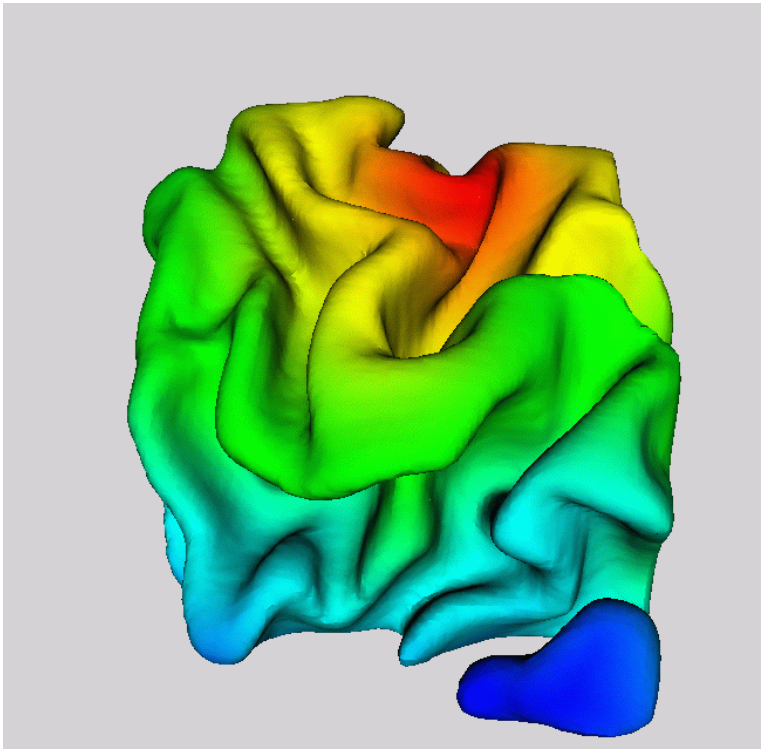

Figure 5: Distance map from a point on a portion of white/gray matter boundary of the cortex.

In Figure 10 we show the computation of sulci (valleys) on an implicit surface representing the boundary between the white and gray matter in a portion of the human cortex (data obtained from MRI). Here the (extended) weight $\tilde{g}$ is a function of the mean curvature given by [6]

$$\tilde{g}_{valley}(x) = \omega + \left(\mathbf{M}(x) - \min_{y \in \Omega_h} \mathbf{M}(y) \right)^p$$

where $\mathbf{M}$ stands for the mean curvature of the level sets of ψ, so it is computed simply as $\mathbf{M}(x) = \Delta\psi(x)$. In the example presented we used $\omega = 100$ and $p = 3$. More details on the use of this approach for detecting valleys (and creases) can be found in [6] and in the references therein.

In Figure 11 we show the computation of geodesic curves with obstacles on implicit surfaces. This is an important computation for topics such as motion planning on surfaces.

5 Extensions

5.1 General Metrics: Solving Hamilton-Jacobi Equations on Implicit Surfaces

Since the very beginning of our exposition we have restricted ourselves to *isotropic* metrics. As stated in the introduction, this already has a tremendous amount of applications, and just a few were shown in the previous section. Since the fast marching approach has been recently extended to more general Hamilton-Jacobi equations by Osher and Helmsen [44], we are immediately tempted to extend our framework to these equations as well (these equations have applications in important areas such as adaptive mesh generation on manifolds, [27], and semiconductors manufacturing).

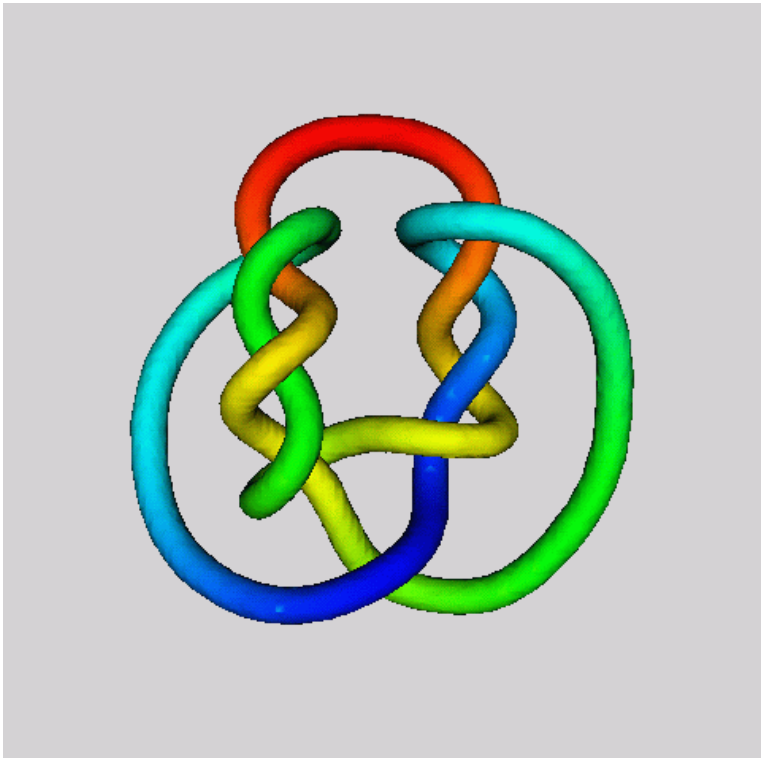

Figure 6: Distance map from one seed point on a knot. In this picture we evidence that the algorithm works well for quite convoluted geometries (as long as h is properly chosen). Note how points close in the Euclidean sense but far away in the intrinsic sense receive very different colors, indicating their large (intrinsic) distance.

