UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a doctoral Dissertation by

Roberto Facundo Mémoli Techera

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committe have been made.

____

Name of Faculty Adviser

____

Signature of Faculty Adviser

____

Date

GRADUATE SCHOOL

ESTIMATION OF DISTANCE FUNCTIONS AND GEODESICS AND ITS USE FOR SHAPE COMPARISON AND ALIGNMENT: THEORETICAL AND COMPUTATIONAL RESULTS

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

ROBERTO FACUNDO MÉMOLI TECHERA

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

GUILLERMO SAPIRO, ADVISER

MAY 2005

Acknowledgments

I would like to express my gratitude to my adviser Guillermo Sapiro for his guidance, encouragement and support. Thanks to Beto, Kedar, Diego, Liron, Anish, Omar, Marcelo, Pierre, Alejandro and Carsten for intersting conversations, not only research. Also to Bonnie, Dalia and Guillermo for taking care of me. To my teachers at the U of MN, I learned a lot from S. Bobkov, R. Gulliver, C. Leung and O. Zeitouni. Gregory Randall provided an environment where a lot of people from my generation were able to start doing research in Uruguay, my thanks to him. Many thanks to IPAM, CSIC-Uruguay and IIE-UdelaR.

A Blanca, la Nona y Cacha.

Abstract

In this work we have tried to solve several problems from Shape Theory. We have dealt with the estimation of certain intrinsic quantities defined on submanifolds of Euclidean space, as well as with the use of this information to perform shape comparison and alignment. We have also approached the problem of surface warping and smoothing of information defined on implicit surfaces. Examples are shown and future lines of research are suggested to both improving the performance and extending reach of the techniques here introduced.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The work presented in this thesis is comprised of several pieces. The different problems approached are of practical nature and as a result there are some data that must be processed. The problems dealt with involve considering a certain *geometrical structure* which, in its most basic form, is a surface in $I\!R^3$. We think of this strucutre as of the *underlying* entity. These structures are assumed to be given to us in either of two ways: Implicitely or as Point Clouds. The former means that the surface is the zero level set of a certain function $\phi : I\!R^3 \rightarrow I\!R$, and the latter means that the input we get is just a finite set of points sampled from the surface (or around the surface) according to a certain probability law.

Some of the problems are about trying to extract certain kinds of information about the underlying surfaces from the discrete data. Some examples are dimensionality, topology, curvatures and geodesic distances and shortest paths bewteen any two points. In this thesis we have considered the extraction of what we believe is a fundamental kind of information: metric information, that is, geodesic distances and shortest paths. In some situations one may be interested in comparing two objects based on certain metric characteristics of each of them. When we compare, we most often need to the put things we are comparing in correspondence with one another. In this thesis we have also considered these two problems. We can say therefore that the problems here dealt with are of metric nature. Roughly speaking, we could say that we first estimate the metrics of the underlying structures, use this information to put them into correspondence and then compare them.

Part I first briefly discusses work done during my Masters which revolved around trying to compute intrinsic distance functions and geodesics on surfaces represented implicitly. This was extended during my PhD to deal with (1) submanifolds of any Euclidean space given either implicitely or (2) as a Point Clouds. In Chapter 4 the results of these extensions are presented. When dealing with point cloud data it turns out that in order to prove (probabilisitic) convergence of the estimated quantity to the real one it suffices to guarantee that certain random coverings defined by these point cloud data converge, in an appropiate sense, to the underlying structure. This is explored in Chapter 3 and then used in Chapter 4.

As an application of the framework here proposed for intrinsic distance computation, my advisor suggested considering the problem of comparing two surfaces given as point clouds in a manner invariant to general isometries. In more detail, the idea was to consider the matrices with elements corresponding to the intrinsic distance between pairs of points on each point cloud and then compare them in some way so as to obtain a measure of similarity. The search for an appropiate framework for this led to the work presented in Chapter 5.

Part II presents work dealing mainly with applications to Brain Imaging. This is also work on a problem proposed to me by my advisor a long time ago. The idea is that among the different tasks that must be performed by researchers who study the behaviour of the brain across populations, we encounter the problem of having to map different brain surfaces into a common, standard, brain surface. This is known as Brain Warping in the specialized literature. This is useful when a study has been performed on a number of subjects and these need to be compared. Imagine that as the consequence of the study we obtain the surface of the brain colored in a certain way, perhaps representing activity of the corresponding cortical area. Then, in order to simplify comparative studies, it seems reasonable to map all the brains to a fixed shape, for example a sphere or the standard brain alluded to above, and then color the target surface with the coloring inherited through the mapping (from the coloring in the domain). It is necessary to require that these mappings achieve low distorsion in some convenient sense. Also, it turns out that there are structures on the brain surface that are stable across subjects, and should be used to specify restrictions to search for the low distorsion mappings. As an initial attempt at the solution I worked on the object of Chapter 6 which proposes a more or less general framework for solving variational problems on implicit surfaces. In this case, we assume that we have data defined on the surface which takes values on another manifold also represented implicitly. This has clear connections with problem of Brain Warping mentioned above. We obtain a set of equations that while defined on the whole Euclidean space, are intrinsic to the implicitly defined target manifold and map into it. This permits the use of classical numerical techniques in Cartesian grids, regardless of the geometry of the target manifold. The extension to open domain surfaces and target submanifolds is addressed as well. In the latter case, the submanifold is defined as the intersection of two higher dimensional hypersurfaces, and all the computations are restricted to this intersection. Examples of the applications of the framework described in Chapter 6 include harmonic maps in liquid crystals, where the target manifold is a hypersphere; probability maps, where the target manifold is a hyperplane; chroma enhancement; texture mapping; and general geometric mapping between high dimensional manifolds.

Later in time, I returned to the problem and Chapters 7 and 8 are the results of that effort. Chapter 7 discusses a couple different applications of both the geodesic distance computation algorithms and the Harmonic Maps framework presented in previous chapters.

In Chapter 8 we return to the problem of Brain Warping with new tools. For this new approach, Minimizing Lipschitz Extensions are considered and the representation of the surfaces is based on Point Clouds. The basic concept is to compute a map between surfaces that minimizes a distortion measure based on geodesic distances while respecting the boundary conditions provided. In particular, the global Lipschitz constant of the map is minimized. This framework allows generic boundary conditions to be applied and allows direct surface-to-surface warping. It avoids the need for intermediate maps that flatten the surface onto the plane or sphere, as is commonly done in the literature on surface-based non-rigid brain image registration. The presentation of the framework is complemented with examples on synthetic geometric phantoms and cortical surfaces extracted from human brain MRI scans.

Chapter 9 is the concluding chapter where we have tried to summarize the main concepts introduced in this work and discuss possible extensions.

1.1 Data Representation

Next, we discuss the two types of data representations we have worked with and their relation to the different chapters in the thesis.

1.1.1 Implicit Surfaces

The implicit representation of surfaces, here introduced for solving variational problems and PDEs on surfaces, is inspired in part by the level-set work of Osher and Sethian [159]. This work, and those that followed it, showed the importance of representing *deforming* surfaces as level-sets of functions with higher dimensional domains, obtaining more robust and accurate numerical algorithms (and topological freedom). Note that, in contrast with the level-set approach of Osher and Sethian, our manifolds are fixed.

Solving PDEs and variational problems with polynomial meshes involves the non-trivial discretization of the equations in general polygonal grids, as well as the difficult numerical computation of other quantities like projections onto the discretized surface (when computing gradients and Laplacians for example). Although the use of triangulated surfaces is quite popular, there is still no consensus on how to compute simple differential characteristics such as tangents, normals, principal directions, and curvatures. On the other hand, it is commonly accepted that computing these objects for iso-surfaces (implicit representations) is simpler and more accurate and robust. This problem becomes even more significant when not only do we have to compute these first and second order differential characteristics of the surface, but also have to use them to solve variational problems and PDEs for data defined on the surface. Very little work has been done on the formal analysis of finite difference schemes on non-Cartesian meshes.[1] Note also that working with polygonal representations is dimensionality dependent, and solving these equations for high dimensional (> 2) surfaces becomes even more challenging and significantly less studied. In Chapter 2 we propose a solution for the problem of estimating weighted intrinsic functions on implicit surfaces. We assume that the implicitating function is in fact the distance to the surface. In Chapter 6 we propose a framework for solving certain variational problems on implicit surfaces via their associated time dependent PDEs, i.e. trough a *gradient descent* procedure.

The works developed in chapters 2 and 6 are valid for hypersurfaces. In Chapter 6 we also show how to extend the initial framework there introduced to Submanifolds by representing this structure as the intersection of implicit hypersurfaces. Also, in the first sections of Chapter 4 we extend the ideas presented in Chapter 2 to any submanifold of euclidean space.

Note that the computational cost of working with implicit representations is not higher than with meshes, since all the work is performed in a narrow band around the level-set(s) of interest.

The framework of implicit representations enables us to perform all the computations on the Cartesian grid corresponding to the embedding function. These computations are, nevertheless, intrinsic to the surface. In Appendix B.3 we provide a *dictionary* that explains how to compute and intrinsic differential operation

[1] Very important work has been done for finite element approaches, e.g., by the group of Prof. M. Rumpf; as well as for particular equations on particular sub-division representations [14].

on a function f by using extrinsic differentials of both f and the implicitating function. Advantages of using Cartesian grid instead of a triangulated mesh include the availability of well studied numerical techniques with accurate error measures and the topological flexibility of the surface, all leading to simple, accurate, robust and elegant implementations.

Numerical schemes that solve gradient descent flows and PDEs onto generic target manifolds $\mathcal{N}$ (and spheres or surfaces in particular) will, in general, move the points outside of $\mathcal{N}$ due to numerical errors. The points will then need to be projected back,[2] see for example [4, 171] for the case of $\mathcal{N}$ being a sphere (where the projection is trivial, just a normalization). For general target manifolds, this projection means that for every point $p \in I\!R^d$ ($\mathcal{N} \subset I\!R^d$) we need to know the closest point to p in $\mathcal{N}$. This means knowing the distance from every point $p \in I\!R^d$ to $\mathcal{N}$ (or at least all points in a band of $\mathcal{N}$). This is nothing else than an implicit representation of the target $\mathcal{N}$, being the particular embedding in this case a distance function. This presents additional background for the framework here introduced, that is, if the embedding function for the surface has to be computed anyway for the projection, why not use it from the beginning if it helps in other steps in the computation?

In a number of applications, surfaces are already given in implicit form, e.g., [33], therefore, the framework introduced in Chapter 6 is not only simple and robust, but it is also natural in those applications. Moreover, in the state-of-the-art and most commonly used packages to obtain 3D models from range data, the algorithms output an implicit (distance) function (see for example [170]). Therefore, it is very important, if nothing else for completeness, to have the computational framework here developed, so that the surface representation is dictated by the data and the application and not the other way around. On the other hand, not all surfaces (manifolds) are originally represented in implicit form. When the target manifold $\mathcal{N}$ is simple, like hyper-spheres in the case of liquid crystals, the embedding process is trivial. For generic surfaces, we need to apply an algorithm that transforms the given explicit representation into an implicit one. Although this is still a very active area of research, many very good algorithms have been developed, e.g., [63, 174, 122, 215]. Note that this translation needs to be done only once for any surface.

1.1.2 Point Clouds

Point clouds are one of the most primitive and fundamental shape representations. One of the most popular sources of point clouds are 3D shape acquisition devices, such as laser range scanners, with applications in geoscience, art (e.g., archival), medicine (e.g., prothestetics), manufacturing (from cars to clothes), and security (e.g., recognition), among other disciplines. These scanners provide in general raw data in the form of (noisy) unorganized point clouds representing surface samples. With the increasing popularity and very broad applications of this source of data, it is natural and important to work directly with this representation, without having to go to the intermediate step of fitting a surface to it (step that can add computational complexity and introduce errors). See for example [20, 57, 61, 88, 133, 162, 163] for a few of the recent works with this type of data. Note that point clouds can also be used as primitives for visualization, e.g., [23, 88, 173], as well as for editing [218].

Another important field where point clouds are found is in the representation of high-dimensional man-

[2] For particular flat target manifolds as the whole space $I\!R^d$ or as those in [161], the projection is not needed. Other authors, e.g., [35, 118], have avoided the projection step for particular cases, while in [201] the authors modify the given variational formulation, in some restricted cases, to include the projection step.

ifolds by samples. This type of high-dimensional and general co-dimension data appears in almost all disciplines, from computational biology to image analysis to financial data. Due to the extremely high dimensions, it is impossible to perform manifold reconstruction, and the work needs to be done directly on the raw data, meaning the point cloud.

A variety of objects/shapes are then naturally represented as point clouds in $I\!R^d$. It is thereby important to be able to derive basic properties of the shape, such as topolgy, geodesic distances and curvatures, directly from this representation. Also, one is often presented with the fundamental problem of having to decide whether two of those point clouds, and their corresponding underlying objects or manifolds, represent the same geometric structure or not (*object recognition and classification*). We are then concerned with questions about the underlying unknown structures (objects), which need to be answered based on discrete and finite measures taken between their respective point clouds. In greater generality, we wonder what is the structural information we can gather about the object itself by exploring the point cloud by which the object is represented. Examples include intrinsic distances, curvatures, normals (see [149]), dimension (see [51]), spectrum of differential operators such as the intrinsic Laplacian (see [125] and references therein), and topological invariants (see [47, 31, 155]).

Chapter 3 is devoted to setting some basic modelling assumptions and to presenting some basic results which will be used in later sections. These results comprise mostly bounds on the *probability of coverage* of a submanifold of $I\!R^d$ by Euclidean balls whose centers are distributed on (or around) the submanifold according to a certain probability measure. This probability measure, for example in the case of shapes acquired by a 3D scanner, models the acquisition process itself.

Chapter 4 addresses one of the most fundamental operations in the study and processing of sub-manifolds of Euclidean space: The computation of intrinsic distance functions and geodesics. We show that this can be done by working directly with the point cloud, without the need for reconstructing the underlying manifold. The results are valid for general dimensions and co-dimensions, and for manifolds with or without boundary. These results include the analysis of noisy point clouds obtained from sampling the manifold and are based on the ideas presented in Chapter 2.

In Chapter 5, a geometric framework for comparing manifolds given by point clouds is presented. The underlying theory is based on Gromov-Hausdorff distances, leading to isometry invariant and completely geometric comparisons. This theory is embedded in a probabilistic setting as derived from random sampling of manifolds, and then combined with results on matrices of pairwise geodesic distances to lead to a computational implementation of the framework. The theoretical and computational results here presented are complemented with experiments for three dimensional shapes.

1.2 Collaborations

The work presented in this thesis is mainly joint with my advisor, Prof. Guillermo Sapiro. Below is a detailed list of co-autorships for the work presented in the different core chapters of this thesis.

Chapter 2

Joint with G. Sapiro.

Chapter 4

Joint with G. Sapiro.

Chapter 5

Joint with G. Sapiro.

Chapter 6

Joint with G. Sapiro and S. Osher.

Chapter 7

Joint with G. Sapiro and P. Thompson.

Chapter 8

Joint with G. Sapiro and P. Thompson.

Part I

Chapter 2

Distance Functions on Implicit Manifolds

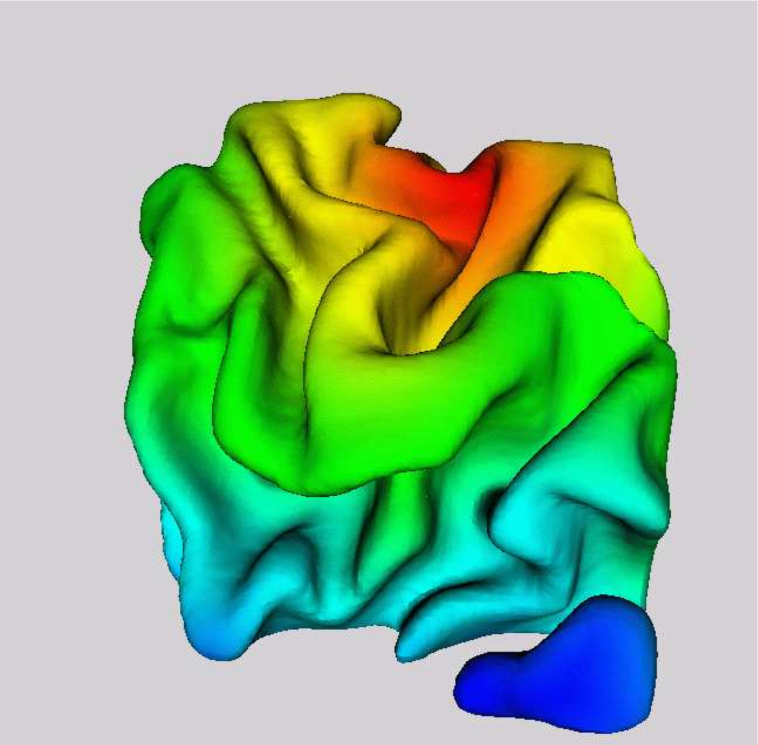


Figure 2.1: Distance map from a point on a portion of white/gray matter boundary of the cortex.

2.1 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 [45]). Geodesics are used for example for path planning in robotics [129], brain flattening and brain warping in computational neuroscience [195, 194, 200, 203, 217], crests, valleys, and silhouettes computations in computer graphics and brain research [16, 115, 208], mesh generation [210], and many applications in mathematical physics. Distance functions are also very important in optimal control [199] and computational geometry for computations such as Voronoi diagrams and skeletons [169]. 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 chapter to present a computationally optimal technique for the computation of intrinsic weighted distance functions on implicit hyper-surfaces. It is well-known already, and it will be further detailed below, that these weighted distances can be obtained as the solution of simple Hamilton-Jacobi equations.

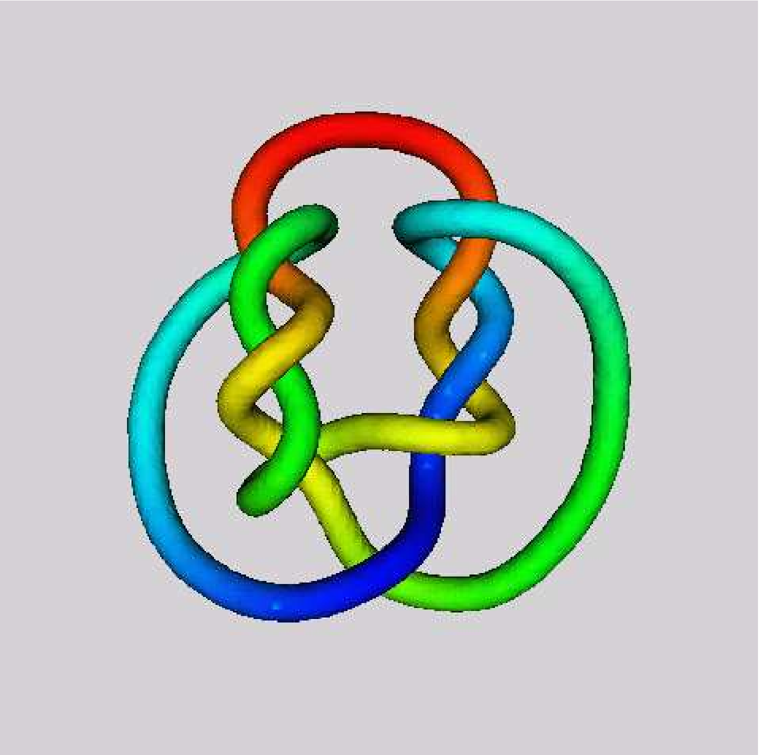


Figure 2.2: Distance map from one seed point on a knot. In this picture we evidence that the algorithm works well for quite convoluted geometries (as long as h is properly chosen). Note how points close in the Euclidean sense but far away in the intrinsic sense receive very different colors, indicating their large (intrinsic) distance.