Then, we are led to investigate the extension of our algorithm to general metrics of the form, $\mathbf{G} : \mathcal{S} \to I\!R^{d \times d}$, that is, a positive definite 2-tensor. Our new definition of weighted length becomes

$$\mathbf{L}_{\mathbf{G}}\{\mathcal{C}\} \stackrel{\triangle}{=} \int_a^b \sqrt{\mathbf{G}(\mathcal{C}(t))[\dot{\mathcal{C}}(t), \dot{\mathcal{C}}(t)]}\ dt$$

and the problem is to find for every $x \in \mathcal{S}$ (for a fixed $p \in \mathcal{S}$)

$$d_{\mathcal{S}}^{\mathbf{G}}(x, p) \stackrel{\triangle}{=} \inf_{\mathcal{C}_{px}[\mathcal{S}]} \{\mathbf{L}_{\mathbf{G}}(\mathcal{C})\} \tag{19}$$

As before, we attempt to solve the approximate problem in the band Ω_h, with an extrinsic distance:

$$d_{\Omega_h}^{\tilde{\mathbf{G}}}(x, p) \stackrel{\triangle}{=} \inf_{\mathcal{C}_{px}[\Omega_h]} \{\mathbf{L}_{\tilde{\mathbf{G}}}(\mathcal{C})\} \tag{20}$$

where

$$\mathbf{L}_{\tilde{\mathbf{G}}}\{\mathcal{C}\} \stackrel{\triangle}{=} \int_a^b \sqrt{\tilde{\mathbf{G}}(\mathcal{C}(t))[\dot{\mathcal{C}}(t), \dot{\mathcal{C}}(t)]}\ dt$$

for an adequate extension $\tilde{\mathbf{G}}$ of $\mathbf{G}$. The solution of the extrinsic problem satisfies (in the viscosity sense) the Eikonal equation

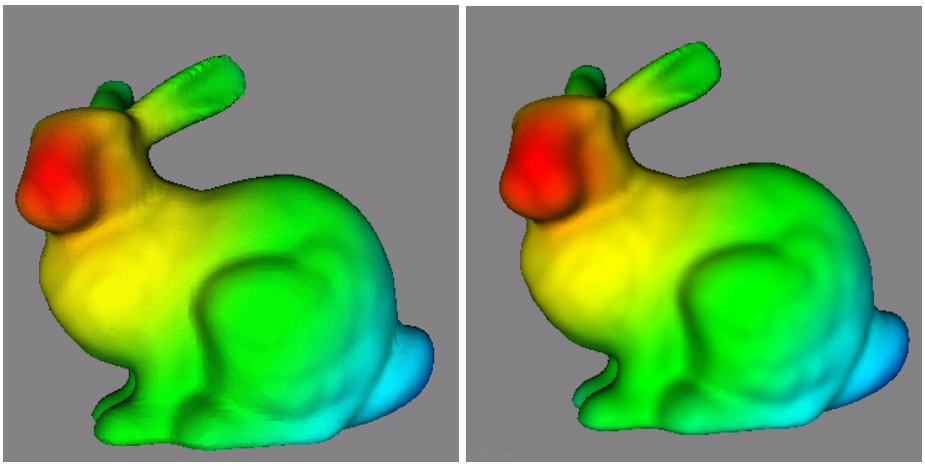

Figure 7: Distance map from a single seed point (situated at the nose) on an implicit bunny ($g = 1$). The figure on the left was obtained with the implicit approach here presented, while the one on the right was derived with the fast marching on triangulated surfaces technique.

$$(\tilde{\mathbf{G}}^{-1})(x)[\nabla d^{\tilde{\mathbf{G}}}_{\Omega_h}, \nabla d^{\tilde{\mathbf{G}}}_{\Omega_h}] = 1 \tag{21}$$

The first issue now is the numerical solvability of the preceding equation using a fast marching type of approach. Osher and Helmsen, [44], have extended the capabilities of the fast marching to deal with Hamilton-Jacobi equations of the form

$$H(x, \nabla f) = a(x)$$

for geometrically based Hamiltonians $H(x, \vec{p}) : \Omega(\subset I\!R^d) \times I\!R^d \to I\!R$ that satisfy

$$\begin{cases} H(x, \vec{p}) > 0 \text{ if } \vec{p} \neq \vec{0} \\ H(x, \vec{p}) \text{ is homogeneous of degree 1 in } \vec{p} \end{cases} \tag{22}$$

We easily observe that these conditions hold for (21) considering $H(x, \vec{p}) \stackrel{\triangle}{=} \sqrt{(\tilde{\mathbf{G}}^{-1})(x)[\vec{p}, \vec{p}]}$, and therefore we can solve these type of Hamilton-Jacobi equations (the extrinsic problem) with the extended fast marching algorithm.

In order to show that our framework is valid for these equations as well, all what we basically need to do is to prove that the extrinsic distance (20) on the offset Ω_h converges to the intrinsic one on the implicit surface $\mathcal{S}$, i.e., (19). This can be done repeating the steps in the convergence proof previously reported in §2.3 for isotropic metrics. Combining this with the results of Osher and Helmsen we then obtain that our framework can be applied to a larger class of Hamilton-Jacobi equations.

5.2 Non Implicit Surfaces

The framework we presented was here developed for implicit surfaces, although it applies to other surface representations as well. First, if the surface is originally given in polygonal or triangulated form, or even as a set of unconnected points and curves, we can use a number of available techniques, e.g., [33, 39, 46, 54, 63, 65] (and some very nice public domain software [39]), to first implicitize

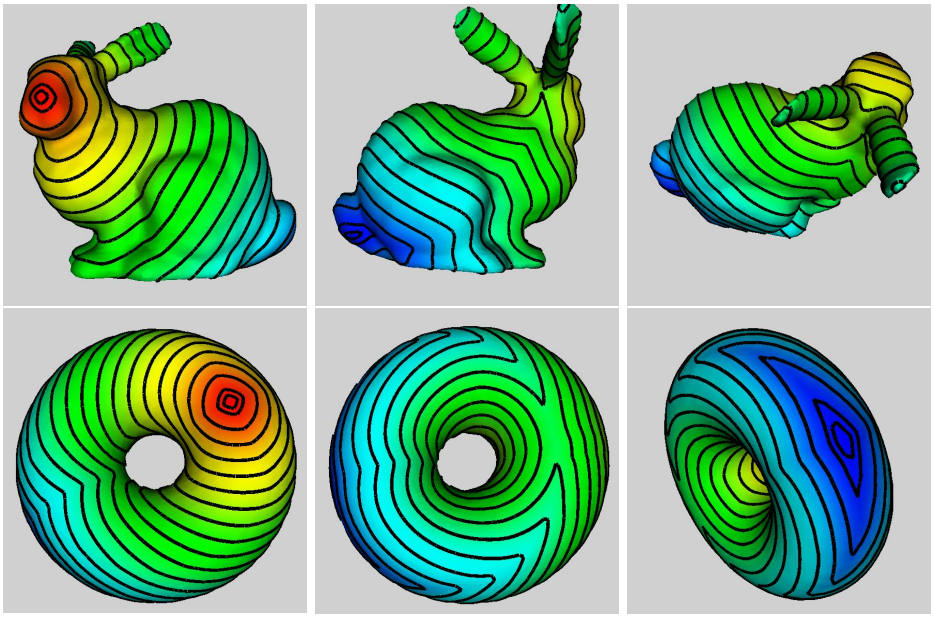

Figure 8: Top: Level lines for the intrinsic distance function depicted in Figure 7 (left). Bottom: Level lines for the intrinsic distance function depicted in Figure 4 (second row). In both rows, the (22) levels shown are 0.03, 0.05, 0.1, ..., 0.95, 0.97 percent of the maximum value of the intrinsic distance, and the coloring of the surface corresponds to the intrinsic distance function. Three views are presented. Note the correct separation between adjacent level lines. Note also how these lines are "parallel".

the surface and then apply the technique here proposed.[9] Note that the implicitation needs to be done only once per surface as a pre-processing step, and will remain valid for all subsequent uses of the surface. Moreover, as we have seen, all what we need is to have a Cartesian grid in a small band around the surface $\mathcal{S}$. Therefore, there is no explicit need to perform an implicitation of the given surface representation. For example, if the surface is given by a cloud of unconnected points, we can compute distances intrinsic to the surface defined by this cloud, as well as intrinsic geodesic curves, without explicitly computing the underlying surface. All what is needed is to embed this cloud of points in a Cartesian grid and consider only those points in the grid at a distance h or less from the points in the cloud. The computations are then done on this band.

6 Concluding Remarks

In this article we have presented a novel computationally optimal algorithm for the computation of intrinsic distance functions and geodesics on implicit hyper-surfaces. The underlying idea is based on using the classical Cartesian fast marching algorithm in an offset bound around the given surface. We have provided theoretical results justifying this approach and presented a number of experimental examples. The technique can also be applied to 3D triangulated surfaces, or even surfaces represented by clouds of unconnected points, after these have been embedded in a Cartesian grid with proper boundaries. The conclusion is that there is no need to further develop algorithms for intrinsic distances and geodesics on hyper-surfaces, original Cartesian approaches are sufficient. We have also discussed that the approach is valid for more general Hamilton-Jacobi equations as well.

[9] The same techniques can be applied to transform any given implicit function into a distance one.

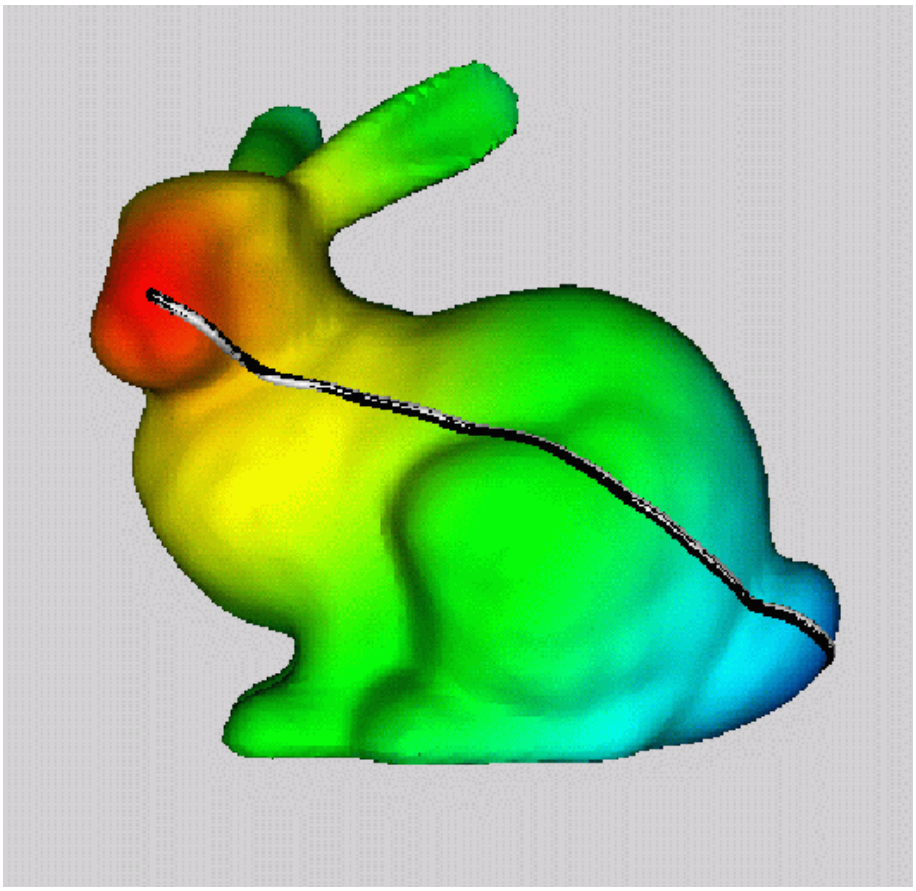

Figure 9: Distance map and geodesic curve between two points on an implicit bunny. We show two geodesics superimposed, the black one is the one obtained via the implicit back propagation described in the text, while the white one is obtained when performing the back propagation computation in the triangulated surface. It is important to note that in both cases the distance function used is the one computed with our implicit approach; to feed the data to the triangulated surfaces back-propagation algorithm, we first interpolated the intrinsic distance to points onto the triangulated surface. We can clearly see that both geodesics overlap almost entirely.