2.1.1 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 $I\!R^d$ defined as the zero level set of a function $\psi : I\!R^d \to I\!R$. That is, $\mathcal{S}$ is given by $\{x \in I\!R^d : \psi(x) = 0\}$. We assume from now on that ψ is a signed distance function to the surface $\mathcal{S}$. (This is 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 $d_{\mathcal{S}}^g(p,x)$ for all desired points $x \in \mathcal{S}$.[1] 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 $(d-1)\times(d-1)$ identity matrix) and not on the embedding Euclidean space. For a given positive 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

$$d_{\mathcal{S}}^g(p,x) \stackrel{\triangle}{=} \inf_{\mathcal{C}_{px}[\mathcal{S}]} \{\mathbf{L}_g(\mathcal{C})\} \tag{2.1}$$

where

$$\mathbf{L}_g\{\mathcal{C}\} \stackrel{\triangle}{=} \int_a^b g\left(\mathcal{C}(l)\right) \, \|\dot{\mathcal{C}}(l)\| \, dl \tag{2.2}$$

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 [58]. 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, with the vertices as the graph nodes and the edges the connections between them (see [120]). The problem is that the optimal paths computed by this algorithm are limited to travel on the graph edges, giving only a first approximation of the true distance. Moreover, Dijkstra's algorithm is not a consistent one: it will not converge to the true desired distance when the graph and grid is refined [147, 148]. The solution to this problem, limited to Cartesian grids, was developed in [99, 178, 179, 199] (and recently extended by Osher and Helmsen, see [158]). 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 [172, 199], while keeping the same optimal complexity of $O(n \log n)$. This work was later extended in [117] for triangulated surfaces (see also [16, 124] for related works on numerics on non-Cartesian grids). We should note that the algorithm developed in [117] 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 [119]. Following [179], we call these *fast marching algorithms*.

The basic idea behind the computationally optimal techniques for finding weighted distances, fast march-

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

ing algorithms, is to note that the distance function satisfies a Hamilton-Jacobi Partial Differential Equation (PDE) in the viscosity sense; see for example [138, 175] for the general topic of distance functions on Riemannian manifolds (and a nice mathematical treatment), and [27, 75, 119, 160, 159, 179] for the planar (and more intuitive) case. This Hamilton-Jacobi equation is given by

$$\|\nabla_{\mathcal{S}} d^g_{\mathcal{S}}\| = g \tag{2.3}$$

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.[2]

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 us 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 [15, 117]). The goal of this chapter is to extend this to implicit hyper-surfaces.

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

2.1.2 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., [19, 33, 40, 174, 158, 159, 180, 214, 208], 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 [117]. 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 [135], 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

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

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 pre-processing. This will be explained and discussed later in the chapter (§2.5). Moreover, we will later 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 §2.5. 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 [40]. 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 equation. Therefore, this approach is not computationally optimal for the class of Hamilton-Jacobi equations discussed in this chapter (although when properly implemented the computational complexity of this iterative scheme is the same as in the fast marching method here proposed, the inner loop is more complex, making the iterative algorithm slower).[3] Second, very careful discretization must be done to the equation proposed in [40] due to the presence of intrinsic jump functions that might change the zero level-set (i.e., the surface). On the other hand, the numerical implementation is not necessarily 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 $\Theta(\sqrt{\Delta x})$ [53]), 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 [99, 158, 178, 179, 199], 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 equation* using the *fast marching* technique. The first step towards our goal is to express all the quantities in the intrinsic Eikonal equation by its *implicit-extended* representations. What we mean is that the *intrinsic* problem (we consider $g = 1$ for simplicity of exposition)

$$\begin{cases} \|\nabla_{\mathcal{S}}\, d_{\mathcal{S}}(x)\| = 1 \ \text{ for } p \,\in\, \mathcal{S} \\ d_{\mathcal{S}}(p) = 0. \end{cases} \tag{2.4}$$

with $p \in \mathcal{S}$ the seed point, is to be extended to all $I\!R^d$ (or at least to a band surrounding $\mathcal{S}$), and the derivatives

[3] The general framework introduced in [40] is applicable beyond the Hamilton-Jacobi equations discussed in this chapter (see also [134, 39]). Here we limit the comparison between the techniques to the equations were both approaches are applicable.

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

$$\begin{cases} \|\nabla \hat{d}(x)\|^2 - |\nabla \hat{d}(x) \cdot \nabla \psi(x)|^2 = 1 \text{ for } x \in I\!R^d \\ \hat{d}(l(p)) = 0. \end{cases} \tag{2.5}$$

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

This is exactly the approach introduced in [40], as discussed above, to build-up the evolutionary approach, given by the following PDE:

$$\phi_t + sgn(\phi_0)\left(\sqrt{\|\nabla\phi\|^2 - |\nabla\phi \cdot \nabla\psi|^2} - 1\right) = 0 \tag{2.6}$$

where $\phi_0(x) = \phi(x, 0)$ is the initial value of the evolving function, generally a step-like function (convolved with the signum) 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 (2.5), and not its iterative counterpart (2.6). It turns out that the basic requirements for the construction of a fast marching method, even with the recent extensions in [158], 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.[4]

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 (2.5)), 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 chapter to present a third approach that addresses all these problems.

2.1.3 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 $I\!R^3$ for which $|\psi(x)| \leq h$. This gives a region in $I\!R^d$ 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 [99, 178, 179, 199], in a Cartesian grid. We then use this Euclidean distance function as an approximation of the intrinsic distance on $\mathcal{S}$. In §2.2 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 [99, 178, 179, 199].[5] Therefore, when adapting these algorithms to work on Euclidean spaces with boundary, adaptation described in §2.3, we obtain an algorithm for the computation of intrinsic distances on implicit surfaces with the same simplicity,

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

[5] In contrast with works such as [1, 165], 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.

computational complexity, and accuracy as the optimal fast marching techniques for computing Euclidean distances on Cartesian grids.[6] In §2.3 we also explicitly discuss the numerical error of our proposed technique. Examples of the algorithm here proposed are given in Chapters 4 and 7. 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 §2.5. This section also discusses the use of the framework here presented for non-implicit surfaces. Finally, some concluding remarks are given in §2.6.

2.2 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 $I\!R^d$ represented as the zero level-set of a distance function $\psi : I\!R^d \to I\!R$. 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

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

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 Chapter 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 ($d_{\mathcal{S}}^g(p)$ is the intrinsic g-weighted distance on $\mathcal{S}$).

$$\begin{cases} \|\nabla_{\mathcal{S}}\, d_{\mathcal{S}}^g(p)\| = g \;\text{ for } p \,\in\, \mathcal{S} \\ d_{\mathcal{S}}^g(q) = 0. \end{cases} \tag{2.7}$$

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

$$\begin{cases} \|\nabla d_{\Omega_h}^{\tilde{g}}(p)\| = \tilde{g} \;\text{ for } p \,\in\, \Omega_h \\ d_{\Omega_h}^{\tilde{g}}(q) = 0. \end{cases} \tag{2.8}$$

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

[6] Although in this chapter we deal with the fast marching techniques, other techniques for computing distance functions on Cartesian grids, e.g., the fast technique reported in [24] 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.

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.

2.2.1 The Extension of the Weight g

We require that $\tilde{g}|_{\mathcal{S}} = g$, and that $\tilde{g}$ is *smooth* and non-negative within Ω_h. There are situations when one has a readily available extension, and others 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 [130]). The *natural* extension then becomes

$$\tilde{g}(x) = F(\Lambda(\mathbf{H}_{\psi|_{T_x\mathcal{S}(x)}}(x))), \;\; x \in \Omega_h \tag{2.9}$$

where $\mathcal{S}(x) \overset{\triangle}{=} \{y \in I\!\!R^d : \;\psi(y) = \psi(x)\}$

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 Chapter 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.:

$$\tilde{g}(x) = g(\Pi_{\mathcal{S}}(x)), \;\; x \in \Omega_h \tag{2.10}$$

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 desired extension Ω. Taking h sufficiently small we can guarantee that $\Omega \supset \Omega_h$ if $\mathcal{S}$ is smooth. See Chapter A for some details.
In practice this extension can be accomplished solving the equation [38]

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

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

2.2.2 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, 209]. 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 $\tilde{g}^2 \mathbb{I}$, where $\mathbb{I}$ now stands for the $d \times d$ identity matrix).

Definition 1 *Let $p, q \in \mathcal{M}$, then if $d_{\mathcal{M}}(\cdot,\cdot) : \mathcal{M} \times \mathcal{M} \to \mathbb{R}$ 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 exists a *shortest path* joining them, see [2]. The following results deal 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 $\mathbb{R}^d$ 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.

It might seem a bit awkward that one cannot achieve a 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 $\mathbb{R}^2$ with the open unit disc removed, see Figure 2.3, and its Euclidean metric. Let γ be one arc-length parameterized shortest path joining A and B. Then, the acceleration $\ddot{\gamma}$ is $\vec{0}$ in all the open segment (AP) , 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*.

Remark 1 *For the general situation, in [3, 139] the authors proved that shortest paths do have* Lipschitz continuous *first derivatives, which 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, 139, 209] and references therein.

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 $d_{\Omega_h} \stackrel{\triangle}{=} d^1_{\Omega_h}$.

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

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

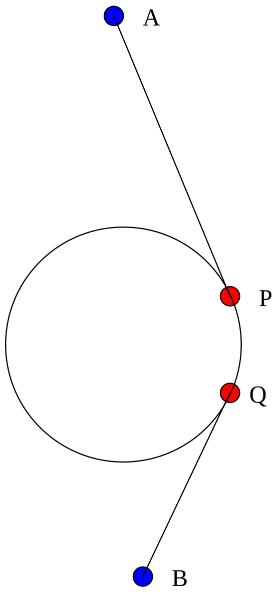


Figure 2.3: The minimal path is C^1, but not C^2.

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

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

Remark 3 *Since we are assuming $\tilde{g}$ to be a* smooth *extension of g to all Ω_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 $M_g \overset{\triangle}{=} \max_{\{x\in\mathcal{S}\}} g(x)$ and $M_{\tilde{g}} \overset{\triangle}{=} \sup_{\{x\in\Omega\}} \tilde{g}(x)$.*

We need the following *Lemma* whose simple proof we omit (see for example [45]).

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

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

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

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*

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

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 [172]. 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 *hyper-surface $\mathcal{S}$. Let $d_h^{\tilde{g}} = d_{\Omega_h}^{\tilde{g}}(A,B)$ and $d_{\mathcal{S}}^g = d_{\mathcal{S}}^g(A,B)$. Then, for points on the surface $\mathcal{S}$, we have that for sufficiently small h*

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

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).

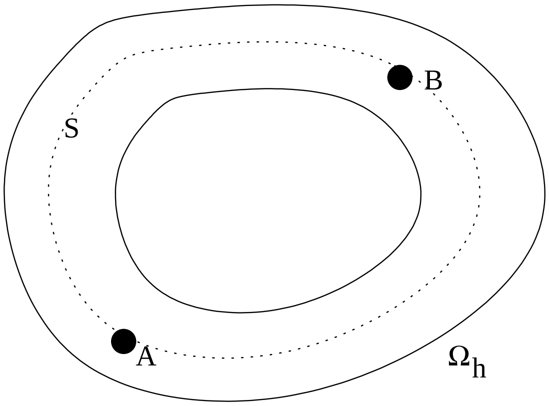


Figure 2.4: Tubular neighborhood.

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 curve will be as smooth as γ for small enough h, see Chapter 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 Chapter 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 §2.2.2 above.

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

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

since

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

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

We then have

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

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 Chapter 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 $\mathbf{H}_\psi\Big|_{\mathcal{S}}$, 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 $\lambda_i\left(\cdot\right)$ 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 $f : [0, d_h] \to I\!\!R$, $f(t) = \psi(\gamma(t))$. 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

$$\int_0^{d_h} \Big|\, \dot f(t) \,\Big| \; dt$$

   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 $|\ddot{f}(t)| \leq \left|\mathbf{H}_{\psi(\gamma(t))}\left[\dot{\gamma}(t), \dot{\gamma}(t)\right]\right| + B_{\tilde{g}}$. 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

$$|\, d^g_{\mathcal{S}} - d^{\tilde{g}}_h\,| \;\leq\; diam(\mathcal{S})\; \sqrt{h}\; \left[M_g\sqrt{\frac{\mathcal{M}_{\mathcal{S}}}{1 - h\mathcal{M}_{\mathcal{S}}} + B_{\tilde{g}}} + M_g\sqrt{h}\frac{\mathcal{M}_{\mathcal{S}}}{1 - h\mathcal{M}_{\mathcal{S}}} + K_{\tilde{g}}\sqrt{h}\right] \tag{2.11}$$

□

From the preceding *Lemma* we obtain:

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

$$\left\|\, d^{\tilde{g}}_{\Omega_h}\Big|_{\mathcal{S}}(q,\cdot) \;-\; d^g_{\mathcal{S}}(q,\cdot)\right\|_{L_\infty(\mathcal{S})} \xrightarrow{h\downarrow 0} 0$$

Remark 2 *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 order 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:*

$$\int_0^{d_h} \left|\,\dot{f}(t)\,\right|\, dt \quad = \quad \sum_{i=1}^N sgn(\dot{f})\Big|_{(a_i,b_i)} \int_{a_i}^{b_i} \dot{f}(t)\, dt$$

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

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 indeed h (at least). Moreover, for simple cases like a sphere (or other convex surfaces), it it very easy to show explicitly that the error is (at least) of order h.[7] *Notwithstanding, it remains to characterize the space of surfaces (and metrics g) for which we can guarantee that $N < \infty$. 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.

2.3 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 with boundary. 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 as 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

$$\begin{cases} \|\nabla f(x)\| = P(x) \;\; \forall\, x \in \Omega \\ f(r) = 0 \end{cases} \tag{2.12}$$

with 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 so the Euclidean gradient is used above.

The upwind numerical scheme to be used for this equation is of the form
($\Delta x_1 = \Delta x_2 = \ldots = \Delta x_d \;=\; \Delta x$) [172]:

$$\begin{cases} \sum_{j=1}^{d} max^2\big(\widehat{f}(p) - m_j, 0\big) = (\Delta x)^2\, P^2(p) \\ m_j = min\big(\widehat{f}(p + \Delta x\vec{e}_j), \widehat{f}(p - \Delta x\vec{e}_j)\big) \end{cases} \tag{2.13}$$

[7] In this case, as in the case of convex surfaces, the geodesic is composed of two straight lines inside the band, tangent to its inner boundary, and a geodesic on the inner boundary of the band; see Figure 1.

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

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

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 [179]. 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 $\widehat{f} = \infty$, forcing them not to be considered as part of the up-winding stencil in the Gudunov's Hamiltonian.

The steps of the algorithm are:

- *Initialization*:

  1. Set $\widehat{f} = 0$ for every point belonging to the set [**Alive**]. These are the seed point/s if they lie on the grid. If the seed is not a grid point, their corresponding **Neighbors**[8] are set **Alive** and are given an initial value $\widehat{f}$ simply computed via interpolation (taking into account the distance from the neighbor grid points to the seed point).
  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 ([**FarAway**]) and insert it to [**NarrowBand**].
  4. Recalculate $\widehat{f}$ for all **Neighbors** using equation (2.13)
  5. Set [**Neighbor**] = *empty set*.
  6. Back to the beginning (step 1).

[8] For a grid point p, any of its $2d$-neighboring points can be written like $p \pm \Delta x_i \vec{e}_i$.

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.

2.3.1 Bounding the Offset h

We now present a technique to estimate h, the size of the offset of the hyper-surface $\mathcal{S}$ 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 (recall that d is the dimension of the space)

$$\Delta x \sqrt{d} \;<\; h \;<\; \min\left\{\frac{1}{\mathcal{M}_{\mathcal{S}}}, W\right\}. \tag{2.14}$$

Let us introduce some additional notation. We denote by *cell* the unit cell of the computational grid. Let x be a point in Ω_h, we denote by $n(x)$ the number of cells $C_1(x), \ldots, C_{n(x)}(x)$ 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 $\mathcal{P}(C)$ 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 $\mathcal{P}(x)$ the set $\bigcup_{i=1}^{n(x)} \mathcal{P}(C_i(x))$.

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 to be as small as possible). That is:

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

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

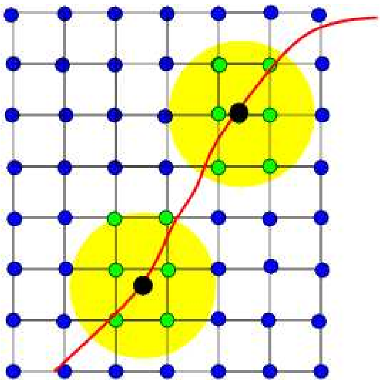


Figure 2.5: We depict the situation that leads to the lower bound for h in the $2D$ case. In red: the curve. In black: the centers of $\overline{B}(x \in \mathcal{S}, d^{1/2}\Delta x)$. In green: the points of $\mathcal{D}(\Omega_h, \Delta x)$ that fall inside $\overline{B}(x, d^{1/2}\Delta x)$ for some $x \in \mathcal{S}$, and in blue those that don't.

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, a situation which can even create a non-simply connected 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 Chapter 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.

2.3.2 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 as 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").

Numerical Error Bound of the Cartesian Fast Marching Algorithm

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 hyper-surface. So we will be dealing with an interpolation stage, that we comment further below in §2.3.2.

Let us fix a point $p \in \mathcal{S}$, and let $\widehat{f}(\cdot)$ be the numerically computed solution (according to (2.13)), and $f(\cdot)$ the *real* viscosity solution of the problem (2.12). The approximation error is then bounded by (see [172])

$$\max_{p \in \mathcal{D}(\Omega, \Delta x)} |\widehat{f}(p) - f(p)| \leq C_L (\Delta x)^{\frac{1}{2}} \tag{2.15}$$

where C_L is a constant. In practice, however, the authors of [172] observed first order accuracy. As we have seen, we 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 2 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 3 *Using the preceding Lemma and (2.15), it is easy to see that for x such that $\mathbf{C}(x) \subset \Omega$:*

$$|\widehat{f}(p) - \widehat{f}(q)| \leq 2C_L (\Delta x)^{\frac{1}{2}} + \|P\|_{L^\infty(\Omega)} \sqrt{d} \Delta x, \quad \forall\, p, q \in \mathcal{P}(x) \tag{2.16}$$

a relation we will shortly use.

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 $\zeta : \mathcal{D}(\Omega, \Delta x) \to I\!\!R$ (Ω being a generic domain, which becomes the band Ω_h for our particular case), 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:

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

for every $x \in \Omega$.

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 $\mathcal{P}(x)$, then:

$$\left| d^g_{\mathcal{S}}(p,x) \,-\, \mathcal{I}\left(\widehat{d^{\tilde{g}}_h}\right)(p,x) \right| \le |d^g_{\mathcal{S}}(p,x) - d^{\tilde{g}}_h(p,x)| \;+\; |d^{\tilde{g}}_h(p,x) - d^{\tilde{g}}_h(p,y)| \tag{2.17}$$

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

and using Proposition 2, Lemma 3, (2.15) and simple manipulations (in that order) we obtain:

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

$$+ \left(\max_{y \in \mathcal{P}(x)} \widehat{d^{\tilde{g}}_h}(p,y) \;-\; \min_{y \in \mathcal{P}(x)} \widehat{d^{\tilde{g}}_h}(p,y) \right)$$

The last term can be dealt with using (2.16). Since we want both $h \downarrow 0$ and $\frac{h}{\Delta x} \uparrow \infty$, in order to have increasing fidelity in the approximation of $d^{\tilde{g}}_h$ by its numeric counterpart $\widehat{d^{\tilde{g}}_h}$,[9] we can choose (for instance) $h = C_x\,(\Delta x)^\gamma$ for some constant $C_x > \sqrt{d}$ and $\gamma \in (0,1)$. We then obtain:

$$\left\| d^g_{\mathcal{S}}(p,\cdot) \,-\, \mathcal{I}\left(\widehat{d^{\tilde{g}}_h}\right)(p,\cdot) \right\|_{L^\infty(\mathcal{S})} \le (\Delta x)^{\frac{\gamma}{2}}\; \mathcal{C}(\Delta x; \mathcal{S}) \tag{2.18}$$

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.

To recap, we have 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 (meaningfully) change the order of the whole numerical approximation, in the worst case scenario. While in the most general case the theoretical bound for the error of our method is of order $h^{1/2}$ and the general order of the error of the underlying numerical scheme is $(\Delta x)^{1/2}$, for all practical purposes the approximation error (over $\mathcal{S}$) between both distances ($d^g_{\mathcal{S}}$ and $d^{\tilde{g}}_h$) is of order h (see remark after Corollary 2.1), and the practical numerical error between $d^{\tilde{g}}_h$ and $\widehat{d^{\tilde{g}}_h}$ is of order $(\Delta x)^\beta$ (for some $\beta \in [\frac{1}{2}, 1)$ for our first order schemes). Then, the practical bound for the total error becomes something of order $(\Delta x)^{\min(\beta,\gamma)}$. Therefore, choosing a big enough γ (< 1) dispels any concerns about worsening the overall error rate when doing Cartesian computations on the band.[10]

[9] This way, we will have an increasing number of points in Ω_h.

[10] Note that the numerical scheme used by the Fast Marching algorithm decreases it's accuracy when non-differentiable points of the distance appear, this can happen for instance when the domain contains the cut locus of the *initial set*, [41]. In any case, $(\Delta x)^{\frac{1}{2}}$ is the slowest error rate achievable.