Many questions remain open. First, we are currently working on tighter bounds for the error between $d^{\tilde{g}}_{\Omega_h}$ and $d^{g}_{\mathcal{S}}$, as well as bounds for the error between their corresponding derivatives. We are interested in extending the framework here presented to the computation of distance functions on high codimension surfaces and general embeddings. More generally, it remains to be seen what class of intrinsic Hamilton-Jacobi (or in general, what class of intrinsic PDE's) can be approximated with equations in the offset band Ω_h. In an even more general approach, what kind of intrinsic equations can be approximated by equations in other domains, being offsets just a particular and important example. Even if fast marching techniques do not exists for these equations, it might be simpler and even more accurate to solve the approximating equations in these domains than in the original surface $\mathcal{S}$. The framework here presented then not only offers a solution to a fundamental problem, but it also opens the doors to a new area of research.

Acknowledgments

Many colleagues and friends helped us during the course of the work here reported, and they deserve our most sincere acknowledgment. Prof. Miguel Paternain, in the early stages of this work, pointed out very relevant references. Prof. Stephanie Alexander, Prof. David Berg, and Prof. Richard Bishop helped us with our questions about geodesics on manifolds with boundaries. Prof. Stanley Osher helped us with issues regarding the fast marching algorithm in general, and his recent extensions in particular, while Prof. Ron Kimmel discussed with us topics related to his triangulated work and provided general comments on the paper. Prof. Osher also helped with the literature on rate of convergence of Hamilton-Jacobi numerical approximations. Prof. Omar Gil helped us with some deep insights into technical aspects of the work. This work benefited from rich conversations with Alberto Bartesaghi (who also provided us with some of his code to extract geodesics in triangulated surfaces), Cesar Perciante, Luis Vásquez, Federico Lecumberry, and Prof. Gregory Randall. We also enjoyed important discussion with Prof. Vicent Caselles on the density of zeros of general functions. We also thank Prof. Luigi Ambrosio for his comments on the regularity of level-sets of distance functions. We are indebted to Alvaro Pardo for his careful reading of the manuscript and for the suggestions he made. Prof. H. Zhao provided some of the implicit surfaces. FM in particular wants to thank Omar Gil, who has been a great and patient teacher, and Gregory Randall for his constant support and encouragement. FM performed part of this work while visiting the University of Minnesota. This work was partially supported by a grant from the Office of Naval Research ONR-N00014-97-1-0509, the Office of Naval Research Young Investigator Award, the Presidential Early Career Awards for Scientists and Engineers (PECASE), a National Science Foundation CAREER Award, by the National Science Foundation Learning and Intelligent Systems Program (LIS), by the Comisión Sectorial de Investigación Científica (CSIC), the Comisión Académica de Posgrado (CAP), and Tecnocom through a scholarship to FM.

A Distance Maps in Euclidean Space

We now present a few important results on distance maps. These has been mainly adapted (and adopted) from [4, 5, 25, 53].

A.1 General Properties

Wherever ψ is smooth we know that it satisfies the *Eikonal* equation

$$\|\nabla\psi\| = 1 \tag{23}$$

The distance function satisfies this PDE everywhere in the *viscosity sense* [28, 19]. It is also well known that within a sufficiently small neighborhood of $\mathcal{S} = \{\psi = 0\}$, $\psi(\cdot)$ is *smooth*, or at least as smooth as $\mathcal{S}$. These assertions can be made precise through the following Lemma from [23]:

Lemma 4 *Let $\mathcal{S}$ be a C^k ($k \geq 2$) codimension 1 closed hypersurface of $\mathbb{R}^d$. Then, the signed distance function to $\mathcal{S}$ is $C^k(U)$ for a certain neighborhood U of $\mathcal{S}$.*

Differentiating $\|\nabla\psi\|^2 = 1$ we obtain

$$D(\nabla\psi)\,\nabla\psi = 0$$

Therefore,

$$\mathbf{H}_\psi \, \nabla\psi = 0 \tag{24}$$

meaning that the normal to $\mathcal{S}$ at p is an eigenvector of the Hessian, associated to the null eigenvalue. Differentiating again we obtain

$$D^3\psi \, \nabla\psi \; + \; (D^2\psi)^2 = 0 \tag{25}$$

The next Lemma, whose detailed proof can be found in [4], is mainly based in the relations (24) and (25), and it is used to verify that the function $\mu : (-\varepsilon, \varepsilon) \to I\!R^{d\times d}$ defined by $\mu(t) = \mathbf{H}_\psi(p_0 + t\nabla\psi(p_0))$ (p_0 is any point in the manifold $\{\psi = 0\}$) satisfies the following **ODE**:

$$\dot{\mu}(t) + \mu^2(t) = 0 \quad t \in (-\varepsilon, \varepsilon)$$

Lemma 5 *The eigenvectors of $\mathbf{H}_\psi$ are constant along the characteristic lines $x(s) = x_0 + s\nabla\psi(x(s))$ (arc length parametrized, x_0 is a point onto $\mathcal{S}$) of ψ within any neighborhood where it is smooth, and the eigenvalues vary according to*

$$\lambda_i(s) = \frac{\lambda_i(0)}{s\,\lambda_i(0) + 1}$$

We use the above formula to bound the maximum offset $|\epsilon|$ of $\{\psi = 0\}$ that keeps $\{\psi = \epsilon\}$ smooth, we just take $|\epsilon| \, (\max_{1\le i\le d-1} |\lambda_i(0)|) < 1$.