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.[11]

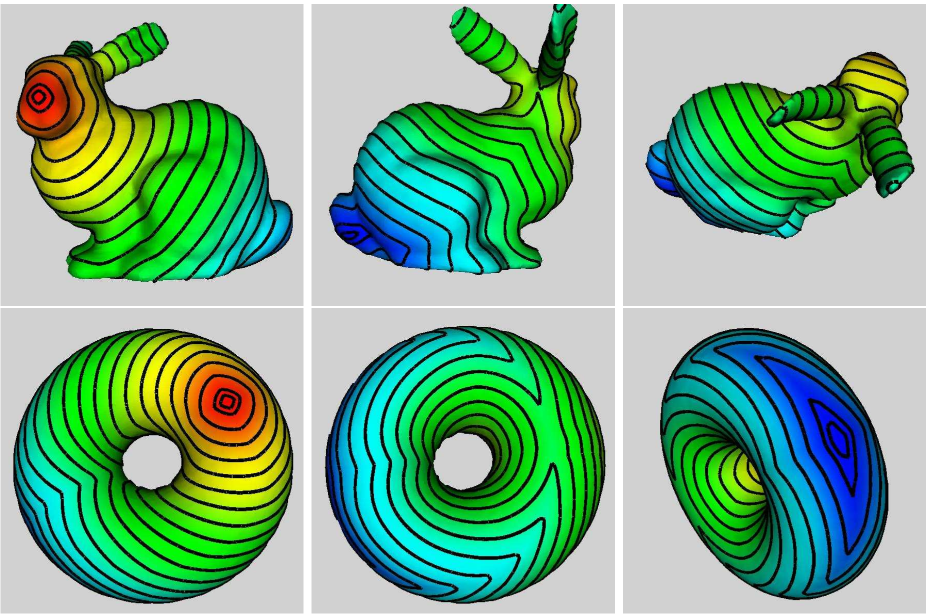


Figure 2.6: Top: Level lines for the intrinsic distance functions on the Stanford Bunny and on a torus. In both rows, the (22) levels shown are 0.03, 0.05, 0.1, ..., 0.95, 0.97 percent of the maximum value of the intrinsic distance, and the coloring of the surface corresponds to the intrinsic distance function. Three views are presented. Note the correct separation between adjacent level lines. Note also how these lines are "parallel".

2.4 Geodesics on Implicit Surfaces

To find geodesic curves on the implicit surface, we back track starting from a specified initial point p towards the seed point p_0, 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 $d^{\tilde{g}}_{\Omega_h}(p_0,\cdot)$, as explained above, we have to solve the following **ODE** (which obviously keeps the curve on $\mathcal{S}$):

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

where $\nabla_\psi d^{\tilde{g}}_{\Omega_h}(p) \stackrel{\triangle}{=} \nabla d^{\tilde{g}}_{\Omega_h}(p) - \left(\nabla d^{\tilde{g}}_{\Omega_h}(p) \cdot \nabla\psi(p) \right) \nabla\psi(p)$ 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.

[11] The number of points in the band can be roughly estimated by the quantity $2\ h\ area[\mathcal{S}]$ when $\Delta x = 1$.

We should note that we are assuming that $\nabla_\psi d^{\tilde{g}}_{\Omega_h}$, the extrinsic gradient of the distance in the band, is a good approximation of $\nabla_{\mathcal{S}} d^g_{\mathcal{S}}$, the intrinsic gradient of the intrinsic distance (and not just $d^{\tilde{g}}_{\Omega_h}$ a good approximation of $d^g_{\mathcal{S}}$ as we have previously proved).

2.5 Extensions

2.5.1 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 [158], 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, [100], and semiconductors manufacturing).

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

$$\mathbf{L}_{\mathbf{G}}\{\mathcal{C}\} \triangleq \int_a^b \sqrt{\mathbf{G}(\mathcal{C}(t))\left[\dot{\mathcal{C}}(t), \dot{\mathcal{C}}(t)\right]}\ dt$$

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

$$d^{\mathbf{G}}_{\mathcal{S}}(x,p) \triangleq \inf_{\mathcal{C}_{px}[\mathcal{S}]}\{\mathbf{L}_{\mathbf{G}}(\mathcal{C})\} \tag{2.19}$$

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

$$d^{\tilde{\mathbf{G}}}_{\Omega_h}(x,p) \triangleq \inf_{\mathcal{C}_{px}[\Omega_h]}\{\mathbf{L}_{\tilde{\mathbf{G}}}(\mathcal{C})\} \tag{2.20}$$

where

$$\mathbf{L}_{\tilde{\mathbf{G}}}\{\mathcal{C}\} \triangleq \int_a^b \sqrt{\tilde{\mathbf{G}}(\mathcal{C}(t))\left[\dot{\mathcal{C}}(t), \dot{\mathcal{C}}(t)\right]}\ dt$$

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

$$\left(\tilde{\mathbf{G}}^{-1}\right)(x)[\nabla d^{\tilde{\mathbf{G}}}_{\Omega_h}, \nabla d^{\tilde{\mathbf{G}}}_{\Omega_h}] = 1 \tag{2.21}$$

The first issue now is the numerical solvability of the preceding equation using a fast marching type of approach. Osher and Helmsen, [158], 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

$$\begin{cases} H(x,\vec{p}) > 0 \;\text{ if } \vec{p} \neq \vec{0} \\ H(x,\vec{p}) \;\text{ is homogeneous of degree 1 in } \vec{p} \\ p_i H_{p_i}(x,\vec{p}) \geq 0 \;\text{ for } 1 \leq i \leq d \;\forall x \in \Omega,\; \forall \vec{p} \end{cases} \tag{2.22}$$

It easily follows that these conditions hold for (2.21) considering $H(x,\vec{p}) \overset{\triangle}{=} \sqrt{(\tilde{\mathbf{G}}^{-1})(x)[\vec{p},\vec{p}]}$, when the matrix $\tilde{\mathbf{G}}^{-1}(x)$ is diagonal. Therefore we can solve this kind 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 (2.20) on the offset Ω_h converges to the intrinsic one on the implicit surface $\mathcal{S}$, i.e., (2.19). This can be done repeating the steps in the convergence proof previously reported in §2.2.3 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: general Intrinsic Eikonal Equations. The extension of this ideas to even more general intrinsic Hamilton-Jacobi equations of the form $\hat{H}(x, \nabla_{\mathcal{S}} u) = \hat{a}(x) \;\; x \in \mathcal{S}$ remains to be studied, and eventual advances will be reported elsewhere.

2.5.2 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., [122, 140, 165, 188, 197, 215, 93] (and some very nice public domain software [140]), to first implicitize the surface and then apply the technique here proposed.[12] 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. This is important, since many applications have been shown to benefit from an implicit surface representation. Moreover, as we have seen, all what we need to have is 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.

2.6 Concluding Remarks

In this chapter 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, see Chapter 4. We have also mentioned that the approach is valid for more general Hamilton-Jacobi equations as well.

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

Many questions remain open. Recently, T. Barth (and independently D. Chopp) have shown techniques to improve the order of accuracy of fast marching methods. It will be interesting to see how the proposed method here can be extended to match such accuracy. Related to this, we are currently working on tighter bounds for the error between $d^{\tilde{g}}_{\Omega_h}$ and $d^{g}_{\mathcal{S}}$. 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, with 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 xooffers a solution to a fundamental problem.

Chapter 3

Covering Submanifolds of $I\!\!R^d$ with Random Euclidean Balls

In practice, we do not have too much control over the way in which points are sampled by the acquisition device (e.g. scanner), or given by the learned sampled data. Therefore it is more realistic to make a probabilistic model of the situation and then try to conveniently estimate the probability of achieving a prescribed level of accuracy in the quantity we wish to estimate. This amounts to assuming the points were sampled on or around the manifold according to some probability measure.[1] Very often it is the case that we need to establish *coverage* properties for the point cloud with respect to the object it represents. We propose a model for this task and derive some bounds that will be useful ahead.

Let $\mathcal{P}_n \triangleq \{p_1, \ldots, p_n\}$ be a set of n different points sampled from the *compact* k-dimensional submanifold $\mathcal{S} \subset I\!\!R^d$ and define $\Omega^h_{\mathcal{P}_n} \triangleq \bigcup_{i=1}^n B(p_i, h)$.[2] In what follows we will first find an upper bound for $\mathbb{P}\left(\mathcal{S} \nsubseteq \Omega^h_{\mathcal{P}_n}\right)$ and then an upper bound for $\mathbb{P}\left(d_{\mathcal{H}}(\mathcal{P}_n, \mathcal{S}) > h\right)$.

We now present our model for the current setting: We assume that the points in $\mathcal{P}_n$ are independently and identically sampled on or around the sub-manifold $\mathcal{S}$ in a with a probability law given by the measure ν.[3] We will write this as $p_i \sim \nu$. As we will see below, the fundamental quantities one must control are $f_\nu(r) \triangleq \min_{x\in\mathcal{S}} \nu(B(x,r))$, which can be interpreted as an indicator of the presence of holes at scale r, and $g_\nu(s) \triangleq \nu\left(\Omega^s_{\mathcal{S}}\right)$ which measures how much probability mass is located inside a (small) tube around $\mathcal{S}$. We will further assume that ν has *no atoms*.

Remark 4 *It is possible to contemplate the case of p_i having different probability laws ν_i but still being independent. In such a case, one should substitute $f_\nu(r)$ by $\min_{1\leq i\leq n} f_{\nu_i}(r)$ and $g_\nu(s)$ by $\min_{1\leq i\leq n} g_{\nu_i}(s)$.*

We now present a few lemmas we will need ahead.

Lemma 4 *Let $x \in \mathcal{S}$ be a fixed point on $\mathcal{S}$. Then under the hypotheses on $\mathcal{P}_n$ described above for small*

[1] In the case of objects sampled using a 3D scanner, this probability measure models the acquisition process itself. As we will see below, one needs to require that the acquisition process does not leave big holes.

[2] The balls now used are defined with respect to the metric of $I\!\!R^d$, they are not intrinsic. Other covering shapes could be used as well, see comments ahead.

[3] This means that for any subset $A \subseteq I\!\!R^d$, and any $p_i \in \mathcal{P}_n$, $\mathbb{P}\left(p_i \in A\right) = \nu(A)$.

enough $h > 0$,

$$\mathbb{P}\left(\left\{x \notin \Omega^h_{\mathcal{P}_n}\right\}\right) \leq (1 - f_\nu(h))^n .$$

Proof:

$$\begin{aligned}
\mathbb{P}\left(\left\{x \notin \Omega^h_{\mathcal{P}_n} \cap \mathcal{S}\right\}\right) &= \mathbb{P}\left(\left\{\bigcap_{i=1}^{n} \{x \notin B(p_i, h)\}\right\}\right) \\
&= \mathbb{P}\left(\left\{\bigcap_{i=1}^{n} \{p_i \notin B(x, h)\}\right\}\right) \\
&= \prod_{i=1}^{n} \mathbb{P}\left(\{p_i \notin B(x, h)\}\right) \\
&= \prod_{i=1}^{n} \left(1 - \mathbb{P}\left(\{p_i \in B(x, h)\}\right)\right) \\
&= \left(1 - \mathbb{P}\left(\{p_i \in B(x, h)\}\right)\right)^n
\end{aligned}$$

and we conclude using the definition of f_ν. □

Corollary 2 *Under the hypotheses of the previous Lemma, let $\delta \in (0, h)$, then, for $q \in \mathcal{S}$:*

$$\mathbb{P}\left(B_{\mathcal{S}}(q, \delta) \not\subseteq \Omega^h_{\mathcal{P}_n}\right) \leq (1 - f_\nu(h - \delta))^n .$$

Proof:

We prove the following inclusion of events holds: $\{q \in \Omega^{h-\delta}_{\mathcal{P}_n}\} \subseteq \{B_{\mathcal{S}}(x, \delta) \subset \Omega^h_{\mathcal{P}_n}\}$. In fact, let $\{q \in \Omega^{h-\delta}_{\mathcal{P}_n}\}$ hold, then for some $p \in \mathcal{P}_n$ $|q - p| \leq h - \delta$. Let $x \in B_{\mathcal{S}}(q, \delta)$, then $|x - q| \leq d_{\mathcal{S}}(x, q) \leq \delta$ and hence $|x - p| \leq |x - q| + |q - p| \leq \delta + h - \delta = h$. □

Proposition 1 *Let the set of hypotheses sustaining all of the previous lemmas hold. Then*

$$\mathbb{P}\left(\mathcal{S} \not\subseteq \Omega^h_{\mathcal{P}_n}\right) \leq \mathcal{N}_{\mathcal{S}}\left(\frac{h}{2}\right) e^{-n f_\nu(\frac{h}{2})}. \tag{3.1}$$

where $\mathcal{N}_{\mathcal{S}}$ stands for the cardinality of a $\frac{h}{2}$-covering net of $\mathcal{S}$ by euclidean balls.

Remark 5 *If a prescribed probability of coverage p is desired, given a certain covering radius h, then we find a lower bound for the number of sample points needed:*

$$n \geq \frac{1}{f_\nu(\frac{h}{2})}\left(\ln(\frac{1}{1-p}) + \ln \mathcal{N}_{\mathcal{S}}(\frac{h}{2})\right) \tag{3.2}$$

provided $f_\nu(\frac{h}{2}) > 0$.

Proof:

Consider a finite $\frac{h}{2}$-net covering $\mathcal{S}$ by euclidean balls, that is $\mathcal{S} = \bigcup_{i=1}^{\mathcal{N}_{\mathcal{S}}(\frac{h}{2})} B(q_i, \frac{h}{2})$, then

$$
\begin{aligned}
\mathbb{P}\left(\mathcal{S} \nsubseteq \Omega^h_{\mathcal{P}_n}\right) &= \mathbb{P}\left(\bigcup_{x\in\mathcal{S}}\{x \notin \Omega^h_{\mathcal{P}_n}\}\right) \\
&= \mathbb{P}\left(\bigcup_{i=1}^{\mathcal{N}_{\mathcal{S}}(\frac{h}{2})} \quad \bigcup_{x\,\in\, B(q_i,\frac{h}{2})} \{x \notin \Omega^h_{\mathcal{P}_n}\}\right) \\
&\leq \mathcal{N}_{\mathcal{S}}\left(\frac{h}{2}\right) \max_{1\leq i\leq \mathcal{N}_{\mathcal{S}}\left(\frac{h}{2}\right)} \mathbb{P}\left(\bigcup_{x\,\in\, B(q_i,\frac{h}{2})} \{x \notin \Omega^h_{\mathcal{P}_n}\}\right) \\
&= \mathcal{N}_{\mathcal{S}}\left(\frac{h}{2}\right) \max_{1\leq i\leq \mathcal{N}_{\mathcal{S}}\left(\frac{h}{2}\right)} \mathbb{P}\left(B(q_i,\frac{h}{2}) \nsubseteq \Omega^h_{\mathcal{P}_n}\right).
\end{aligned}
$$

We conclude by using Corollary 2 and then the inequality $1-u \leq e^{-u}$, valid for $u \geq 0$. □

Lemma 5 ***(Bounding the Covering Number)*** *Under the hypotheses of the previous Lemma and further assuming $\mathcal{S}$ to be compact, we have that for any small $\delta > 0$ there exists a δ-covering of $\mathcal{S}$ by euclidean balls with cardinality bounded by*

$$
\mathcal{N}_{\mathcal{S}}\left(\delta\right) \leq \frac{1}{f_\nu(\frac{\delta}{2})}. \tag{3.3}
$$

Proof:
The idea is constructive and folkloric. Let q_1 be any point in $\mathcal{S}$, then choose $q_2 \in \mathcal{S}\backslash B(q_1,\delta)$. Then choose $q_3 \in \mathcal{S}\backslash\{B(q_1,\delta)\cup B(q_2,\delta)\}$. Iterate this procedure until it is no longer possible to choose any point $q \in \mathcal{S}\backslash\{\cup_{k=1}^{\mathcal{N}_{\mathcal{S}}(\delta)} B(q_k,\delta)\}$, in that case $\mathcal{S} \subset \cup_{k=1}^{\mathcal{N}_{\mathcal{S}}(\delta)} B(q_k,\delta)$. Note that $B(q_k,\frac{\delta}{2}) \cap B(q_l,\frac{\delta}{2}) = \emptyset$ if $k \neq l$. Then we have

$$
1 \geq \nu\left(\cup_{k=1}^{\mathcal{N}_{\mathcal{S}}(\delta)} B(q_k,\frac{\delta}{2})\right) = \sum_{i=1}^{\mathcal{N}_{\mathcal{S}}(\delta)} \nu(B(q_k,\frac{\delta}{2})) \geq \mathcal{N}_{\mathcal{S}}(\delta)\; f_\nu\left(\frac{\delta}{2}\right).
$$

□

Remark 6 *Using the last Lemma, we find a somewhat simpler bound for the probability of not achieving coverage:*

$$
\mathbb{P}\left(\mathcal{S} \nsubseteq \Omega^h_{\mathcal{P}_n}\right) \leq \frac{e^{-nf^h_\nu}}{f^h_\nu} \tag{3.4}
$$

where $f^h_\nu := f_\nu(\frac{h}{2})$.

Remark 7 *In general, or at least in the applications that follow, Chapters 4 and 5, one will require h tending to* 0.

Note that for $\{a_m\}_{m\in I\!N}$, $a_m \downarrow 0$, $\frac{e^{-ma_m}}{a_m}$ goes to zero as $m \uparrow \infty$ if a_m is asymptotically greater than or equal $\frac{\log m}{m}$. Then, in order to have the right hand side of (3.4) tend to zero we should have, for a sequence $\{h_n\}_n$ with $h_n \downarrow 0$ as $n \uparrow \infty$:[4]

$$
f^h_\nu \gtrsim \frac{\log n}{n}. \tag{3.5}
$$

[4] This kind of conditions are common in the literature of Random Coverings, [182, 60].

Let's consider now the simple case of having a Uniform probability measure on $\mathcal{S}$.[5] *In this case,* $f_\nu(r) = \min_{x\in\mathcal{S}} \frac{\mu(B(x,r)\cap\mathcal{S})}{\mu(\mathcal{S})} \geq \min_{x\in\mathcal{S}} \frac{\mu(B_\mathcal{S}(x,r))}{\mu(\mathcal{S})}$. *Now, using Bishop's Volume Comparison Theorem (see [37, 175]), we obtain* $\min_{\zeta\in\mathcal{S}} \mu\left(B_\mathcal{S}(\zeta,r)\right) \geq \omega_k r^k + \theta_\mathcal{S}(r)$, *where* $\frac{\theta_\mathcal{S}(r)}{r^q} \to 0$ *when* $r \to 0$ *for* $q \leq k+1$. *Hence* $f_\nu(r) \geq \frac{\omega_k r^k + \theta_\mathcal{S}(r)}{\mu(\mathcal{S})}$ *and the condition relating* h, k *and* n *should then be* $h^k \gtrsim \left(\mathbf{a}(\mathcal{S})\frac{2^k}{\omega_k}\right)\frac{\log n}{n}$. *Also, under condition (3.5) we can estimate the rate at which* $\frac{e^{-nf_\nu^h}}{f_\nu^h}$ *approaches zero as* $n \uparrow \infty$. *For example, with* $f_\nu^h \simeq \frac{\log n}{n}$, $\frac{e^{-nf_\nu^h}}{f_\nu^h} \simeq \frac{1}{\log n}$ *as* $n \uparrow \infty$. *Note that of course we can speed up the convergence towards zero by choosing slower variations of* $f_\nu^{h_n}$ *with* n, *for instance, with* $f_\nu^{h_n} \simeq \frac{\log n^\gamma}{n}$, *with* $\gamma \geq 1$ *we have* $\frac{e^{-nf_\nu^h}}{f_\nu^h} \simeq \frac{1}{\gamma(\log n)n^{\gamma-1}}$ *as* $n \uparrow \infty$.

Bounds for $\mathbb{P}\left(\mathcal{S} \nsubseteq \Omega^h_{\mathcal{P}_n}\right)$ *similar to ours can be found in [73, 101]. We should finally point out that the problem of covering a certain domain (usually* S^1*) with balls centered at random points sampled from this domain has been studied by many authors, [184, 73, 72, 105, 182, 114, 95, 101], and even by Shannon in [181].*

Controlling the Hausdorff Distance

It will also come handy to obtain a lower bound for $\mathbb{P}\left(d_\mathcal{H}(\mathcal{P}_n,\mathcal{S}) \leq \delta\right)$.[6] Clearly, this probability equals $\mathbb{P}\left(\left\{\mathcal{S} \subseteq \Omega^\delta_{\mathcal{P}_n}\right\} \cap \left\{\mathcal{P}_n \subseteq \Omega^\delta_\mathcal{S}\right\}\right)$, by definition of Hausdorff distance. Now, using the union bound and independence of p_i, p_j when $i \neq j$ we immediately find:

$$\begin{aligned}\mathbb{P}\left(d_\mathcal{H}(\mathcal{P}_n,\mathcal{S}) \leq \delta\right) &\geq \mathbb{P}\left(\mathcal{S} \subseteq \Omega^\delta_{\mathcal{P}_n}\right) + \mathbb{P}\left(\mathcal{P}_n \subseteq \Omega^\delta_\mathcal{S}\right) - 1 \\ &\geq -\frac{e^{-nf_\nu^\delta}}{f_\nu^\delta} + \left(g_\nu(\delta)\right)^n \qquad (3.6)\end{aligned}$$

To sum up, in this chapter we have presented basic conditions for the union of (Euclidean) balls centered at the point cloud to cover (with probability) the underlying shape. When these conditions hold, we are then free to work with this Euclidean structure, as done for example in the next chapter for computing intrinsic geodesic distances without manifold reconstruction.

[5] For simplicity of exposition we will restrict ourselves to the case when $\mathcal{S}$ has no boundary. The modifications needed in our arguments are of the same nature as those in [17].

[6] This is immaterial when the sampling is noiseless: $\mathcal{P}_n \subseteq \mathcal{S}$.

Chapter 4

Distance Functions on Point Clouds

The goal of this chapter is to show how to compute geodesic distances for manifolds given as point cloud data, one if not the most fundamental computation for shape analysis. A number of key building blocks are part of the framework here introduced. The first one is based on the fact that distance functions intrinsic to a given sub-manifold of $I\!R^d$ can be accurately approximated by Euclidean distance functions computed in a thin offset band that surrounds this manifold. This concept was introduced in Chapter 2, where convergence results were given for co-dimension one sub-manifolds of $I\!R^d$ (hyper-surfaces) without boundary.

The approximation of intrinsic distance functions (and geodesics) by extrinsic Euclidean ones permits to compute them using computationally optimal algorithms in Cartesian grids (as long as the discretization operation is permitted, memory wise, see §4.3.) These algorithms are based on the fact that the distance function satisfies a Hamilton-Jacobi partial differential equation (see §4.1), for which consistent and fast algorithms have been developed in Cartesian grids [99, 178, 179, 199][1] (see [117] for extensions to triangular meshes and [197] for other Hamilton-Jacobi equations).

Once these basic results are available, we can then proceed and work with point clouds. The basic idea here is to construct the offset band directly from the point cloud and without the intermediate step of manifold reconstruction. This is addressed in §4.2.1 and §4.2.2 for points which are (maybe noisy) manifold samples. For this (random) cases, we use the bounds for the probability that the constructed offset band contains the underlying manifold, as presented in Chapter 3. In the experimental section, §4.3, we present a number of important applications. These applications are given to show the importance of this novel computational framework, and are by no means exhaustive.

We should note that to the best of our knowledge, the only additional works explicitly addressing the computation of distance functions and geodesics for point clouds are the ones reported in [17, 189][2] and recently the one reported in [81]. This last paper is also mesh based, and follows the geodesics approach in Isomap with a novel neighborhood/connectivity concept and a number of interesting theoretical results and novel dimensionality estimation contributions.

Very recently, some further work, in a very similar spirit to ours, has been done to understand topological

[1] Tsitsiklis first described an optimal-control type of approach to solve the Hamilton-Jacobi equation, while independently Sethian and Helmsen both developed techniques based on upwind numerical schemes.

[2] In addition to studying the computation of distance functions on point clouds, [17, 189] address the important combination of this with multidimensional scaling for manifold analysis. Prior work on using geodesic distances and multidimensional scaling can be found in [177].

properties of a submanifold represented by a point cloud under probabilistic assumptions on the sampling, [155]. Some of the results there can be obtained following our approach.

4.1 Prelude

In Chapter 2, we presented a new approach for the computation of weighted intrinsic distance functions on hyper-surfaces. We proved convergence theorems and addressed the fast, computationally optimal, computation of such approximations, see comments after Theorem 3 below. The key starting idea is that distance functions satisfy the (intrinsic) Eikonal equation, a particular case of the general class of Hamilton-Jacobi partial differential equations. Given $p \in \mathcal{S}$ (an hyper-surface in $I\!R^d$), we want to compute $d_{\mathcal{S}}(p, \cdot) : \mathcal{S} \to I\!R^+ \cup \{0\}$, the intrinsic distance function from every point on $\mathcal{S}$ to p. It is well known that the distance function $d_{\mathcal{S}}(p, \cdot)$ satisfies, in the viscosity sense (see [138]), the equation

$$\begin{cases} \|\nabla_{\mathcal{S}} d_{\mathcal{S}}(p, x)\| = 1 \ \forall x \in \mathcal{S} \\ d_{\mathcal{S}}(p, p) = 0, \end{cases}$$

where $\nabla_{\mathcal{S}}$ is the intrinsic differentiation (gradient). Instead of solving this intrinsic Eikonal equation on $\mathcal{S}$, we solve the corresponding extrinsic one in the offset band $\Omega^h_{\mathcal{S}}$:

$$\begin{cases} \|\nabla_x d_{\Omega^h_{\mathcal{S}}}(p, x)\| = 1 \ \forall x \in \Omega^h_{\mathcal{S}} \\ d_{\Omega^h_{\mathcal{S}}}(p, p) = 0, \end{cases}$$

where $d_{\Omega^h_{\mathcal{S}}}(p, \cdot)$ is the Euclidean distance and therefore now the differentiation is the usual one.

We now recall the following theorem from Chapter 2 for the reader's convenience.

Theorem 3 *Let p and q be any two points on the smooth (orientable, without boundary) hyper-surface $\mathcal{S}$, then $\left| d_{\mathcal{S}}(p, q) - d_{\Omega^h_{\mathcal{S}}}(p, q) \right| \le C_{\mathcal{S}} \sqrt{h}$, for small enough h, where $C_{\mathcal{S}}$ is a constant depending on the geometry of $\mathcal{S}$.*

This simplification of the intrinsic problem into an extrinsic one permits the use of the computationally optimal algorithms mentioned in the introduction. This makes computing intrinsic distances, and from them geodesics, as simple and computationally efficient as computing them in Euclidean spaces. Moreover, as detailed in Chapter 2, the approximation of the intrinsic distance $d_{\mathcal{S}}$ by the extrinsic Euclidean one $d_{\Omega^h_{\mathcal{S}}}$ is never less accurate than the numerical error of these algorithms.

The result above was limited to hyper-surfaces of $I\!R^d$ (co-dimension one submanifolds of $I\!R^d$) without boundary, and the theory was applied to implicit surfaces, where computing the offset band is straightforward. It is the purpose of the present chapter to extend the aforementioned Theorem to deal with: (1) submanifolds of $I\!R^d$ of any codimension and possibly with boundary[3], (2) convergence of geodesic curves in addition to distance functions, (3) submanifolds of $I\!R^d$ represented as point clouds and (4) random sampling of submanifolds of $I\!R^d$ in presence of noise. We should note that Theorem 3 holds even when the metric is not the one inherited from $I\!R^d$, obtaining weighted distance functions, Chapter 2.

[3] We will later impose some convexity conditions on the boundary in order to get rate of convergence estimates. However, the uniform convergence in itself doesn't require other hypotheses beyond smoothness.

4.2 Submanifolds of $I\!R^d$ with Boundary

We first extend Theorem 1 to more general manifolds and we deal not only with distance functions but also with geodesics. The first extension is important for the learning of high-dimensional manifolds from samples and for scanned open volumes. The extension to geodesics is important for path planning on surfaces and for finding special curves such as crests and valleys, see [15] and also Chapter 7.

First we need to recall some results that will be key ingredients in our proofs below. All our results rest upon a certain degree of smoothness of geodesics in manifolds with boundary. We use "shortest path" and "minimizing geodesic" interchangeably.

Theorem 4 ([2]) *Let $\mathcal{M}$ be a C^3 Riemannian manifold with C^1 boundary $\partial\mathcal{M}$. Then, any shortest path of $\partial\mathcal{M}$ is C^1.*

We will eventually need more regularity on the geodesics than simply C^1. This is achieved by requiring more regularity of the boundary.

Theorem 5 ([139]) *Let $\mathcal{U} : I\!R^d \to I\!R$ be a C^3 function such that for some $h \in I\!R$*

1. *the interior of $\{x \in I\!R^d |\, \mathcal{U}(x) = h\}$ is non-empty and there we have $D\mathcal{U}(x) \neq 0$.*
2. *the "obstacle" $\{x \in I\!R^d |\, \mathcal{U}(x) \geq h\}$ is compact.*

Let p and q be any two points in the same connected component of $\{x \in I\!R^d |\, \mathcal{U}(x) \leq h\}$, then the shortest (constrained) path joining both points is C^1 and has Lipschitz first derivative.

We now present the usual definition of length:

Definition 1 *Let $\alpha : [a,b] \to I\!R^d$ be a curve, then we define its length $\mathbf{L}(\alpha)$ as*

$$\mathbf{L}(\alpha) \overset{\triangle}{=} \sup_{a=t_0<\ldots<t_N=b} \sum_{k=0}^{N-1} \|\alpha(t_{k+1}) - \alpha(t_k)\|.$$

Remark 8 *Note that if α is Lipschitz with constant $\mathcal{L}_\alpha$, then $\mathbf{L}(\alpha) = \int_a^b \|\dot\alpha(t)\| dt$ and $\mathbf{L}(\alpha) \leq \mathcal{L}_\alpha (b-a)$.*

Proposition 2 *Let $\mathcal{S}$ be a smooth compact submanifold of $I\!R^d$ with boundary $\partial\mathcal{S}$. Let x, y be any two points in $\mathcal{S}$. Then, $d_{\Omega^h_{\mathcal{S}}}(x,y)$ converges pointwise as $h \downarrow 0$.*

Proof:
Since $\Omega^h_{\mathcal{S}} \subseteq \Omega^{h'}_{\mathcal{S}}$ if $h' \geq h$, we have that $d_{\Omega^h_{\mathcal{S}}}(x,y) \geq d_{\Omega^{h'}_{\mathcal{S}}}(x,y)$ Also, for any $h>0$, $d_{\Omega^h_{\mathcal{S}}}(x,y) \leq d_{\mathcal{S}}(x,y) \leq \mathbf{diam}(\mathcal{S}) < +\infty$. Hence, the sequence $\{d_{\Omega^h_{\mathcal{S}}}(x,y)\}_{h>0}$ (for fixed x and y over $\mathcal{S}$) is bounded and non-decreasing, therefore it converges to the supremum of its range. □

Theorem 6 *Let $\mathcal{S}$ be a compact C^2 submanifold of $I\!R^d$ with (possibly empty) smooth boundary $\partial\mathcal{S}$. Let x, y be any two points in $\mathcal{S}$. Then we have*

1. *Uniform convergence of distances:*

$$d_{\Omega_{\mathcal{S}}^h}|_{\mathcal{S}\times\mathcal{S}}(\cdot,\cdot) \overset{h\downarrow 0}{\rightrightarrows} d_{\mathcal{S}}(\cdot,\cdot).$$

2. *Convergence of geodesics: Let x and y be joined by a unique minimizing geodesic $\gamma_{\mathcal{S}}:[0,1]\to\mathcal{S}$ over $\mathcal{S}$, and let $\gamma_h:[0,1]\to\Omega_{\mathcal{S}}^h$ be a $\Omega_{\mathcal{S}}^h$-minimizing geodesic, then*

$$\gamma_h \overset{h\downarrow 0}{\rightrightarrows} \gamma_{\mathcal{S}}.$$

Proof:
Given our hypothesis on $\mathcal{S}$, and according to [69], there exists $H>0$ such that $\partial\Omega_{\mathcal{S}}^h$ is $C^{1,1}$ for all $0<h\leq H$. Then Theorem 4 guarantees that for $0<h\leq H$, $\gamma_h:[0,1]\to\Omega_{\mathcal{S}}^H$, the $\Omega_{\mathcal{S}}^h$ length minimizing geodesic joining x and y is of class C^1. Since $d_{\Omega_{\mathcal{S}}^h}(x,y)\leq d_{\mathcal{S}}(x,y)\leq \mathbf{diam}(\mathcal{S})<+\infty$ for any $h\in(0,H]$, we see that we can admit our $\Omega_{\mathcal{S}}^h$-geodesics to have Lipschitz constant $\mathcal{L}\leq\mathbf{diam}(\mathcal{S})$. Obviously, the set $\Omega_{\mathcal{S}}^H$ is bounded, and then the family $\{\gamma_h\}_{0<h\leq H}$ is bounded and equicontinuous. Hence, by Ascoli-Arzelá's Theorem, there exist a subsequence $\{\gamma_{h_k}\}_{k\in I\!N}$ and a curve $\gamma_0\in C^0([0,1],\mathcal{S})$ such that $\max_{t\in[0,1]}\|\gamma_{h_k}(t)-\gamma_0(t)\|\overset{h_k\downarrow 0}{\longrightarrow}0$.

Moreover, by writing $|\gamma_0(t)-\gamma_0(t')|\leq|\gamma_{h_k}(t)-\gamma_0(t)|+|\gamma_{h_k}(t')-\gamma_0(t')|+\mathcal{L}|t-t'|$ and using the (pointwise) convergence of γ_{h_k} towards γ_0, we find that $\mathcal{L}$ is also a Lipschitz constant for γ_0. Then we have $\gamma_0\in C^{0,1}([0,1],\mathcal{S})$.

Now, since γ_0 lies on $\mathcal{S}$ but may not be a shortest path, we have that its (finite) length is greater than or equal to $d_{\mathcal{S}}(x,y)$. We also have the trivial inequality $d_{\mathcal{S}}(x,y)\geq d_{\Omega_{\mathcal{S}}^h}(x,y)$. Putting all together we obtain

$$\mathbf{L}(\gamma_h)=d_{\Omega_{\mathcal{S}}^h}(x,y)\leq d_{\mathcal{S}}(x,y)\leq\mathbf{L}(\gamma_0).$$

Therefore

$$\limsup_{h\downarrow 0}\mathbf{L}(\gamma_h)=\limsup_{h\downarrow 0}d_{\Omega_{\mathcal{S}}^h}(x,y)\leq d_{\mathcal{S}}(x,y)\leq\mathbf{L}(\gamma_0).$$

Note that $\mathbf{L}(\gamma_0)=\mathbf{L}(\lim_{h_k\downarrow 0}\gamma_{h_k})\leq\liminf_{h_k\downarrow 0}\mathbf{L}(\gamma_{h_k})$. This is the semicontinuity of length, an immediate consequence of its definition , see [121].

Since $\liminf_{h_k\downarrow 0}(\cdot)\leq\limsup_{h_k\downarrow 0}(\cdot)\leq\limsup_{h\downarrow 0}(\cdot)$, we see that $\limsup_{h\downarrow 0}d_{\Omega_{\mathcal{S}}^h}(x,y)=\limsup_{h\downarrow 0}\mathbf{L}(\gamma_h)$ equals $d_{\mathcal{S}}(x,y)$, for all x and y in $\mathcal{S}$. From Proposition 1, we find that in fact $\lim_{h\downarrow 0}d_{\Omega_{\mathcal{S}}^h}(x,y)$ exists and equals $d_{\mathcal{S}}(x,y)$. Then, we have that the function $d_{\Omega_{\mathcal{S}}^h}|_{\mathcal{S}\times\mathcal{S}}(\cdot,\cdot)$ satisfies:

1. $d_{\Omega_{\mathcal{S}}^h}|_{\mathcal{S}\times\mathcal{S}}:\mathcal{S}\times\mathcal{S}\to I\!R\cup\{0\}$ is continuous for each $H>h>0$.
2. for each $(x,y)\in\mathcal{S}\times\mathcal{S}$, $\{d_{\Omega_{\mathcal{S}}^h}|_{\mathcal{S}\times\mathcal{S}}(x,y)\}_h$ is non-decreasing.
3. $d_{\Omega_{\mathcal{S}}^h}|_{\mathcal{S}\times\mathcal{S}}(\cdot,\cdot)$ converges pointwise towards $d_{\mathcal{S}}(\cdot,\cdot)$, which is continuous.

Then by Dini's Uniform Convergence Theorem (see [9]) we can conclude that the convergence is uniform.

We can also see that γ_0 must be a minimizing geodesic of $\mathcal{S}$ since from the above chain of equalities $\mathbf{L}(\gamma_0)=d_{\mathcal{S}}(x,y)$. Then, if there was only one such curve joining x with y, we would have uniform convergence (along any subsequence!) of γ_h towards γ_0.[4] □

Remark 9 *In the previous Theorem, the convergence (of distances) is uniform but we will have forfeited rate of convergence estimates unless we impose additional conditions on $\partial\mathcal{S}$, as we do in Corollary 3. Note that the new setting is wider than the one considered in Theorem 3 since the codimension of the underlying*

[4] This follows from the fact that uniform convergence of γ_h to γ_0 is equivalent to the statement that for any subsequence $\{\gamma_{h_i}\}$ there exists a further subsubsequence $\{\gamma_{h_{i_k}}\}$ uniformly converging to γ_0.

manifold is not necessarily 1*. This is very important for applications such as dimensionality reduction, where the dimension of the underlying manifold is unknown beforehand.*

Corollary 3 *Let* $\mathcal{S}$ *and* $\partial\mathcal{S}$ *satisfy the hypotheses of Theorem 6 . Let* $\{\Sigma_i\}_{i\in I\!N}$ *be a family of compact of sets in* $I\!R^d$ *such that* $\mathcal{S} \subseteq \Sigma_i \ \forall \ i \in I\!N$ *and* $d_{\mathcal{H}}(\Sigma_i, \mathcal{S}) \overset{i\uparrow+\infty}{\longrightarrow} 0$*. Then,*

$$d_{\Sigma_i}(\cdot,\cdot)\,|_{\mathcal{S}\times\mathcal{S}} \overset{i\uparrow+\infty}{\rightrightarrows} d_{\mathcal{S}}(\cdot,\cdot),$$

where $d_{\mathcal{H}}$ *stands for the Hausdorff distance between sets.*

We now present a uniform rate of convergence result for the distance in the band in the case $\partial\mathcal{S} = \emptyset$, and from this we deduce Corollary 3 below, which deals with the case $\partial\mathcal{S} \neq \emptyset$. This result generalizes the one presented in Chapter 2 because it allows for any codimension.

Theorem 7 *Under the same hypotheses of the Theorem above, with* $\partial\mathcal{S} = \emptyset$*, we have that for small enough* $h > 0$*:*

$$\max_{(x,y)\in\mathcal{S}\times\mathcal{S}} \left| d_{\Omega^h_{\mathcal{S}}}\,|_{\mathcal{S}\times\mathcal{S}}(x,y) - d_{\mathcal{S}}(x,y) \right| \leq C_{\mathcal{S}}\sqrt{h}, \tag{4.1}$$

where the constant $C_{\mathcal{S}}$ *does not depend on* h*. Also, we have the "relative" rate of convergence bound:*

$$1 \leq \sup_{\substack{x,\,y\,\in\,\mathcal{S}\\ x\neq y}} \frac{d_{\mathcal{S}}(x,y)}{d_{\Omega^h_{\mathcal{S}}}(x,y)} \leq 1 + C_{\mathcal{S}}\sqrt{h}. \tag{4.2}$$

Proof:
This is a remake of our proof of the main theorem in Chapter 2, therefore we skip some technical details which can be found there. All along the proof we will sometimes write d_h instead of $d_{\Omega^h_{\mathcal{S}}}$ for the sake of notational simplicity. We will denote by k $(\leq n-1)$ the dimension of $\mathcal{S}$.

Let γ_0 be the arc length parametrized $\mathcal{S}$-shortest path Joining the points $x,\ y \in \mathcal{S}$; clearly, we have $trace(\gamma_0) \subset \mathcal{S}$. Let γ_h be the $\Omega^h_{\mathcal{S}}$ arc length parametrized shortest path joining x and y, which, as we know from Theorem 6, uniformly converges toward γ_0. For a number H as in the proof of Theorem 6, we have $\gamma_h \in C^{1,1}([0,d_h],\mathcal{S})$, and also $\eta : \Omega^H_{\mathcal{S}} \to I\!R$ defined by $\eta(x) \overset{\triangle}{=} \frac{1}{2}d^2(x,\mathcal{S})$ is smooth, see Appendix A.2. We define the projection operator $\Pi_{\mathcal{S}} : \Omega^H_{\mathcal{S}} \to \mathcal{S}$ by $\Pi_{\mathcal{S}}(x) = x - D\eta(x)$. We refer the reader to Appendix A.2 for properties of $\Pi_{\mathcal{S}}$ and η which we use below.