We now obtain bounds on the eigenvalues of the Hessian of the distance function:

Corollary 2 *The eigenvalues $\lambda_i(p)$ of $\mathbf{H}_\psi(p)$ (principal curvatures of $\{x : \psi(x) = \psi(p)\}$) are absolutely bounded by*

$$|\lambda_i(p)| \le \frac{\mathcal{M}_\mathcal{S}}{1 - |\psi(p)|\mathcal{M}_\mathcal{S}}$$

where $\mathcal{M}_\mathcal{S}$ absolutely bounds all eigenvalues of $\mathbf{H}_\psi(p)$, $p \in \mathcal{S}$; and $|\psi(p)|$ is sufficiently small.

To conclude, let's present some concepts on projections onto the implicit surface $\mathcal{S}$, zero level-set of the distance function ψ. It is clear that the projection of a point $p \in I\!R^d$ onto $\mathcal{S}$ is given by

$$\Pi_\mathcal{S}(p) = p - \psi(p)\nabla\psi(p).$$

This projection is well defined as long as there is only one $x \in \mathcal{S}$ such that $\Pi_\mathcal{S}(p) = x$. This can be guaranteed when working within a small tubular neighborhood of a smooth surface $\mathcal{S}$. Moreover, this map is smooth within a certain tubular neighborhood of $\mathcal{S}$ [53]:

Theorem 3 *If $\mathcal{S} \subset I\!R^d$ is a compact C^k ($k \ge 2$) codimension 1 hypersurface, then there is a $h(\mathcal{S}) > 0$ such that the map $\Pi_\mathcal{S}$ is well defined and belongs to $C^{k-1}(\{x : d(x, \mathcal{S}) < h\}, I\!R^d)$.*

B Technical Lemma

Lemma

Let $f : [a,b] \to I\!R$ be a $C^1([a,b])$ function such that f' is Lipschitz. Let $\varphi \in L^\infty([a,b])$ denote (one of) f''s weak derivative. Then one has:

$$\int_a^b f'^2(x)\,dx = f\,f'|_a^b \; - \int_a^b f(x)\varphi(x)\,dx$$

Proof:
Let $ext(f')$ denote the Lipschitz extension of f' to all $I\!R$ given by

$$ext(f')(x) = \begin{cases} f'(a) \; for \; x < a \\ f'(x) \; for \; x \in [a,b] \\ f'(b) \; for \; x > b \end{cases}$$

Then let $ext(f)$ be given by any (bounded) primitive of $ext(f')$, that is $ext(f) = \int ext(f')$. Let $\widehat{\varphi} \in L^\infty I\!R$ denote $ext(f')$'s weak derivative, and we have that $\widehat{\varphi}|_{[a,b]}$ and φ coincide as weak derivatives of f. Let $\{\eta_\epsilon(\cdot)\}_{\{\epsilon>0\}}$ be a family of bounded support mollifiers. Then we define the function

$$f_\epsilon = ext(f) * \eta_\epsilon$$

It is clear that we will have

(a)

$$f_\epsilon \overset{\epsilon\downarrow 0}{\rightrightarrows} ext(f) \; over \; compact \; sets \; of \; I\!R$$

(b)

$$f'_\epsilon \overset{\epsilon\downarrow 0}{\rightrightarrows} ext(f') \; over \; compact \; sets \; of \; I\!R$$

(c)

$$f''_\epsilon \overset{\epsilon\downarrow 0}{\longrightarrow} \widehat{\varphi} \; locally \; in \; L^2(I\!R)$$

Since $f'_\epsilon \in C^\infty(I\!R)$ we can use integration by parts to conclude that:

$$\int_a^b f'^2_\epsilon(x)\,dx \; = \; f'_\epsilon f_\epsilon|_a^b \; - \int_a^b f_\epsilon(x)\,f''_\epsilon(x)\,dx$$

Now the left hand side will converge to $\int_a^b f'^2(x)\,dx$ in view of (b); the first term in the right hand side will obviously converge to $f\,f'|_a^b$. For the remaining term we observe the following ,using Cauchy-Schwartz inequality (let $<,>: L^2([a,b]) \times L^2([a,b]) \to I\!R$ denote $L^2([a,b])$'s internal product):

$$\begin{aligned} & \left| \int_a^b f_\epsilon(x)\,f''_\epsilon(x)\,dx \; - \; \int_a^b f(x)\varphi(x)\,dx \right| \\ = \; & \left| < f''_\epsilon, f_\epsilon > \; - \; < \varphi, f > \right| = \left| < f''_\epsilon, f_\epsilon - f > \; + \; < f, f''_\epsilon - \varphi > \right| \\ \leq \; & (b-a) \left(\left(\max_{\{x\in[a,b]\}} \left| f''_\epsilon(x) \right| \right) \; \| f_\epsilon - f \|_{L^2([a,b])} \; + \; \left(\max_{\{x\in[a,b]\}} |f(x)| \right) \; \| f''_\epsilon - \varphi \|_{L^2([a,b])} \right) \end{aligned}$$

Now, everything is under control since

$$\max_{\{x\in[a,b]\}} \left| f''_\epsilon(x) \right| \leq \|\varphi\|_{L^\infty([a,b])}$$

Hence we have proved

$$\int_a^b f_\epsilon(x)\, f''_\epsilon(x)\, dx \;\xrightarrow{\epsilon\downarrow 0}\; \int_a^b f(x)\varphi(x)\, dx$$

the last step of the proof.

■

References

[1] D. Adalsteinsson and J.A. Sethian, "A fast level set method for propagating interfaces," *J. Comp. Physics* **118**, pp. 269-277, 1995.

[2] R. Alexander & S. Alexander, "Geodesics in Riemannian manifolds with boundary," *Indiana University Mathematics Journal* **30:4**, 1981.