Now, $d_{\Omega^h_{\mathcal{S}}}(x,y) = \mathbf{L}(\gamma_h) \leq d_{\mathcal{S}}(x,y) \leq \mathbf{L}(\Pi_{\mathcal{S}}(\gamma_h))$, then

$$\begin{aligned}
d_{\mathcal{S}}(x,y) - d_{\Omega^h_{\mathcal{S}}}(x,y) &\leq \left|\mathbf{L}(\Pi_{\mathcal{S}}(\gamma_h)) - \mathbf{L}(\gamma_h)\right| \\
&\leq \int_0^{d_h} \left\| \dot{\overline{\Pi_{\mathcal{S}}(\gamma_h(t)) - \gamma_h(t)}} \right\| \, dt \\
&= \int_0^{d_h} \left\| \dot{\overline{D\eta(\gamma_h(t))}} \right\| \, dt \\
&\leq \sqrt{d_h \int_0^{d_h} \dot{V}(t)\cdot\dot{V}(t)\, dt} \quad \text{(by Cauchy-Schwarz Ineq.)} \\
&\leq \sqrt{d_h \int_0^{d_h} V(t)\cdot\ddot{V}(t)\, dt} \quad \text{(Integrating by parts, see below.)}
\end{aligned}$$

where $V(t) \overset{\triangle}{=} D\eta(\gamma_h(t))$ and $V(0) = V(1) = 0$, see Appendix A.2.

Also $\dot{V}(t) = D^2\eta(\gamma_h(t))\dot{\gamma}(t)$ and since $\dot{\gamma_h}$ is Lipschitz and η is smooth, $\ddot{V}(t)$ exists almost everywhere and $\ddot{V}(t) = D^3\eta(\gamma_h(t))[\dot{\gamma}_h(t), \dot{\gamma}_h(t)] + D^2\eta(\gamma_h(t))\ddot{\gamma}(t)$ at points of existence. Then since $D^3\eta D\eta = D^2\eta(I - D^2\eta)$ and $D^2\eta D\eta = D\eta$ (see Appendix A.2),

$$\begin{aligned} V \cdot \ddot{V} &= D^3\eta(\gamma_h)[D\eta(\gamma_h), \dot{\gamma_h}, \dot{\gamma_h}] + D^2\eta[\ddot{\gamma_h}, D\eta(\gamma_h)] \\ &= \left(D^2\eta(\gamma_h)\left(I - D^2\eta(\gamma_h)\right)\right)[\dot{\gamma_h}, \dot{\gamma_h}] + \ddot{\gamma_h} \cdot D\eta(\gamma_h). \end{aligned}$$

The matrix $\Lambda(t) \overset{\triangle}{=} D^2\eta(\gamma_h(t))\left(I - D^2\eta(\gamma_h(t))\right)$ filters out normal components, and has eigenvalues associated with the tangential bundle given by (let $d(t) = d(\gamma_h(t), \mathcal{S})$),

$$\lambda_i(t) = \frac{d(t)\lambda_i(0)}{\left(1 + d(t)\lambda_i(0)\right)^2} \text{ for } 1 \leq i \leq k.$$

Note that $max_{1 \leq i \leq k}|\lambda_i(t)|$ can be bounded by $d(t)$ times a certain finite constant K' independent of h.

On the other hand, we can bound a.e. $|\ddot{\gamma}_h(t)|$ by a finite constant, say K, which takes into account the maximal curvature of all the boundaries $\partial\Omega^h_{\mathcal{S}}$, $0 < h < H$, but does not depend on h.

Putting all this together, we find (recall that $\|D\eta(x)\| = \sqrt{2\eta(x)} = d(x, \mathcal{S})$, see Appendix A.2):

$$\begin{aligned} \left(d_{\mathcal{S}}(x,y) - d_{\Omega^h_{\mathcal{S}}}(x,y)\right)^2 &\leq d_h \int_0^{d_h} \Lambda(t)[\dot{\gamma}_h, \dot{\gamma}_h]dt \\ &+ d_h \int_0^{d_h} \|\ddot{\gamma}_h\| \, \|D\eta(\gamma_h)\|dt \\ &\leq K' \max_{t \in [0,d_h]} d(t)\, d_h^2 + K \max_{t \in [0,d_h]} d(t) d_h^2. \end{aligned}$$

Now, remembering that d_h stands for $d_{\Omega^h_{\mathcal{S}}}(x,y)$, that $trace(\gamma_h) \subset \Omega^h_{\mathcal{S}}$, and defining $C = K + K'$, we arrive, with just simple additional work at the relations (4.1) or (4.2). □

Remark 10 *Note that, as the simple case of a circle in the plane shows, the rate of convergence is at most $C \cdot h$.*

We immediately obtain the following Corollary which will be useful ahead.

Corollary 4 *Let $p \in \mathcal{S}$, and $r \leq H$, then $B(p,r) \cap \mathcal{S} \subseteq B_{\mathcal{S}}(p, r(1 + C_{\mathcal{S}}\sqrt{r}))$.*

Proof:
Let $q \in B(p,r) \cap \mathcal{S}$, then by (4.2), $d_{\mathcal{S}}(p,q) \leq d_{\Omega^r_{\mathcal{S}}}(p,q)(1 + C_{\mathcal{S}}\sqrt{r})$. But $q \in B(p,r) \subset \Omega^r_{\mathcal{S}}$, then $d_{\Omega^h_{\mathcal{S}}}(p,q) = \|p - q\| \leq r$, what completes the proof. □

Definition 2 *([59]) We say that the compact manifold $\mathcal{S}$ with boundary $\partial\mathcal{S}$ is* strongly convex *if for every pair of points x and y in $\mathcal{S}$, there exists a unique minimizing geodesic joining them whose interior is contained in the interior of $\mathcal{S}$.*

Using basically the same procedure as in Theorem 7 with the convexity hypotheses above we can prove the following Corollary.

Corollary 5 *($\partial\mathcal{S} \neq \emptyset$) Under the hypotheses of Theorem 4, and assuming $\mathcal{S}$ to be strongly convex, we have for small enough $h > 0$ the same conclusions of Theorem 7 (rate of convergence).*

Proof:
Let $\mathcal{M}$ be an extension of $\mathcal{S}$ such that $\mathcal{S}$ is still strongly convex in $\mathcal{M}$ and let $0 < \delta \overset{\triangle}{=} \min_{x\in\mathcal{S}} \min_{z\in\mathcal{M}} \|x-z\|$. Then, $\overline{B(x,\alpha)} \cap \overline{B(z,\beta)} = \emptyset$ for all $x \in \mathcal{S}$, $z \in \partial\mathcal{M}$ and $\alpha, \beta < \frac{\delta}{3}$. Hence, $\Omega_{\mathcal{S}}^{\alpha} \cap \Omega_{\partial\mathcal{M}}^{\beta} = \emptyset$ for $\alpha, \beta \leq \frac{\delta}{3}$.

For any $x, y \in \mathcal{S}$ consider γ_h the $\Omega_{\mathcal{M}}^{h}$-minimizing geodesic, $\mathbf{L}(\gamma_h) = d_{\Omega_{\mathcal{M}}^{h}}(x,y)$.

By convexity of $\mathcal{S}$ there exists a unique $\mathcal{M}$-minimizing geodesic $\gamma_0 \subset \mathcal{S}$ joining x, y and then, by Theorem 6, γ_h uniformly converges to γ_0. In particular, for any $\epsilon > 0$ there exists $h_\epsilon > 0$ such that $\gamma_h \subset \Omega_{\gamma_0}^{\epsilon}$ for all $h < h_\epsilon$. Choose $\epsilon \leq \frac{\delta}{3}$ then $\gamma_h \subset \Omega_{\gamma_0}^{\epsilon} \subset \Omega_{\mathcal{S}}^{\epsilon}$. Furthermore, if $h \leq \frac{\delta}{3}$, then $\Omega_{\gamma_0}^{\epsilon} \cap \Omega_{\mathcal{M}}^{h} = \emptyset$ and therefore γ_h does not touch $\partial\Omega_{\mathcal{M}}^{h} \cap \partial\Omega_{\partial\mathcal{M}}^{h}$. Thus, γ_h is $C^{1,1}$ for $h \leq \frac{\delta}{3}$. Note that with this choice of h we have $\Omega_{\mathcal{S}}^{h} \cap \mathcal{M} \subset \text{int}(\mathcal{M})$ and therefore we also have a smooth orthogonal projection operator $\Pi : \Omega_{\mathcal{S}}^{h} \to \mathcal{M}$.

Proceeding as in the first steps of the proof of Theorem 7 we have $\mathbf{L}(\gamma_h) = d_{\Omega_{\mathcal{M}}^{h}}(x,y) \leq d_{\mathcal{M}}(x,y) \leq \mathbf{L}(\Pi(\gamma_h))$, since $\Pi(\gamma_h) \subset \mathcal{M}$ but may not be a minimizing path. Then, using the convexity of $\mathcal{S}$ in $\mathcal{M}$, $d_{\mathcal{M}}(x,y) = d_{\mathcal{S}}(x,y)$ and therefore $0 \leq d_{\mathcal{S}}(x,y) - d_{\Omega_{\mathcal{M}}^{h}}(x,y) \leq |\mathbf{L}(\Pi\gamma_h) - \mathbf{L}(\gamma_h)|$ which can be bounded by a constant times $\sqrt{h}$ just mimicking the proof of Theorem 7. We conclude by noting that $\Omega_{\mathcal{S}}^{h} \subset \Omega_{\mathcal{M}}^{h}$ hence $d_{\mathcal{S}}(x,y) - d_{\Omega_{\mathcal{M}}^{h}}(x,y) \geq d_{\mathcal{S}}(x,y) - d_{\Omega_{\mathcal{S}}^{h}}(x,y)$. □

Remark 11 *Note that in case $\partial\mathcal{S} \neq \emptyset$ is not strongly convex, then obviously the same statement of Corollary 5 remains valid for any strongly convex subset of $\mathcal{S}$.*

To conclude, in this chapter we extended the results in Chapter 2 to geodesics and distance functions in general codimension manifolds with or without (smooth) boundary, thereby covering all possible manifolds in common shape, graphics, visualization, and learning applications. We are now ready to extend this to manifolds represented as point clouds.

4.2.1 Distance Functions on Point Clouds

We are now interested in making distance and geodesic computations on manifolds represented as point clouds, i.e. *sampled manifolds*.

Let h, h' and $\mathcal{P}_n$ be such that $\mathcal{S} \subseteq \Omega_{\mathcal{P}_n}^{h}$ and $\mathcal{P}_n \subseteq \Omega_{\mathcal{S}}^{h'}$ and $\max(h, h') \leq H$. Note that h' represents a level of noise present in the sampling.

We then have $\mathcal{S} \subseteq \Omega_{\mathcal{P}_n}^{h} \subseteq \Omega_{\mathcal{S}}^{h+h'}$. We want to consider $d_{\Omega_{\mathcal{P}_n}^{h}}(p,q)$ for any pair of points $p, q \in \mathcal{S}$ and prove some kind of proximity to the real distance $d_{\mathcal{S}}(p,q)$. The argument carries over easily since $d_{\Omega_{\mathcal{S}}^{h+h'}}(p,q) \leq d_{\Omega_{\mathcal{P}_n}^{h}}(p,q) \leq d_{S}(p,q)$, hence $0 \leq d_{\mathcal{S}}(p,q) - d_{\Omega_{\mathcal{P}_n}^{h}}(p,q) \leq d_{\mathcal{S}}(p,q) - d_{\Omega_{\mathcal{S}}^{h+h'}}(p,q)$, and the rightmost quantity can be bounded by $C_{\mathcal{S}}\,(h+h')^{1/2}$ (see §4.2) in the case that $\partial\mathcal{S}$ is either strongly convex or void. The key condition is $d_{\mathcal{H}}(\mathcal{S}, \mathcal{P}_n) \leq \hat{h}$ for some prespecified $\hat{h}$. In the noiseless case ($h' = 0$), the key condition is $\mathcal{S} \subset \Omega_{\mathcal{P}_n^h}$, something that can obviously be coped with using the compactness of $\mathcal{S}$.[5] We can then state the following:

Theorem 8 (Uniform Convergence for Noiseless Point Clouds) *Let $\mathcal{S}$ be a compact smooth submanifold of $\mathbb{R}^d$ possibly with boundary $\partial\mathcal{S}$. Then*

[5] By compactness, given $h > 0$ we can find finite $N(h)$ and points $p_1, p_2, \ldots, p_{N(h)} \in \mathcal{S}$ such that $\mathcal{S} = \cup_{i=1}^{N(h)} B_{\mathcal{S}}(p_i, h)$. But since for $p \in \mathcal{S}$, $B_{\mathcal{S}}(p,h) \subset B(p,h) \cap \mathcal{S}$, and we also get $\mathcal{S} \subset \cup_{i=1}^{N(h)} B(p_i, h)$.

1. **General Case**: *Given $\varepsilon > 0$, there exists $h_\varepsilon > 0$, such that $\forall\ 0 < h \le h_\varepsilon$ one can find finite $n(h)$ and a set of points $\mathcal{P}_{n(h)}(h) = \{p_1(h), \ldots, p_{n(h)}(h)\}$ sampled from $\mathcal{S}$ such that*

$$\max_{p,q \in \mathcal{S}} \left(d_{\mathcal{S}}(p,q) - d_{\Omega^h_{\mathcal{P}_{n(h)}(h)}}(p,q) \right) \le \varepsilon$$

2. **$\partial\mathcal{S}$ is either void or convex:** *For every sufficiently small $h > 0$ one can find finite $n(h)$ and a set of points $\mathcal{P}_{n(h)}(h) = \{p_1(h), \ldots, p_{n(h)}(h)\}$ sampled from $\mathcal{S}$ such that*

$$\max_{p,q \in \mathcal{S}} \left(d_{\mathcal{S}}(p,q) - d_{\Omega^h_{\mathcal{P}_{n(h)}(h)}}(p,q) \right) \le C_{\mathcal{S}} \sqrt{h}$$

In practise, one must worry about both the number (n) of points and the radii (h) of the balls. Obviously, there is a tradeoff between these quantities. If we want to use only few points, in order to cover $\mathcal{S}$ with the balls we have to increase the value of the radius. Clearly, there exists a value H such that for values of h smaller than H we don't change the topology, see [7, 57, 81]. This implies that the number of points must be larger than a certain lower bound. This result can be generalized to ellipsoids which can be locally adapted to the geometry of the point cloud.

We now consider the general case of (maybe noisy) random sampling.

4.2.2 Random Sampling of Manifolds

We have to define the way in which we are going to measure accuracy. A possibility for such a measure is (for each $\varepsilon > 0$)
$\mathbb{P}\left(\max_{p,q\in\mathcal{S}} \left(d_{\mathcal{S}}(p,q) - d_{\Omega^h_{\mathcal{P}_n}}(p,q) \right) > \varepsilon \right)$. Notice that we are somehow considering $d_{\Omega^h_{\mathcal{P}_n}}$ to be defined for all pairs of points in $\mathcal{S} \times \mathcal{S}$, even if it might happen that $\mathcal{S} \cap \Omega^h_{\mathcal{P}_n} \ne \mathcal{S}$. In any case, we extend $d_{\Omega^h_{\mathcal{P}_n}}$ to all of $\mathbb{R}^d \times \mathbb{R}^d$ by a large constant say $K \cdot \mathbf{diam}(\mathcal{S})$, $K \gg 1$.

Let us define the events

$$\mathcal{E}_\varepsilon \triangleq \left\{ \max_{p,q\in\mathcal{S}} \left(d_{\mathcal{S}}(p,q) - d_{\Omega^h_{\mathcal{P}_n}}(p,q) \right) > \varepsilon \right\} \text{ and } \mathcal{I}_{h,n} \triangleq \left\{ \mathcal{S} \subseteq \Omega^h_{\mathcal{P}_n} \right\} \cap \left\{ \mathcal{P}_n \subseteq \Omega^h_{\mathcal{S}} \right\}.$$

Now, since $\mathcal{E}_\varepsilon = (\mathcal{E}_\varepsilon \cap \mathcal{I}_{h,n}) \cup (\mathcal{E}_\varepsilon \cap \mathcal{I}^c_{h,n})$, using the union bound and then Bayes rule we have

$$\mathbb{P}(\mathcal{E}_\varepsilon) \le \mathbb{P}(\mathcal{E}_\varepsilon \,|\mathcal{I}_{h,n}) + \mathbb{P}\left(\mathcal{I}^c_{h,n}\right). \tag{4.3}$$

It is clear now that we should use a convenient upper bound for the second term in the previous expression. The first term can be easily dealt with using the convergence theorems presented in previous sections.

Combining the preceding discussion with the results in Chapter 3 we obtain the following convergence theorem, where the dependence of ν with n (written as ν_n) means that there is a noise level present, which we require to vanish as $n \uparrow \infty$ in order to recover the true geodesic distance. In the noiseless case, the support of ν is $\mathcal{S}$ and therefore $\nu_n = \nu$ for all n.

Theorem 9 *Let $\mathcal{S}$ be a k-dimensional smooth compact submanifold of $\mathbb{R}^d$. Let $\mathcal{P}_n = \{p_1, \ldots, p_n\} \subseteq \mathbb{R}^d$ be an i.i.d. set of points such that $p_i \sim \nu_n$ for $1 \le i \le n$. Then if $h = h_n$ and ν_n are such that $h_n \downarrow 0$,*

$f_{\nu_n}(\frac{h_n}{2}) \gtrsim \frac{\ln n}{n}$ and $|1 - g_{\nu_n}(h_n)| \lesssim \frac{1}{n^{1+\alpha}}$ for some $\alpha > 0$ hold as $n \uparrow \infty$, we have that for any $\varepsilon > 0$, $\mathbb{P}\left(\mathcal{E}_\varepsilon\right) \overset{n\uparrow\infty}{\longrightarrow} 0$.

Proof:
We base our proof on equations ((3.6) and (4.3). We first note that $\mathbb{P}\left(\mathcal{E}_\varepsilon \,|\, \mathcal{I}_{h,n}\right) = 0$ for n large enough because, from considerations at the beginning of §4.2.1,
$\max_{p,q\in\mathcal{S}}\left(d_{\mathcal{S}}(p,q) - d_{\Omega_{\mathcal{P}_n}^{h_n}}(p,q)\right) \leq C_{\mathcal{S}}\sqrt{2h_n}$ whenever $d_{\mathcal{H}}(\mathcal{S},\mathcal{P}_n) \leq h_n$ holds. Let $N = N(\varepsilon) \in I\!N$ be such that $h_n < \frac{1}{2}\left(\frac{\varepsilon}{C_{\mathcal{S}}}\right)^2$ for all $n \geq N(\varepsilon)$. Then, for $n \geq N(\varepsilon)$, $\mathbb{P}\left(\mathcal{E}_\varepsilon\right) \leq \mathbb{P}\left(\mathcal{I}_{h,n}^c\right) \leq \frac{e^{-nf_\nu^{h_n}}}{f_\nu^{h_n}} + 1 - (g_{\nu_n}(h_n))^n \lesssim \frac{1}{\log n} + \frac{n}{n^{1+\alpha}}$, and the right hand side goes to 0 as $n \uparrow \infty$. □

Noiseless Sampling

The following remarks are valid for the noiseless case.

Remark 12 *1. As can be gathered from the preceding proof, for fixed $\varepsilon > 0$ and large $n \in I\!N$, $\mathbb{P}\left(\mathcal{E}_\varepsilon\right)$ can be upper bounded by $\frac{e^{-nf_\nu^{h_n}}}{f_\nu^{h_n}}$. For example, setting $f_\nu^{h_n} = \gamma\frac{\log n}{n}$ for $\gamma \geq 1$ yields (n big enough)*

$$\mathbb{P}\left(\mathcal{E}_\varepsilon\right) \leq \frac{1}{\gamma\, n^{\gamma-1}\log n}. \tag{4.4}$$

2. Then we see that by requiring $\sum_{n\geq 1}\frac{e^{-nf_\nu^{h_n}}}{f_\nu^{h_n}} < \infty$ and using the Borel-Cantelli Lemma we obtain almost sure convergence*, namely:*

$$\mathbb{P}\left(\lim_{n\uparrow\infty}\max_{p,q\in\mathcal{S}}\left(d_{\mathcal{S}}(p,q) - d_{\Omega_{\mathcal{P}_n}^{h_n}}(p,q)\right) = 0\right) = 1.$$

This can be guaranteed (for example) by setting $f_\nu^{h_n} = \gamma\frac{\log n}{n}$ for $\gamma > 2$.

Perhaps the following simple observation is of more practical value:

Remark 13 *Given $h > 0$, $p \in (0,1)$ and $\varepsilon \in (0, C_{\mathcal{S}}\sqrt{h})$, if (3.2) holds then $\mathbb{P}\left(\mathcal{E}_\varepsilon\right) \geq p$.*

Noisy Sampling of Manifolds

We now elaborate on a couple of noisy models for the sampling and derive some rate estimates based on Remark 12 and equation (3.6):

- $p_i \sim \mathbf{U}[\Omega_{\mathcal{S}}^{\Delta_n}]$. In this case, assuming $r \geq \Delta_n$ we obtain $f_\nu(r) \geq \frac{\mathbf{a}(B(\cdot,\Delta_n))}{\mathbf{a}\left(\Omega_{\mathcal{S}}^{\Delta_n}\right)}$ and $g_\nu(r) = 1$ since $B(\cdot,\Delta_n) \subset B(\cdot,r)$. Moreover, using Weyl's Tube Theorem, (see [85]) we find an explicit formula for the lower bound: $f_\nu(r) \geq \frac{\omega_d\Delta_n^d}{\mathbf{a}(\mathcal{S})\Delta_n^{d-k}+\kappa(\Delta_n)}$ where $\kappa(\cdot)$ is a higher order term. Hence, (4.4) holds if we set $\frac{\omega_d\Delta_n^k}{\mathbf{a}(\mathcal{S})+\frac{\kappa(\Delta_n)}{\Delta_n^{d-k}}} \simeq \gamma\frac{\log n}{n}$, and $h_n \geq 2\Delta_n$. Note that as h_n vanishes, the condition becomes $\Delta_n^k \simeq \gamma'\frac{\log n}{n}$ for some constant γ'.

- $p_i = u + \zeta \vec{n}_u$ where $u \sim \mathbf{U}[\mathcal{S}]$ and $\zeta \sim \mathbf{E}(0, \beta_n)$, u and ζ are independent and where $\vec{n}_t$ is unit norm and uniformly distributed in the normal space to $\mathcal{S}$ at the point t.[6] Note that since $\{u \in \mathcal{S}, |\zeta| \leq r\} \subseteq \{u + \zeta \vec{n}_u \in \Omega^r_{\mathcal{S}}\}$, then $g_{\nu_n}(r) \geq 1 - e^{-\beta_n r}$. Then, in order to satisfy $|1 - g_{\nu_n}(h_n)| \lesssim \frac{1}{n^{1+\alpha}}$ we can ask for the following condition (1): $\beta_n \gtrsim (1+\alpha)\frac{\log n}{h_n}$ to hold. Consider, for $z \in \mathcal{S}$ and $r > 0$ the set $\mathcal{C}_{z,r} = \{y \in I\!R^d \; y = t + w \text{ where } t \in B_{\mathcal{S}}(z, r/2) \text{ and } w = C\vec{n}_t, 0 \leq C \leq r/2\}$. It is then easy to check that $\mathcal{C}_{z,r} \subset B(z,r)$. Hence, $\mathbb{P}\left(p_i \in B(z,r)\right) \geq \mathbb{P}\left(p_i \in \mathcal{C}_{z,r}\right) \geq \mathbb{P}\left(u \in B_{\mathcal{S}}(z, r/2)\right) \mathbb{P}\left(|\zeta| \leq r/2\right)$ and therefore
$$f_\nu(r) \geq \min_{z \in \mathcal{S}} \frac{\mathbf{a}\left(B_{\mathcal{S}}(z, r/2)\right)}{\mathbf{a}\left(\mathcal{S}\right)} (1 - e^{-h_n \beta_n}).$$
Now, assuming condition (1) holds, we find $f_{\nu_n}(\frac{h_n}{2}) \gtrsim \omega h_n^k (1 - \frac{c}{n^{1+\alpha}})$ for some constants c and ω, which tells us that for n large enough, $f_{\nu_n}(\frac{h_n}{2}) \gtrsim \omega' h_n^k$. We then see that we could still impose, as in Remark 7, that $h_n^k \simeq C \frac{\log n}{n}$. The resulting restriction for β_n is $\beta_n \gtrsim (\log n)^{1-1/k} n^{1/k}$.

Note that although the results in this and in previous sections were presented for Euclidean balls of the same radius, this can easily be extended to more general covering shapes, e.g. following [46], or using minimal spanning trees still for balls but with different radii , or from the local directions of the data [162]. The band itself can be computed in several ways, and for the examples below we have used constant radii. Locally adaptive radii can be used, based for example on diameters obtained from minimal spanning trees. Automatic and local estimation of h defining $\Omega^h_{\mathcal{P}_n}$ was no pursued and is the subject of current research, we are studying a multiscale approach.

4.3 Implementation Details and Examples

We now present examples of distance matrices and geodesics for point clouds, Figure 4.1; use these computations to find intrinsic Voronoi diagrams, Figure 4.2 (see also [123, 131]); and compare the results with those obtained with mesh-based techniques, Figure 4.3. We also present examples in high dimensions. These exercises intend to exemplify the importance of computing distance functions and geodesics on point clouds, and are by no means exhaustive. The 3D data sets used come from real point cloud data, and have been obtained from range scanners (*David* model).

The theoretical results presented in previous sections show that the intrinsic distance and geodesics can be approximated by the Euclidean ones computed in the band defined (for example) by the union of balls centered at the points of the cloud. The problem is then simplified to first computing this band (no need for mesh computation of course), and then use well known computationally optimal techniques to compute the distances and geodesics inside this band, exactly as done in Chapter 2 for implicit surfaces. The band itself can be computed in several ways, and for the examples below we have used constant radii. Locally adaptive radii can be used, based for example on diameters obtained from minimal spanning trees or on the recent work reported in [164]. Automatic and local estimation of h defining $\Omega^h_{\mathcal{P}_n}$, which will improve the bounds here reported, was not pursued in this chapter and is the subject of current implementation efforts.

The software implementation of the algorithm is based on using the fast Euclidean distance computation algorithms, usually referred to as *fast marching* algorithm [99, 178, 179, 199], twice. This algorithm has been described in Chapter 2. The starting point is defining a grid over which all the computations are performed.

[6] $\mathbf{E}(0, \beta_n)$ denotes the Exponential distribution with zero mean and parameter β_n: $\mathbb{P}\left(\zeta \in [a,b]\right) = \int_a^b \frac{\beta_n}{2} e^{-\beta_n |z|} \, dz$.

This amounts to choosing Δ_{x_i}, the grid spacing in each direction $i = 1, \ldots, d$, which will determine the accuracy of the numerical implementation (the offset band includes less than 10 grid points).[7] In the first round we compute the band $\Omega^h_{\mathcal{P}_n} = \{x \in I\!R^d : d(\mathcal{P}_n, x) \leq h\}$ by specifying a value of zero for the function $\Psi(x) = d(\mathcal{P}_n, x)$ on the points $x \in \mathcal{P}_n$. Since in general this points will not be on the grid, we use a simple multilinear interpolation procedure to specify the values on neighboring grid points. The second use of the fast distance algorithm is also simply reduced to using Ψ to define $\Omega^h_{\mathcal{P}_n}$. The computation of geodesics was done using a simple Runge-Kutta gradient descent procedure, much in the way described in Chapter 2, with some obvious modifications.

All the code and 3D visualization was developed in C++ using both Flujos (which is written using Blitz++, see [74]) and VTK (see [202]). For matrix manipulation and visualization of other results we used matlab. We are currently working on a more advanced implementation of the proposed framework that permits to work with high dimensional data without having memory allocation problems that result from blind and straightforward allocation of resources to empty and non-used grids.

4.3.1 High Dimensional Data

In this section we present a simple example for high dimensional data. We embedded a circle of radius 15 in $I\!R^5$, and use a grid of size $34 \times 4 \times 4 \times 4 \times 34$ (with uniform spacing $\Delta x = 1$) such that each of the sample points is of the form $p_i = 15\left(\cos(\frac{2\pi i}{N}), 0, 0, 0, \sin(\frac{2\pi i}{N})\right) + (17, 2, 2, 2, 17)$, for $1 \leq i \leq N$. We then used our approach to compute the (approximate) distance function d_h in a band in $I\!R^5$, and then, the error $e_{ij} = |d_{\mathcal{S}}(p_i, p_j) - d_h(p_i, p_j)|$ for $i, j \in \{1, \ldots, N\}$. In our experiments we used $h = 2.5 > \Delta x\sqrt{5}$. We randomly sampled 500 points from the $N = 1000$ points used to construct the union of balls to build the 500×500 error matrix $((e_{ij}))$. We found $max_{ij}\{e_{ij}\} = 2.0275$, that is a 4.3% L_∞-error. In Figure 4.4 we show the histogram of all the (500^2) entries of $((e_{ij}))$. We should also note that when following the dimensionality reduction approach in [189], with the geodesic distance computation here proposed, the correct dimensionality of the circle was obtained.

In high dimensions, when the grid is too large, our current numerical implementation becomes unusable. The problem stems from the fact that we require too much memory space, most of which is not really used, since the computations are conducted only in a band around $\mathcal{P} \subset I\!R^d$. To be more precise, the memory requirements of our current direct implementation, which uses a d-dimensional array to make the computations, are $\simeq (\max_i l_i)^d$, whereas, we really need a storage capacity of order $\mu_k(\mathcal{S})h^{d-k}$, where l_i is the size of $\mathcal{P}$s bounding box along the i'th direction, $1 \leq i \leq d$, and $\mu_k(\mathcal{S})$ is the measure of the k-dimensional manifold $\mathcal{S}$ (embedded in $I\!R^d$). This memory problem is to be addressed by a computation that is not based on discretizing the whole band (note of course that the theoretical foundations presented in this chapter are independent of the particular implementation). We are currently working on addressing this specific issue.

For further examples of our framework see [142] and for other applications [150] (for point cloud simplification) and [151] (for meshless subdivision of point clouds).

[7] Adaptive grids inside the fix or variable width offset band could be used as well, see for example [79].

Comparison with Mesh-based Strategies for Distance Calculation in Presence of Noise

We now make some very basic comparisons between our approach to geodesic distance computations and those based on graph approximations to the manifold, such as the one in Isomap [189, 81].[8] The goal is to show that such graph-based techniques are more sensitive to noise in the point cloud sample (and the error can even increase to infinity with the increase in the number of points). This is expected, since the geodesic in such techniques goes through the noisy samples, while in our approach, they just go through the union of balls. We only make our argument for the $1D$ case, while the high dimensional cases can be similarly studied.

1D Theoretical Case

Let's consider a rectilinear segment of length L and $n+1$ equi-spaced points $p_1, \ldots, p_{n+1}$ in that segment. Consider the *noisy* points $q_i = p_i + \zeta_i \vec{n}$ where $\vec{n}$ is the normal to the segment and ζ_i $1 \leq i \leq n$ are independent RV uniformly distributed in $[-\Delta, \Delta]$. Let $l = L/n$ denote the distance between adjacent p_is. Let d_g^Δ denote the length of the polygonal path $\overline{q_1 q_2 \ldots q_{n+1}}$ and $d_0 = L$. Then obviously $d_g^\Delta \geq d_0$ for any realization of the RVs ζ_i. Let $d_i = \|p_i - p_{i+1}\|$, then by Pythagoras theorem $d_i = \sqrt{l^2 + z_i^2}$, where $z_i = \zeta_i - \zeta_{i+1}$ are RVs with triangular density in $[-2\Delta, 2\Delta]$.

Then we compute $\mathbb{E}\left(d_i\right) = \frac{1}{2\Delta}\int_{-2\Delta}^{2\Delta} \sqrt{l^2+z^2}(1-\frac{|z|}{2\Delta})\, dz$. The result is

$$\mathbb{E}\left(d_i\right) = \sqrt{l^2+4\Delta^2} + \frac{l^2}{2\Delta}\log\left(\frac{2\Delta + \sqrt{l^2+4\Delta^2}}{l}\right) - \frac{1}{6\Delta^2}\left((l^2+4\Delta^2)^{3/2} - l^3\right).$$

Now assuming $\frac{\Delta}{l} \ll 1$, we find that up to first order $\mathbb{E}\left(d_i\right) \simeq l + \Delta$, and

$$\mathbb{E}\left(d_g^\Delta - d_0\right) \simeq n\Delta.$$

From this we also get[9]

$$p_g \triangleq \mathbb{P}\left(d_g^\Delta - d_0 > \varepsilon\right) \lesssim \frac{n\Delta}{\varepsilon}.$$

On the other hand, for our approximation, d_h^Δ, if the segment is contained in the union of the balls centered at the sampling points, $d_h^\Delta = d_0$. The probability of covering the segment by the band can be made arbitrarily close to 1 by increasing n. More precisely, one can prove that if p stands for the value of the probability of *not covering* the segment, then $p \leq k\frac{L}{\Delta}(1-k'\frac{\Delta}{L})^n$, for some positive constants k and k'. Then, we can write

$$p_h \triangleq \mathbb{P}\left(d_h^\Delta - d_0 > \varepsilon\right) \leq \frac{k''}{\varepsilon}\frac{L}{\Delta}(1-k'\frac{\Delta}{L})^{n+1}.$$

The comparison is now easy. We see that in order to have p_g vanish as $n \uparrow \infty$, Δ must go to zero *faster* than $\frac{1}{n}$. Whereas, we know that by requiring $\Delta \simeq \frac{\log n}{n} \gtrsim \frac{1}{n}$ we have $p_h \downarrow 0$ as $n \uparrow \infty$. This means that the graph approximation of the distance is more sensitive to noise than ours.[10] This gives some evidence on why

[8] Isomap builds a mesh by locally connecting the (noisy) samples.

[9] Also, with similar arguments we can prove that $\max_{\zeta_1,\ldots,\zeta_{n+1}}\left(d_g^\Delta - d_0\right) \simeq \frac{2n^2\Delta^2}{L}$.

[10] Another way of seeing this is noting that for a fixed noise level Δ, by increasing n we actually worsen the graph approximation, whereas we are making our approximation better.

our approach is more robust than popular mesh-based ones. Next we present results of some simulations carried in order to further verify our claim.

Simulations

In the table below we present results of simulations carried out for the SwissRoll dataset [189], see Figure 4.3 . We used 10000 points to define the manifold. We then generated 10000 noise vectors, being each component uniform with power one and zero mean. Then, we generated noisy datasets from the noiseless SwissRoll dataset by adding the noise vector times a constant n_k to each vector of the noiseless initial dataset. We then chose 1000 corresponding points in each dataset and computed the intrinsic pairwise distance approximation obtaining the matrices $\{(D_{ij}^{g,n_k})\}$ and $\{(D_{ij}^{h,n_k})\}$ for the graph-based and our approach respectively, where $k = 1, 2, \dots, 5,\ i, j \in [1, 1000]$, and n_k denotes the noise level. We then computed the values of $max_{ij}|D_{ij}^{g,n_k} - D_{ij}^{g,0}|$ and $max_{ij}|D_{ij}^{h,n_k} - D_{ij}^{h,0}|$ for each k, where $D_{ij}^{g,0}$ and $D_{ij}^{h,0}$ stand for noiseless intrinsic distance approximations. In the table below, h indicates the radii and k the size of the neighborhood for Isomap. The graph approximation shows less robustness to noise than our method, as was argued above. This is also true for the sensitivity,[11] where our approach outperforms the graph-based one by at least one order of magnitude. Note that the sensitivity for our approach can be formally studied from Theorem 5.

Noise Power (n_k^2)	$max_{ij}\|D_{ij}^{g,n_k} - D_{ij}^{g,0}\|$	**k**	$max_{ij}\|D_{ij}^{h,n_k} - D_{ij}^{h,0}\|$	**h**
0.0001	2.5222	7	0.5266	1.8
0.01	4.6409	7	0.9430	1.8
0.04	5.1737	7	1.2489	1.8
0.09	5.3292	7	1.4682	1.8
0.16	5.4651	7	1.7965	1.8

[11] Sensitivity is defined as $\left|1 - \frac{\text{distance for noisy points}}{\text{distance for clean points}}\right|$.

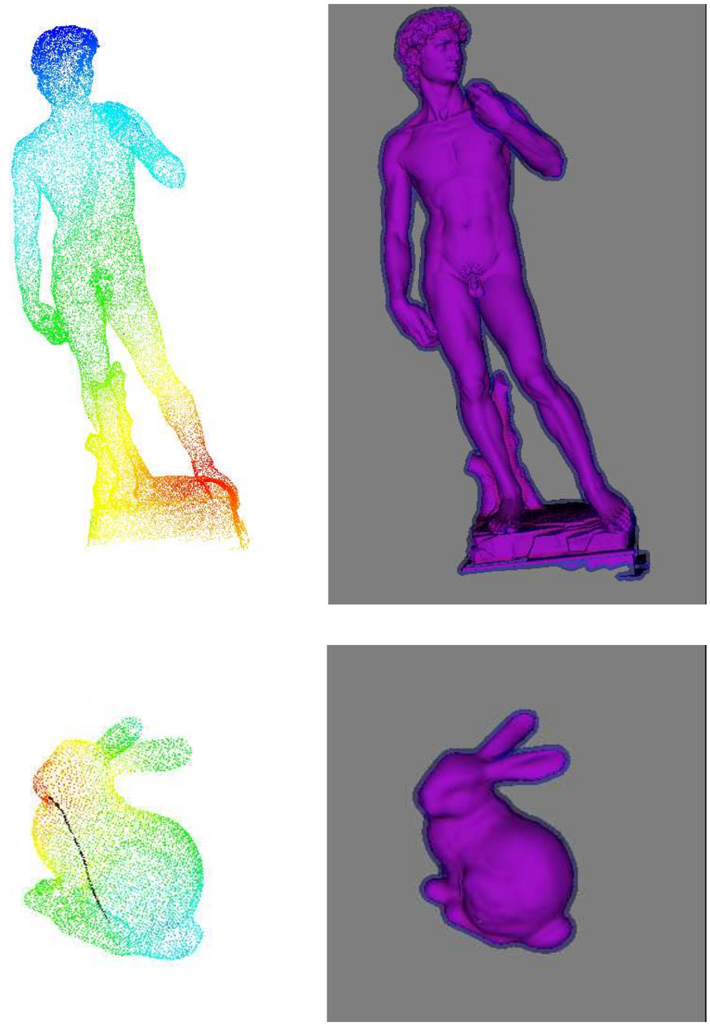


Figure 4.1: *Top: Intrinsic distance function for a point cloud. A point is selected in the head of the David, and the intrinsic distance is computed following the framework here introduced. The point cloud is colored according to their intrinsic distance to the selected point, going from bright red (far) to dark blue (close). The offset band, given by the union of balls, is shown next to the distance figure. Bottom: Same as before, with a geodesic curve between two selected points.*

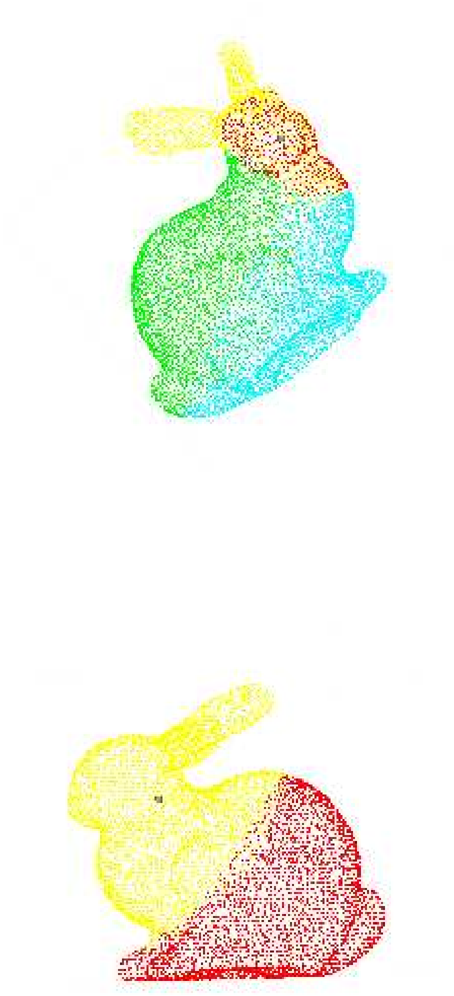


Figure 4.2: *Voronoi diagram for point clouds. Four points (left) and two points (right) are selected on the cloud, and the point cloud is divided (colored) according to their geodesic distance to these four points. Note that this is a surface Voronoi, based on geodesics computed with our proposed framework, not an Euclidean one.*