[3] S. Alexander, I.D. Berg, and R.L. Bishop, "The Riemannian obstacle problem," *Illinois Journal of Mathematics* **31:1**, Spring 1987.

[4] L. Ambrosio and N. Dancer, *Calculus of Variations and Partial Differential Equations*, Topics on Geometrical Evolution Problems and Degree Theory, Springer-Verlag, New York, 2000.

[5] L. Ambrosio and H.M. Soner, "Level set approach to mean curvature flow in arbitrary codimension," *J. Diff. Geom.* **43**, pp. 693-737, 1996.

[6] A. Bartesaghi and G. Sapiro, "A system for the generation of curves on 3D brain images," *Human Brain Mapping*, to appear.

[7] T. J. Barth and J. A. Sethian, "Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains," *Journal of Computational Physics* **145**, pp. 1-40, 1998.

[8] M. Bertalmio, L. T. Cheng, S. Osher, and G. Sapiro, "Variational problems and partial differential equations on implicit surfaces: The framework and examples in image processing and pattern formation," *IMA University of Minnesota Preprint*, June 2000.

[9] Blitz++ website: `www.oonumerics.org/blitz`

[10] J. Bloomenthal, *Introduction to Implicit Surfaces*, Morgan Kaufmann Publishers, Inc., San Francisco, California, 1997.

[11] M. Boue and P. Dupuis, "Markov chain approximation for deterministic control problems with affine dynamics and quadratic cost in the control," *SIAM J. Numer. Anal.* **36**, pp. 667-695, 1999.

[12] A. M. Bruckstein, "On shape from shading," *Comp. Vision Graph. Image Processing* **44**, pp. 139-154, 1988.

[13] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert, "Minimal surfaces based object segmentation," *IEEE-PAMI*, **19:4**, pp. 394-398, April 1997.

[14] S. Chen, B. Merriman, S. Osher, and P. Smereka, "A simple level set method for solving Stefan problems," *Journal of Computational Physics* **135**, pp. 8, 1995.

[15] L. T. Cheng, *The Level Set Method Applied to Geometrically Based Motion, Material Science, and Image Processing, PhD Thesis Dissertation, UCLA*, June 2000.

[16] L. T. Cheng, P. Burchard, B. Merriman, and S. Osher, "Motion of curves constrained on surfaces using a level set approach," *UCLA CAM Report* **00-32**, September 2000.

[17] L. D. Cohen, and R. Kimmel, "Global minimum for active contours models: A minimal path approach," *International Journal of Computer Vision* **24**, pp. 57-78, 1997.

[18] M.G. Crandall and P.L. Lions "Two approximations of solutions of Hamilton-Jacobi equations," *Math. of Comp.* **43:167**, pp. 1-19, July 1984.

[19] M.G. Crandall and P.L. Lions, "Viscosity solutions of Hamilton-Jacobi equations," *Transactions of the American Mathematical Society* **277:1**, May 1983.

[20] E. Dijkstra, "A note on two problems in connection with graphs," *Numerische Math.* **1**, pp. 269-271, 1959.

[21] M. Desbrun, M. Meyer, P. Scröeder, and A. Barr, "Discrete differential-geometry operators in nD," *Caltech, USC Report*, July 22, 2000.

[22] S. Fomel, "A variational formulation of the fast marching Eikonal solver," *Stanford Exploration Project*, Report SERGEY, May 1, pp. 357-376, May 2000.

[23] R.L. Foote, "Regularity of the distance function," *Proceedings of American Mathematical Society* **92:1**, September 1984.

[24] S. F. Frisken, R. N. Perry, A. Rockwood, and T. Jones, "Adaptively sampled fields: A general representation of shape for computer graphics," *Computer Graphics (SIGGRAPH)*, New Orleans, July 2000.

[25] J. Gomes and O. Faugeras, "Representing and evolving smooth manifolds of arbitrary codimension embedded in $\mathbb{R}^n$ as the intersection of n hypersurfaces: The vector distance functions," *RR-4012* INRIA, Oct. 2000.

[26] J. Helmsen, E. G. Puckett, P. Collela, and M. Dorr, "Two new methods for simulating photolithography development in 3D," *Proc. SPIE Microlithography* **IX**, pp. 253, 1996.

[27] P. Hoch and M. Rascle "Hamilton-Jacobi equations on a manifold and applications to grid generation or refinement," `http://www-math.unice.fr/~hoch/psfiles/hamja.ps`

[28] H. Ishii, "A simple direct proof of uniqueness for solutions of the Hamilton-Jacobi equations of Eikonal type," *Proc. Amer. Math. Soc.* **100:2**, pp. 247-151, 1987.

[29] N. Khaneja, M.I. Miller, and U. Grenander, "Dynamic programming generation of geodesics and sulci on brain surfaces," *IEEE Trans. on Pattern Analysis and Machine Intelligence* **20:11**, pp.1260-1265, November, 1998.

[30] R. Kimmel, "Numerical geometry of images: Theory, algorithms, and applications," *Technion CIS Report* **9910**, October 1999.

[31] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," *Proc. National Academy of Sciences* **95:15**, pp. 8431-8435, 1998.

[32] N. Kiryati and G. Székely, "Estimating shortest paths and minimal distances on digitized three dimensional surfaces," *Pattern Recognition* **26**, pp. 1623–1637, 1993.

[33] V. Krishnamurthy and M. Levoy, "Fitting smooth surfaces to dense polygon meshes," *Computer Graphics*, pp. 313-324, 1996.

[34] J. C. Latombe, *Robot Motion Planning*, Kluwer Academic Publishers, Boston, MA, 1991.

[35] F. Lafon and S. Osher, "High order two dimensional nonoscillatory methods for solving Hamilton-Jacobi scalar equations," *Journal of Computational Physics* **123**, pp. 235-253, 1996.

[36] J. M. Lee, *Riemannian Manifolds, An Introduction to Curvature,* Springer-Verlag, New York, Inc., 1997.