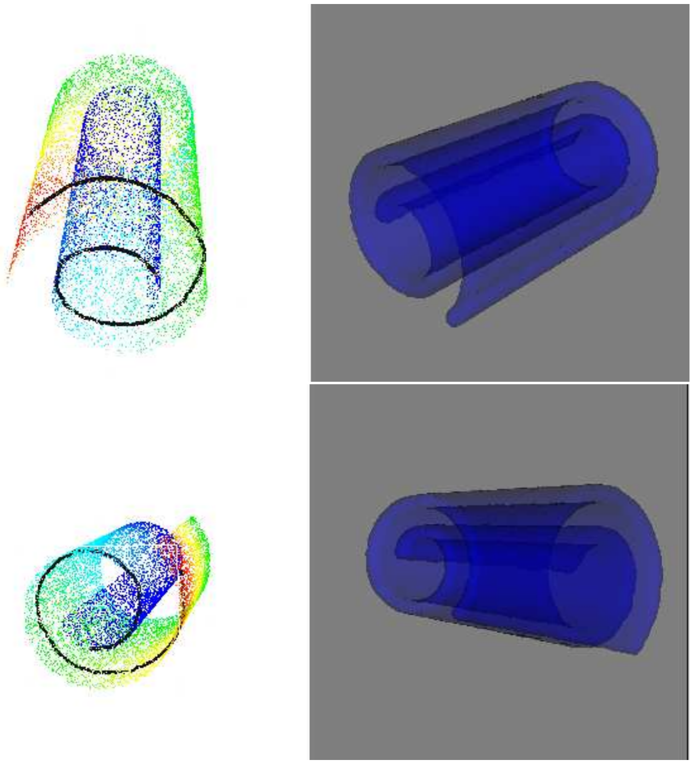


Figure 4.3: *Examples of geodesic computations. This data is used to study the algorithm robustness to noise, see §4.3.1.*

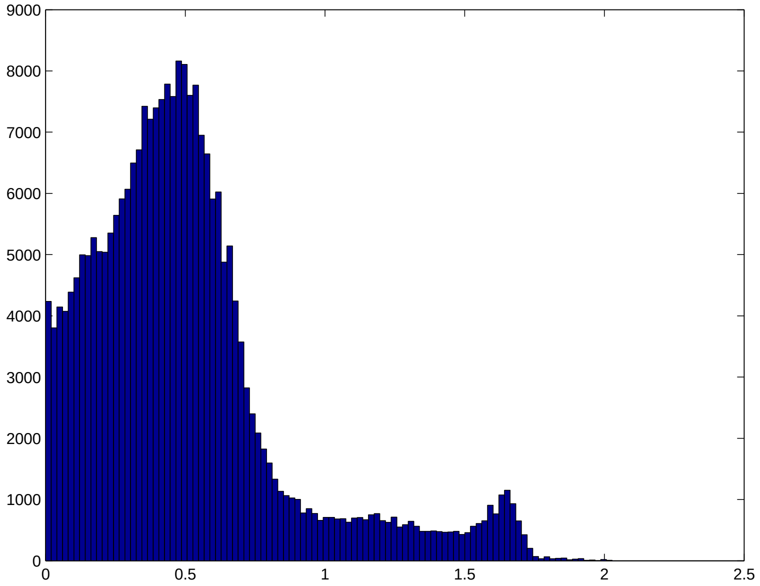


Figure 4.4: *Histogram for the error in the case of a circle embedded in* $\mathbb{R}^5$.

Chapter 5

Comparison of Point Clouds

5.1 Introduction

The goal of this work, inspired in part by [66] and the tools developed in Chapter 4 and [189], is to develop a theoretical and computational framework to compare shapes represented as point clouds. We are then going to assume the existence of an underlying structure from which our point cloud data are obtained through a sampling/acquisition process. Also, eventually we introduce the further assumption that the underlying structures we want say things about all belong to a family or class of objects which satisfy certain tameness properties.

As we have mentioned in the Introduction to this thesis (§1.1), a variety of objects can be represented as point clouds in $\mathbb{R}^d$. One is often presented with the problem of having to decide whether two of those point clouds, and/or the corresponding underlying objects or manifolds, represent the same geometric structure or not (*object recognition and classification*). We are then concerned with questions about the underlying unknown structures (objects), which need to be answered based on discrete measures taken between their respective point clouds. In greater generality, we may wonder what is the structural information we can gather about the object itself by exploring a point cloud which represents it.[1]

Multidimensional scaling (MDS),[2] for example, has been used to partly approach this general problem of object analysis/recognition, by means of checking whether the underlying space (object) is flat or not, and also providing information about the object's dimensionality (as a subset of $\mathbb{R}^d$) and its projection into a reduced space. Procedures based on MDS require that one first computes the inter-point distance matrix for all the members of the point cloud (or for a representative selected sub-set of them). If one is interested in comparing two different objects, the problem is reduced to a comparison between the corresponding inter-point distance matrices of their point clouds. If the distance we use is the Euclidean one, these matrices only provide information about their *rigid similarity*, and (assuming the matrices are of the same size) if they are equal (up to a permutations of the indices of all elements),[3] we can only conclude that there exists

[1] A related important question is what conditions must a point verify in order to faithfully represent an object, not to mention that one must ascribe a meaning to the word *faithfully*.

[2] For Multimensional Scaling, see for example [21].

[3] Boutin and Kemper, [25], have approached the recognition problem for (discrete objects) by looking only at the histogram of inter-point squared Euclidean distances. Interestingly, they showed that while there are counterexamples for the recognition problem with this kind of input, they constitute a very small fraction of all the possible point configurations. Such histograms

a rigid isometry (rotation, reflection, translation) from one point cloud to the other. Under assumptions of compactness we can also say something about the true underlying objects. Being more precise, let the point clouds $\mathcal{P}_i \subset S_i$ be ϵ_i-coverings of the compact surfaces $\mathcal{S}_i$ in $\mathbb{R}^3$, for $i = 1, 2$ (this will be formally defined below). Then assuming there exists a rigid isometry $\tau : \mathbb{R}^3 \to \mathbb{R}^3$ such that $\tau(\mathcal{P}_1) = \mathcal{P}_2$, we can bound the Hausdorff distance (which we will also formally define below) between $\tau(\mathcal{S}_1)$ and $\mathcal{S}_2$ as follows:

$$\begin{aligned} d_{\mathcal{H}}^{\mathbb{R}^3}(\tau(\mathcal{S}_1), \mathcal{S}_2) &\leq d_{\mathcal{H}}^{\mathbb{R}^3}(\tau(\mathcal{S}_1), \tau(\mathcal{P}_1)) + d_{\mathcal{H}}^{\mathbb{R}^3}(\tau(\mathcal{P}_1), \mathcal{P}_2) + d_{\mathcal{H}}^{\mathbb{R}^3}(\mathcal{P}_2, \mathcal{S}_2) \\ &= d_{\mathcal{H}}^{\mathbb{R}^3}(\mathcal{S}_1, \mathcal{P}_1) + d_{\mathcal{H}}^{\mathbb{R}^3}(\tau(\mathcal{P}_1), \mathcal{P}_2) + d_{\mathcal{H}}^{\mathbb{R}^3}(\mathcal{P}_2, \mathcal{S}_2) \\ &\leq \epsilon_1 + 0 + \epsilon_2 \end{aligned} \tag{5.1}$$

And of course the same kind of bound holds for the Hausdorff distance between the point clouds once we assume the underlying continuous objects are rigidly isometric, see §5.2.1 below.

One possible modification would be considering, still for compact surfaces, the *intrinsic* distance instead of the Euclidean (extrinsic) one for the construction of the aforementioned inter-point distance matrices. A comparison of these new distance matrices would then allow for more freedom in deciding when 2 objects are similar since now bends are allowed.

If $\mathcal{S}_1$ and $\mathcal{S}_2$ happen to be isometric (here also allowing for bends and not only rigid transformations) we wonder whether we will be able to detect this by looking at (finite) point clouds $\mathcal{P}_i$ sampled from each $\mathcal{S}_i$. This problem is harder to tackle. We approach it through a probabilistic model, since in principle there might exist even for the same object, two different samplings that look quite dissimilar (under discrete measures we can cope with computationally), for arbitrarily fine scales (see below).

With the help of the theory here presented we recast these considerations in a rigorous framework and address the case where the distances considered to characterize each point cloud (object) are more general. We concentrate on the case when there exists an intrinsic notion of distance for each object we sample. For the applications of isometry invariant shape (surfaces) recognition, one must therefore consider the distance as measured by paths constrained to travel on the surface of the objects, better referred to as *geodesic distance*.

These ideas have been introduced and used in [26, 66] for bending invariant recognition in $3D$ (without the theoretical foundations here introduced), see also [97]; and in [81, 189] to detect intrinsic surface dimensionality. The works [26, 66] argue in favor of invariance to full isometries in the case of face recognition. We have recently discovered the works of Patrizio Frosini and his collaborators at the University of Bologna who have developed interesting theoretical tools for the problem of comparing shapes, see [80] and references therein.

In this Chapter we introduce both a theoretical and computational framework for the so called *isometry invariant shape recognition problem.* The theory we use and build our framework upon is that pioneered by Gromov [87], in which a metric is introduced in the space of all (compact) metric spaces. For the sake of generality we present most of the framework for metric spaces, but the reader, at any moment, is invited to think of surfaces for simplicity. We will abuse of terminology in the following sense: Since we are dealing both with metric spaces and finite sets of samples from them, we are going to speak of continuous and

have been used earlier by the Princeton Shape Analysis Group, see `www.cs.princeton.edu/gfx/proj/shape/index.html`.

discrete metric spaces. For instance, given a metric space (X, d_X) we consider a finite subset of it, $\mathbb{X}_m \subset X$ which we endow with the metric of X to conform a *discrete* metric space, then X will be called *continuous* (we will use this nomenclature from now on). This is in analogy with the sampling of signals.

The fundamental approach used for isometry invariant recognition in this Chapter is derived then from the *Gromov-Hausdorff distance*, which we now proceed to present. Suppose X and Y are two (objects) compact subsets of a common bigger metric space (Z, d_Z), and we want to compare X to Y in order to decide whether they are/represent the same object or not. Then, an idea that one might come up with very early on is that of computing the *Hausdorff distance* between them (see for example [36, 103] for an extensive use of this for shape statistics and image comparison):

$$d_{\mathcal{H}}^{Z}(X,Y) \overset{\triangle}{=} \max(\sup_{x\in X} d_Z(x,Y), \sup_{y\in Y} d_Z(y,X)) \tag{5.2}$$

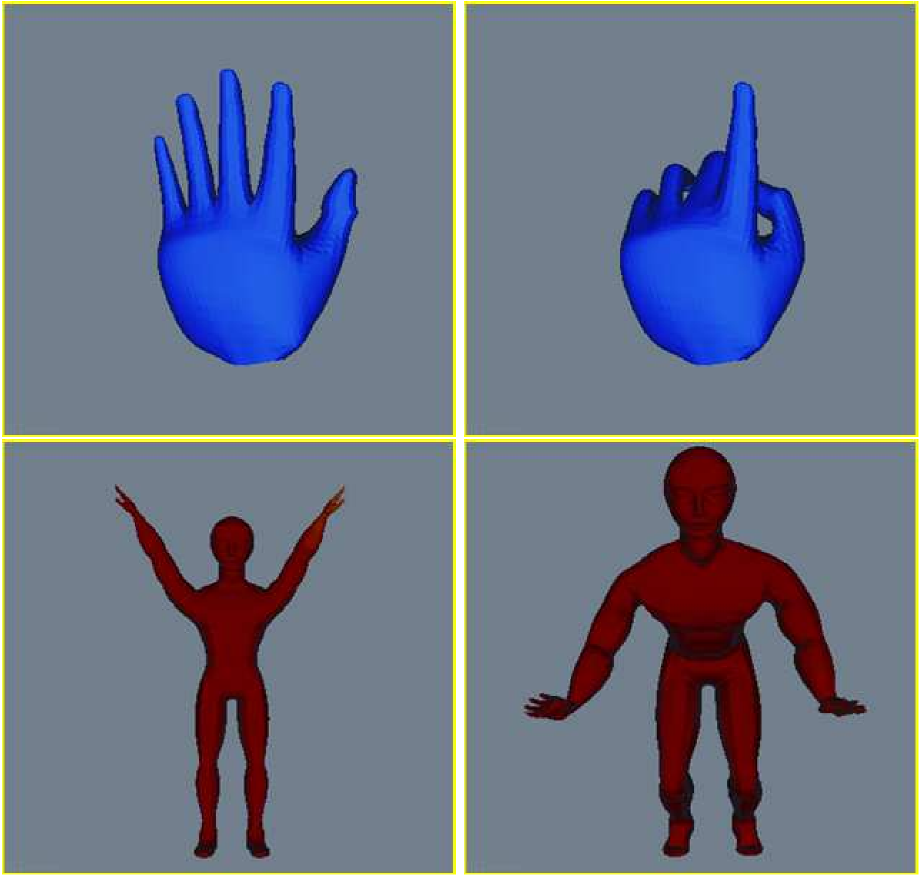


Figure 5.1: Two examples of general isometries.

But, what happens if we want to allow for certain deformations to occur and still decide that the objects are the same? More precisely, we are interested in being able to find a distance between metric spaces that is *blind* to isometric transformations ("bends"). This will permit a truly geometric comparison between the manifolds, independently of their embedding and bending position. Following [87], we introduce the *Gromov-Hausdorff distance* between Metric Spaces:

$$d_{\mathcal{GH}}(X,Y) \stackrel{\triangle}{=} \inf_{Z,f,g} d_{\mathcal{H}}^{Z}(f(X),g(Y)) \tag{5.3}$$

where $f : X \to Z$ and $g : Y \to Z$ are *isometric embeddings* (distance preserving) into the metric space Z. It turns out that this measure of metric proximity between metric spaces is well suited for our problem at hand and will allow us to give a formal framework to address the isometric shape recognition problem (for point cloud data). However, this notion of distance between metric spaces encodes the "metric" disparity between them, at first glance, in a computationally impractical way. We derive below new results that connect this notion of disparity with other more computationally appealing expressions.

Remark 14 *In [66] the authors proposed to use MDS applied to the geodesic distance matrices of each point cloud in order to obtain a new pair of point clouds in $\mathbb{R}^3$, such that the Euclidean distance matrices of these new point clouds resemble as well as possible (according to some criterion) the geodesic distance matrices between the original point clouds. The comparison then proceeds by computing some metric in $\mathbb{R}^3$ to measure the dissimilarity between the new point clouds. One could use, for example, the rigid-isometries invariant Hausdorff distance $d_{\mathcal{H}}^{\mathbb{R}^3,rigid}(\cdot,\cdot)$, see §5.2.1 ahead. This process can be rewritten in a more appealing way as follows. Let $\mathcal{P}_1 \subset \mathbb{R}^3$ and $\mathcal{P}_2 \subset \mathbb{R}^3$ be the original point clouds and $\mathcal{Q}_1 \subset \mathbb{R}^3$ and $\mathcal{Q}_2 \subset \mathbb{R}^3$ the corresponding new (projected) point clouds. Let also $\widehat{f} : \mathbb{R}^3 \to \mathbb{R}^3$ and $\widehat{g} : \mathbb{R}^3 \to \mathbb{R}^3$ be such that $\widehat{f}(\mathcal{P}_1) = \mathcal{Q}_1$ and $\widehat{g}(\mathcal{P}_2) = \mathcal{Q}_2$. Then, the number we compute is $d_{\mathcal{H}}^{\mathbb{R}^3,rigid}(\widehat{f}(\mathcal{P}_1),\widehat{g}(\mathcal{P}_2))$ which has an interesting resemblance with the formula in the definition of the Gromov-Hausdorff distance.*[4]

Since we have in mind specific applications and scenarios such as those described above, and in particular surfaces and sub-manifolds of some Euclidean space $\mathbb{R}^d$, we assume that we are given as input points *densely* sampled from the metric spaces (surfaces, manifolds). This will manifest itself in many places in the theory described below. We will present a way of computing a discrete approximation (or bound) to $d_{\mathcal{GH}}(,)$ based on the metric information provided by these point clouds.

The problem of isometry invariant shape recognition at hand can be split in two parts. Firstly, suppose the metric spaces under consideration happen to be isometric. We then have to guarantee that we can discover this by looking at a computable discrete measure of metric similarity based just on our observed data, that is, the point clouds. Secondly, if that measure of (discrete) metric similarity is "small," what can we say about that metric similarity between the underlying metric spaces? Both parts are addressed in our work. One cannot perform object recognition without either of them.

The rest of this Chapter is organized as follows: The basic theoretical foundations are given in Section §5.2, Section §5.3 presents the computational foundations, Section §5.4 illustrates the use of the framework with real examples, and finally Section §5.5 concludes the paper and describes current efforts and future directions.

We should note that this is a mainly theoretical work which proposes a framework (that leads to a possible practical algorithm) and that the examples provided in §5.4 are not exhaustive and do not make use of all the machinery here introduced, they simply exemplify and illustrate the application of the framework. More comprehensive experimentation is subject of current efforts and will be reported in the future.

[4] Of course, $\widehat{f}$ and $\widehat{g}$ are not isometries, in general.

5.2 Theoretical Foundations

This section covers the fundamental theory behind the bending invariant recognition framework we develop. We first introduce some basic notation, definitions, and classical results. We use basic concepts on metric spaces, see for example [110] for a simple exposition of this.

Definition 3 (Metric Space) *A set M is a metric space if for every pair of points $x, y \in M$ there is a well defined function $d_M(x, y)$ whose values are non-negative real numbers, such that (a) $d_M(x, y) = 0 \Leftrightarrow x = y$, and (b) $d_M(x, y) \le d_M(y, z) + d_M(z, x)$ for any x, y and $z \in M$. We call $d_M : M \times M \to I\!R$ the metric or distance. For clarity we will specify a metric space as the pair (M, d_M).*

Definition 4 (Covering) *For a point x in the metric space (X, d_X) and $r > 0$, we will denote by $B_X(x, r)$ the set $\{z \in X |\, d_X(x, z) < r\}$. For a subset A of X, we use the notation $B_X(A, r) = \cup_{a \in A} B_X(a, r)$. We say that a set $C \subset X$ is an R-covering of X if $B_X(C, R) = X$. We will also frequently say that the set A is a n-covering of X if A constitutes, for some $r > 0$, a covering of X by n-balls with centers in points of A.*

Definition 5 (Isometry) *We say the metric spaces (X, d_X) and (Y, d_Y) are isometric when there exists a bijective mapping $\Phi : X \to Y$ such that $d_X(x_1, x_2) = d_Y(\Phi(x_1), \Phi(x_2))$ for all $x_1, x_2 \in X$. Such a Φ is an isometry between (X, d_X) and (Y, d_Y).*

Next, we state some well known properties of the Gromov-Hausdorff distance $d_{\mathcal{GH}}(,)$ which will be useful for our presentation.

Proposition 3 *1. Let (X, d_X), (Y, d_Y) and (Z, d_Z) be metric spaces then*

$$d_{\mathcal{GH}}(X, Y) \le d_{\mathcal{GH}}(X, Z) + d_{\mathcal{GH}}(Z, Y).$$

2. If $d_{\mathcal{GH}}(X, Y) = 0$ and (X, d_X), (Y, d_Y) are compact metric spaces, then (X, d_X) and (Y, d_Y) are isometric.

*3. Let $\{x_1, \ldots, x_n\} \subset X$ be a R-covering of the compact metric space (X, d_X).
Then $d_{\mathcal{GH}}(X, \{x_1, \ldots, x_n\}) \le R$.*

4. For compact metric spaces (X, d_X) and (Y, d_Y):

$$\frac{1}{2} \left| \mathbf{diam}(X) - \mathbf{diam}(Y) \right| \le d_{\mathcal{GH}}(X, Y) \le \frac{1}{2} \max \left(\mathbf{diam}(X), \mathbf{diam}(Y) \right)$$

where $\mathbf{diam}(X) := \max_{x, x' \in X} d_X(x, x')$ stands Diameter of the metric space (X, d_X).

5. For bounded metric spaces (X, d_X) and (Y, d_Y),

$$d_{\mathcal{GH}}(X, Y) = \inf_{\substack{\phi : X \to Y \\ \psi : Y \to X}} \sup_{\substack{x_1, x_2 \in X \\ y_1, y_2 \in Y \\ (x_i, y_i) \in \mathcal{G}(\phi, \psi)}} \frac{1}{2} |d_X(x_1, x_2) - d_Y(y_1, y_2)|$$

where $\mathcal{G}(\phi, \psi) = \{(x, \phi(x)),\, x \in X\} \cup \{(\psi(y), y),\, y \in Y\}$ and the infimum is taken over all arbitrary maps $\phi : X \to Y$ and $\psi : Y \to X$.