[37] C. Mantegazza and A.C. Menucci, "Hamilton-Jacobi equations and distance functions on Riemannian manifolds," `http://cvgmt.sns.it/papers/manmen99/`.

[38] A. Marino and D. Scolozzi, "Geodetiche con ostacolo," *Boll. Un. Mat. Ital.* **6:2-B**, pp. 1-31, 1983.

[39] S. Mauch, "Closest point transform,"
`www.ama.caltech.edu/~ seanm/software/cpt/cpt.html`.

[40] F. Memoli, G. Sapiro, and S. Osher, "Harmonic maps onto implicit manifolds," pre-print, March 2001.

[41] J. S. B. Mitchell, "An algorithmic approach to some problems in terrain navigation," *Artificial Intelligence* **37**, pp. 171-201, 1988.

[42] J. S. B. Mitchell, D. Payton, and D. Keirsey, "Planning and reasoning for autonomous vehicle control," *International Journal of Intelligent Systems* **2**, pp. 129-198, 1987.

[43] S. Osher, "A level-set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations," *SIMA J. Numer. Anal.* **24**, pp. 1145, 1993.

[44] S. J. Osher and R. P. Fedkiw, "Level set methods," *ULCA CAM Report* **00-07**, February 2000.

[45] S. J. Osher and J. A. Sethian, "Fronts propagation with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations," *Journal of Computational Physics* **79**, pp. 12-49, 1988.

[46] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, "A PDE-based fast local level set method," *Journal of Computational Physics* **155**, pp. 410-438, 1999.

[47] F. P. Preparata and M. I. Shamos, *Computational Geometry*, Texts and Monographs in Computer Science, Springer-Verlag, New York, 1990.

[48] E. Rouy and A. Tourin, "A viscosity solutions approach to shape-from-shading," *SIAM. J. Numer. Analy.* **29:3**, pp. 867-884, 1992.

[49] T. Sakai, *Riemannian Geometry*, AMS Translations of Mathematical Monographs **149**.

[50] J. Sethian, "Fast marching level set methods for three-dimensional photolithography development," *Proc. SPIE International Symposium on Microlithography*, Santa Clara, California, March, 1996.

[51] J. A. Sethian, "A fast marching level-set method for monotonically advancing fronts," *Proc. Nat. Acad. Sci.* **93:4**, pp. 1591-1595, 1996.

[52] J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Sciences,* Cambridge University Press, Cambridge-UK, 1996.

[53] L. Simon, *Theorems on Regularity and Singularity of Energy Minimizing Maps*, Birkhäuser, Boston, 1996.

[54] G. Taubin, "Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation," *IEEE Trans. PAMI* **13:11**, pp. 1115-1138, 1991.

[55] A. W. Toga, *Brain Warping*, Academic Press, New York, 1998.

[56] P. Thompson, R. Woods, M. Mega, and A. Toga, "Mathematical/computational challenges in creating deformable and probabilistic atlases of the human brain," *Human Brain Mapping* **9:2**, pp. 81-92, 2000.

[57] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *IEEE Transactions on Automatic Control* **40** pp. 1528-1538, 1995.

[58] D. C. Van Essen, H. Drury, S. Joshi and M. I. Miller, "Functional and structural mapping of human cerebral cortex: Solutions are in the surfaces," *Proceedings of the National Academy of Science* **95**, pp. 788-795, February 1998.

[59] B. Wandell, S. Chial, and B. Backus, "Cortical visualization," *Journal of Cognitive Neuroscience*, to appear.

[60] A. Witkin and P. Heckbert, "Using particles to sample and control implicit surfaces," *Computer Graphics (SIGGRAPH)*, pp. 269-278, 1994.

[61] F.E. Wolter, "Cut loci in bordered and unbordered Riemannian manifolds," *Doctoral Dissertation, Technische Universität Berlin*, 1985.

[62] Z. Wood, M. Desburn, P. Schröder, and D. Breen, "Semi-Regular mesh extraction from volume," *Proc. IEEE Visualization*, 2000.

[63] G. Yngve and G. Turk, "Creating smooth implicit surfaces from polygonal meshes," *Technical Report GIT-GVU-99-42, Graphics, Visualization, and Usability Center. Georgia Institute of Technology*, 1999 (obtained from `www.cc.gatech.edu/gvu/geometry/publications.html`).

[64] A. Yezzi, S. Kichenassamy, P. Olver, and A. Tannenbaum, "Geometric active contours for segmentation of medical imagery," *IEEE Trans. Medical Imaging* **16**, pp. 199-210, 1997.

[65] H. Zhao, S. Osher, B. Merriman, and M. Kang, "Implicit, non-parametric shape reconstruction from unorganized points using a variational level set method," *Comp. Vision and Image Understanding* **80**, pp. 295-314, 2000.

[66] G. Zigelman, R. Kimmel, and N. Kiryati, "Texture mapping using surface flattening via multi-dimensional scaling," *Technion-CIS Technical Report* **2000-01**, 2000.

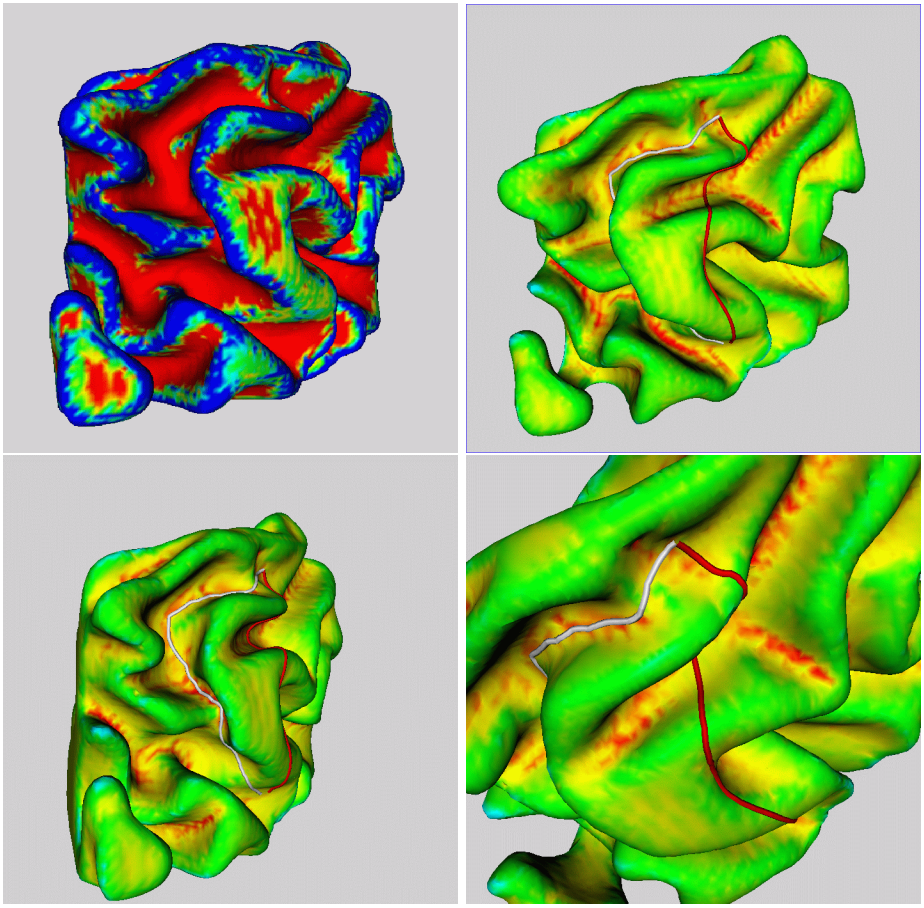

Figure 10: These four figures show the detection of valleys over implicit surfaces representing a portion of the human cortex. We use a mean curvature based weighted distance. In the left-upper corner we show the mean curvature of the brain surface (clipped to improve visualization). It is quite convincing that this quantity can be of of great help to detect valleys. In the remaining figures we show two curves over the surface, whose coloring correspond to the mean curvature (not clipped, from red, yellow, green to blue, as the value increases). The red curve is the one that corresponds to the *natural* geodesic ($g = 1$), while the white curve is the weighted-geodesic that should travel through "nether" regions. Indeed, a very clear difference exists between both trajectories, since the white curve makes its way through regions where the mean curvature attains low values. The figure in the right-low quadrant is a zoomed view of the same situation.

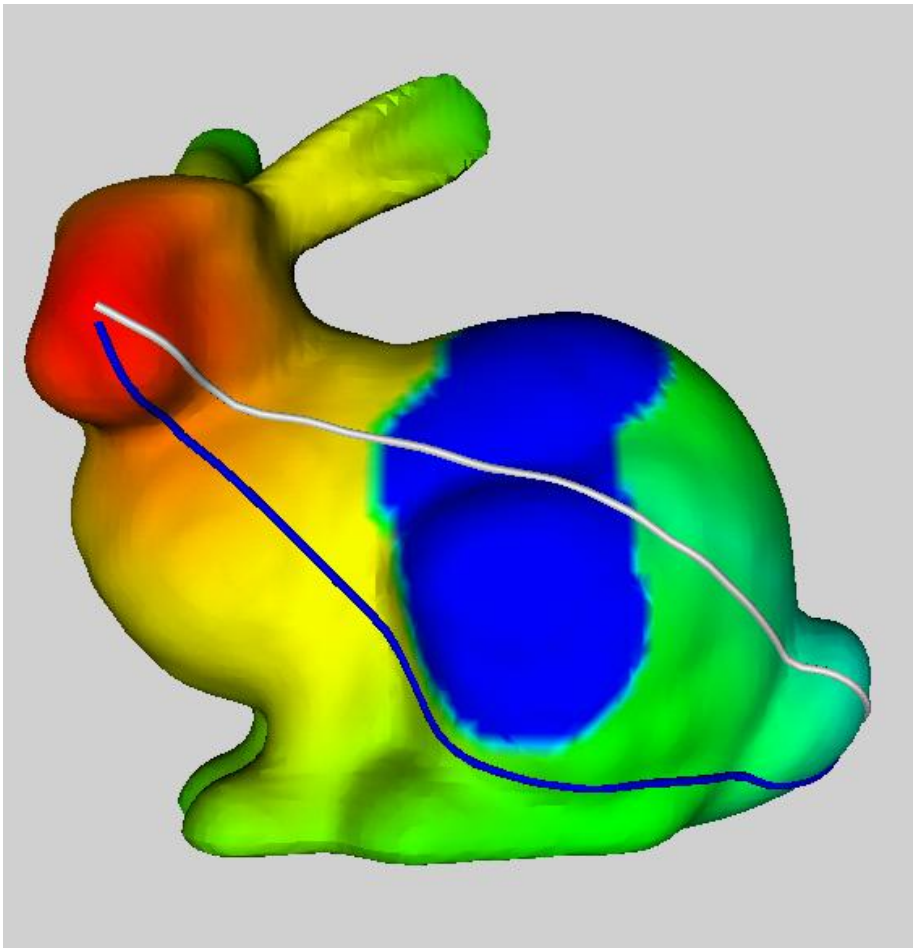

Figure 11: Distance map and geodesic curve between two points on an implicit bunny surface with an intrinsic obstacle on it. We now use a binary weight, $g = \{1, \infty\}$, being infinity at the obstacle. This permits, as illustrated in the figure, the computation of optimal paths with obstacles on implicit surfaces. The blue path corresponds to the obstacle-weighted distance function, and the white one to the natural ($g = 1$) distance function. Both geodesics are shown over the surface of the bunny, the pseudocolor representing the weighted distance for the surface with obstacle. The obstacle is also shown in blue.