The proofs of Properties 1 to 4 can be gleaned from [28, 87, 89, 168], and Property 5 can be found in [112]. Also of great informative value is [167].

Remark 15 *Note that Property 5 above can be recast in a somewhat more clear form: Let* $A(\phi) = \sup_{x_1,x_2 \in X} |d_X(x_1,x_2) - d_Y(\phi(x_1),\phi(x_2))|$*,* $B(\psi) = \sup_{y_1,y_2 \in Y} |d_X(\psi(y_1),\psi(y_2)) - d_Y(y_1,y_2)|$ *and* $C(\phi,\psi) = \sup_{x \in X,\, y \in Y} |d_X(x,\psi(y)) - d_Y(\phi(x),y)|$*, then*

$$d_{\mathcal{GH}}(X,Y) = \inf_{\substack{\phi\,:\,X \to Y \\ \psi\,:\,Y \to X}} \frac{1}{2} \max\left(A(\phi), B(\psi), C(\phi,\psi)\right) \tag{5.4}$$

It is interesting to note the following: Assume that $d_{\mathcal{GH}}(X,Y) \leq \eta$ *for small* η*, then roughly speaking, we can find* ϕ *and* ψ *such that (1)* ϕ *provides a low metric distortion map from* X *to* Y *(because* $A(\phi) \leq 2\eta$*), (2)* ψ *provides a low metric distortion map from* Y *to* X *(because* $B(\psi) \leq 2\eta$*), and (3)* ϕ *and* ψ *are "almost" inverses of one another (because* $C(\phi,\psi) \leq 2\eta$*, then taking* $y = \phi(x)$ *in the definition of* C *we find* $d_X(x,\psi(\phi(x))) \leq 2\eta$ *for all* $x \in X$*; and also, symmetrically,* $d_Y(y,\phi(\psi(y))) \leq 2\eta$ *for all* $y \in Y$*).*

Remark 16 *From Property 4 it follows that two metric spaces whose diameters differ must be at a positive* $d_{\mathcal{GH}}(,)$ *distance, as intuition requires.*

From these properties, we can also easily obtain the following important result:

Corollary 6 *Let* X *and* Y *be compact metric spaces. Let moreover* $\mathbb{X}_m$ *be a* r*-covering of* X *(consisting of* m *points) and* $\mathbb{Y}_{m'}$ *be a* r'*-covering of* Y *(consisting of* m' *points). Then*

$$|d_{\mathcal{GH}}(X,Y) - d_{\mathcal{GH}}(\mathbb{X}_m, \mathbb{Y}_{m'})| \leq r + r'$$

We can then say that if we could compute $d_{\mathcal{GH}}(,)$ for discrete metric spaces which are dense enough samplings of the "continuous" underlying ones, that number would be a good approximation to what happens between the continuous spaces. Currently, there is no computationally efficient way to directly compute $d_{\mathcal{GH}}(,)$ between discrete metric spaces in general. This forces us to develop a roundabout path, see §5.2.2 ahead. Before going into the general case, we discuss next the application of the ideas of our framework to a simpler but important case.

5.2.1 Intermezzo: The Case of Rigid Isometries

When we try to compare two (compact) subsets X and Y of a larger metric space Z, the situation is a bit simpler. The measure of similarity boils down to a somewhat simpler Hausdorff distance between the sets (which of course must take into account self-isometries of Z). In more detail, one must compute

$$d_{\mathcal{H}}^{Z,rigid}(X,Y) \triangleq \inf_{\Phi} d_{\mathcal{H}}^{Z}(X, \Phi(Y)) \tag{5.5}$$

where $\Phi : Z \to Z$ ranges over all self-isometries of Z. If we knew an efficient way of computing $\inf_\Phi d_{\mathcal{H}}^{Z}(X,\Phi(Y))$, then this restricted shape recognition problem would be well posed for Z, in view of an adapted version of Proposition 3 and Corollary 6, as soon as we can give guarantees of coverage. For the sake of completeness we state such a result.

Proposition 4 *$d_{\mathcal{H}}^{Z,rigid}(\cdot,\cdot)$ satisfies the triangle inequality and in particular, the following relation holds:*

$$\left| d_{\mathcal{H}}^{Z,rigid}(X,Y) - d_{\mathcal{H}}^{Z,rigid}(\mathbb{X}_m, \mathbb{Y}_{m'}) \right| \leq r + r'$$

for compact $X, Y \subset Z$ such that $X \subset B_Z(\mathbb{X}_m, r)$ and $Y \subset B_Z(\mathbb{Y}_{m'}, r')$.

Coverage can be guaranteed, in the case of sub-manifolds of $I\!R^d$, by imposing a probabilistic model on the samplings $\mathbb{X}_m$ of the manifolds, and a bound on the curvatures of the family of manifolds one wishes to work with. In more detail, for given $r > 0$ and $p \in (0,1)$, we can show that there exists finite $m_{p,r}$ such that

$$\mathbb{P}\left(d_{\mathcal{H}}^{I\!R^d}(X, \mathbb{X}_m) > r \right) \leq 1 - p$$

for $m \geq m_{p,r}$, see Section §5.3.2.

In the case of surfaces in $Z = I\!R^3$, Φ sweeps all rigid isometries, and there exist good algorithms which can actually solve the problem approximately. For example, in [84] the authors report an algorithm which for any given $0 < \alpha < 1$ can find a rigid transformation $\widehat{\Phi}_\alpha$ such that

$$d_{\mathcal{H}}^{I\!R^3}(\mathbb{X}_m, \widehat{\Phi}_\alpha(\mathbb{Y}_{m'})) \leq (8 + \alpha) \inf_{\Phi} d_{\mathcal{H}}^{I\!R^3}(\mathbb{X}_m, \Phi(\mathbb{Y}_{m'}))$$

with complexity $O(s^4 \log s)$ where $s = \max(m, m')$. This computational result, together with simple considerations, makes the problem of surface recognition (under rigid motions) well posed and well justified. In fact, using Proposition 4 we obtain a bound between the distance we want to estimate $d_{\mathcal{H}}^{I\!R^3,rigid}(X,Y)$ and the observable (computable) value $d_{\mathcal{H}}^{I\!R^3}(\mathbb{X}_m, \widehat{\Phi}_\alpha(\mathbb{Y}_{m'}))$:

$$d_{\mathcal{H}}^{I\!R^3,rigid}(X,Y) - (r + r') \leq \; d_{\mathcal{H}}^{I\!R^3}(\mathbb{X}_m, \widehat{\Phi}_\alpha(\mathbb{Y}_{m'})) \;\; \leq 10 \left(d_{\mathcal{H}}^{I\!R^3,rigid}(X,Y) + (r + r') \right) \tag{5.6}$$

Equation (5.6) gives a formal justification to the procedure outlined for this surface recognition problem. To the best of our knowledge, this is the first time such formality is presented for this very important problem, both in the particular case just shown and for the general one addressed next. In any case, if $d_{\mathbb{S}}$ is the measure of similarity we are considering, and $\widehat{d_{\mathbb{S}}}$ is the *computable* approximate measure of similarity, the kind of relation we seek to establish is

$$A(d_{\mathbb{S}}(X,Y) - \alpha) \leq \widehat{d_{\mathbb{S}}}(\mathbb{X}_m, \mathbb{Y}_{m'}) \leq B(d_{\mathbb{S}}(X,Y) + \beta) \tag{5.7}$$

for some constants A, B and numbers α and β which can be made small by refining the samplings. Moreover, it may happen that relation (5.7) holds *with a certain probability.* Every recognition task needs to be supported by a relation of this type, see also [56].

5.2.2 The General Case

The theory introduced by Gromov addresses the concept of metric approximation between metric spaces. When dealing with discrete metric spaces, as those arising from samplings or coverings of continuous ones, it is convenient to introduce another distance between them which ultimately is the one we compute for point clouds, see §5.3.6 ahead. For discrete metric spaces (both of cardinality n) $(\mathbb{X} = \{x_1, \ldots, x_n\}, d_{\mathbb{X}})$ and

$(\mathbb{Y} = \{y_1, \ldots, y_n\}, d_{\mathbb{Y}})$ we define the distance: [5]

$$d_{\mathcal{I}}(\mathbb{X}, \mathbb{Y}) \triangleq \min_{\pi \in \Pi_n} \max_{1 \leq i,j \leq n} \frac{1}{2} |d_{\mathbb{X}}(x_i, x_j) - d_{\mathbb{Y}}(y_{\pi_i}, y_{\pi_j})| \tag{5.8}$$

where Π_n stands for the set of all permutations of $\{1, \ldots, n\}$. A permutation π provides the correspondence between the points in the sets, and $|d_{\mathbb{X}}(x_i, x_j) - d_{\mathbb{Y}}(y_{\pi_i}, y_{\pi_j})|$ gives the pointwise distance/disparity once this correspondence has been assumed.

It is evident that one has, by virtue of Property 5 from Proposition 3 (and Remark 15, if we take the infimum over invertible maps $\psi = \phi^{-1}$):

$$d_{\mathcal{GH}}(\mathbb{X}, \mathbb{Y}) \leq d_{\mathcal{I}}(\mathbb{X}, \mathbb{Y}) \tag{5.9}$$

Moreover, we easily derive the following bound, whose usefulness will be made evident in §5.3.

Corollary 7 *Let (X, d_X) and (Y, d_Y) be compact metric spaces. Let $\mathbb{X} = \{x_1, \ldots, x_n\} \subset X$ and $\mathbb{Y} = \{y_1, \ldots, y_n\} \subset Y$, such that $B_X(\mathbb{X}, R_X) = X$ and $B_Y(\mathbb{Y}, R_Y) = Y$ (the point clouds provide R_X and R_Y coverings respectively). Then*

$$d_{\mathcal{GH}}(X, Y) \leq R_X + R_Y + d_{\mathcal{I}}(\mathbb{X}, \mathbb{Y}) \tag{5.10}$$

with the understanding that $d_{\mathbb{X}} = d_X \mid_{\mathbb{X} \times \mathbb{X}}$ and $d_{\mathbb{Y}} = d_Y \mid_{\mathbb{Y} \times \mathbb{Y}}$.

Remark 17 *This result tells us that if we manage to find coverings of X and Y for which the distance $d_{\mathcal{I}}$ is small, then if the radii those coverings are also small, the underlying manifolds X and Y sampled by these point clouds must be close in a metric sense. Another way of interpreting this is that we will never see a small value of $d_{\mathcal{I}}(\mathbb{X}, \mathbb{Y})$ whenever $d_{\mathcal{GH}}(X, Y)$ is big, a simple statement with practical value, since we will only be able to look at values of $d_{\mathcal{I}}$, which depend on the point clouds. This is because, in contrast with $d_{\mathcal{GH}}(,)$, the distance $d_{\mathcal{I}}$ is (approximately) computable from the point clouds, see §5.3.6.*

We now introduce some additional notation regarding coverings of metric spaces. Given a metric space (X, d_X), the discrete subset $N_{X,n}^{(R,s)}$ denotes a set of points $\{x_1, \ldots, x_n\} \subset X$ such that (1) $B_X(N_{X,n}^{(R,s)}, R) = X$, and (2) $d_X(x_i, x_j) \geq s$ whenever $i \neq j$. In other words, the set constitutes a R-covering and the points in the set are not too close to each other.

Remark 18 *For each $r > 0$ denote by $N(r, X)$ the minimum number of closed balls of radii r needed to cover X. Then, ([168], Chapter 10), we can actually show that the class $(\mathcal{M}, d_{\mathcal{GH}})$ of all compact metric spaces X whose covering number $N(r, X)$ are bounded for all (small) positive r by a function $N : (0, C_1) \to (0, \infty)$ is* totally bounded. *This means that given $\rho > 0$, there exist a finite positive integer $k(\rho)$ and compact metric spaces $X_1, \ldots, X_{k(\rho)} \in \mathcal{M}$ such that for any $X \in \mathcal{M}$ one can find $i \in \{1, \ldots, k(\rho)\}$ such that $d_{\mathcal{GH}}(X, X_i) \leq \rho$. This is very interesting from the point of view of applications since it gives formal justification to the classification problem of metric spaces. For example, in a system of storage/retrieval of faces/information manifolds, this concept permits the design of a clustering procedure for the shapes.*

The following Proposition will also be fundamental for our computational framework in §5.3.

[5] One can easily check that this is really a distance.

Proposition 5 ([87]) *Let (X, d_X) and (Y, d_Y) be any pair of given compact metric spaces and let $\eta = d_{\mathcal{GH}}(X, Y)$. Also, let $N_{X,n}^{(R,s)} = \{x_1, \ldots, x_n\}$ be given. Then, given $\alpha > 0$ there exist points $\{y_1^\alpha, \ldots, y_n^\alpha\} \subset Y$ such that*

1. $d_{\mathcal{I}}(N_{X,n}^{(R,s)}, \{y_1^\alpha, \ldots, y_n^\alpha\}) \leq (\eta + \alpha)$
2. $B_Y(\{y_1^\alpha, \ldots, y_n^\alpha\}, R + 2(\eta + \alpha)) = Y$
3. $d_Y(y_i^\alpha, y_j^\alpha) \geq s - 2(\eta + \alpha)$ *for* $i \neq j$.

Remark 19 *This proposition tells us that if the metric spaces happen to be sufficiently close in a metric sense, then given a s-separated covering on one of them, one can find a (s′-separated) covering in the other metric space such that $d_{\mathcal{I}}$ between those coverings (point clouds) is also small. This, in conjunction with Remark 17, proves that in fact our goal of trying to determine the metric similarity of metric spaces based on discrete observations of them is, so far, a (theoretically) well posed problem.*

Since by Tychonoff's Theorem the n-fold product space $Y \times \ldots \times Y$ is compact, if $s - 2\eta \geq c > 0$ for some positive constant c, by passing to the limit along the subsequences of $\{y_1^\alpha, \ldots, y_n^\alpha\}_{\{\alpha > 0\}}$ (if needed) above one can assume the existence of a set of different points $\{\bar{y}_1, \ldots, \bar{y}_n\} \subset Y$ such that $d_{\mathcal{I}}(\{\bar{y}_1, \ldots, \bar{y}_n\}, N_{X,n}^{(R,s)}) \leq \eta$, $\min_{i \neq j} d_Y(\bar{y}_i, \bar{y}_j) \geq s - 2\eta > 0$, and $B_Y(\{\bar{y}_1, \ldots, \bar{y}_n\}, R + 2\eta) = Y$.

Since we are only given finite sets of points sampled from each metric space, the existence of $\{\bar{y}_1, \ldots, \bar{y}_n\}$ guaranteed by Proposition 5 and Remark 19 doesn't seem to make our life a lot easier since those points could very well not be contained in our given finite datasets. The simple idea of using a triangle inequality (with metric $d_{\mathcal{I}}$) to deal with this does not work in principle, since one can find, for the same underlying space, two covering nets whose $d_{\mathcal{I}}$ distance is not small, see [29, 141]. Let us explain this in more detail. Assume that as input we are given two finite sets of points $\mathbb{X}_m$ and $\mathbb{Y}_m$ on two metric spaces, X and Y respectively, which we assume to be isometric. Then the results above ensure that for any given $N_{X,n}^{(R,s)} \subset \mathbb{X}_m$ there exists a $N_{Y,n}^{(R,s)} \subset Y$ such that $d_{\mathcal{I}}(N_{X,n}^{(R,s)}, N_{Y,n}^{(R,s)}) = 0$. However, it is clear that this $N_{Y,n}^{(R,s)}$ has no reason to be contained in the given point cloud $\mathbb{Y}_m$. The obvious idea would be try to rely on some kind of property of independence on the sample representing a given metric space, namely that for any two different covering nets N_1 and N_2 (of the same cardinality and with small covering radii) of X the distance $d_{\mathcal{I}}(N_1, N_2)$ is also small. If this were granted, we could proceed as follows:

$$\begin{aligned} d_{\mathcal{I}}(N_{X,n}^{(R,s)}, N_{Y,n}^{(\hat{R},\hat{s})}) &\leq d_{\mathcal{I}}(N_{X,n}^{(R,s)}, N_{Y,n}^{(R,s)}) + d_{\mathcal{I}}(N_{Y,n}^{(\hat{R},\hat{s})}, N_{Y,n}^{(R,s)}) \\ &= 0 + \text{small}(R, \hat{R}) \end{aligned} \tag{5.11}$$

where *small*$(R, \hat{R})$ is small number depending only on R and $\hat{R}$. The property we fancy to rely upon was conjectured by Gromov in [86] (see also [196]) and disproved by Burago & Kleiner in [29] and Mc.Mullen in [141], see also [153] for certain positive results. Their counterexamples are for separated covering nets in $\mathbb{Z}^2$. It is not known whether one can construct counterexamples for compact metric spaces, or if there exists a characterization of a family of n-points separated covering nets of a given compact metric space such that any two of them are at a small $d_{\mathcal{I}}$-distance which can be somehow controlled with n. A first step towards this is the density condition introduced in [30].

If counterexamples didn't exist for compact metric spaces, then the above inequality would be sufficient. Without assuming this, we give below an argument which tackles the problem in a probabilistic way. In other words, we use a probabilistic approach to bound $d_{\mathcal{J}}$ for two different samples from a given metric space. For this, we pay the price of assuming the existence of a measure which comes with our metric space.[6] On the other hand, probabilistic frameworks are natural for (maybe noisy) random samples of manifolds as obtained in real applications.[7]

5.2.3 A Probabilistic Setting for Sub-manifolds of $I\!R^d$

We now limit ourself to smooth Riemannian sub-manifolds of $I\!R^d$ endowed with the metric inherited from ambient space. However, the work can be extended to more general metric spaces, see further comments in §5.5.2. In what follows, for an *event* $\mathcal{E}$, $\mathbb{P}(\mathcal{E})$ will denote its probability and for a random variable x, $\mathbb{E}(x)$ will denote its expected value.

Let Z be a smooth and compact sub-manifold of $I\!R^d$ with intrinsic (geodesic) distance function $d_Z(\cdot,\cdot)$. We can now speak more freely about points $\{z_i\}_{i=1}^m$ sampled uniformly from X: We say that the random point $\widehat{z}$ is *uniformly distributed* on Z if for any measurable $C \subset Z$, $\mathbb{P}(\widehat{z} \in C) = \frac{\mathbf{a}(C)}{\mathbf{a}(Z)}$, where $\mathbf{a}(B)$ denotes the area of the measurable set $B \subset Z$. This uniform probability measure can be replaced by other probability measures which for example adapt to the geometry of the underlying surface, and the framework here developed can be extended to those as well, see comments in §5.5.2.

Let $\mathbb{Z} = \{z_1, \ldots, z_n\}$ and $\mathbb{Z}' = \{z'_1, \ldots, z'_n\}$ be two discrete subsets of Z (two point clouds). For any permutation $\pi \in \Pi_n$ and $i, j \in \{1, \ldots, n\}$,

$$|d_Z(z_i, z_j) - d_Z(z'_{\pi_i}, z'_{\pi_j})| \leq d_Z(z_i, z'_{\pi_i}) + d_Z(z_j, z'_{\pi_j})$$

and therefore we have

$$d_{\mathcal{B}}^Z(\mathbb{Z}, \mathbb{Z}') \overset{\triangle}{=} \min_{\pi \in \Pi_n} \max_k d_Z(z_k, z'_{\pi_k}) \geq d_{\mathcal{J}}(\mathbb{Z}, \mathbb{Z}') \tag{5.12}$$

This is known as the *Bottleneck Distance* between $\mathbb{Z}$ and $\mathbb{Z}'$, both being subsets of Z. This is one possible way of measuring distance between two different samples of the same metric space.[8]

Instead of dealing with (5.11) deterministically, after imposing conditions on the underlying metric spaces X and Y, we derive probabilistic bounds for the left hand side. We also make evident that by suitable choices of the relations among the different parameters, this probability can be chosen at will. This result is then used to bound the distance $d_{\mathcal{J}}$ between two point cloud samples of a given metric space, thereby leading to the type of bound expressed in Equation (5.11) and from this, the bounds on the original Gromov-Hausdorff distance between the underlying objects.

We introduce the *Voronoi* diagram $\mathcal{V}(\mathbb{Z})$ on Z, determined by the points in $\mathbb{Z}$ (see for example [131]).

[6] In the present work we therefore deal only with the case of sub-manifolds of $I\!R^d$.

[7] In more generality, data are acquired by sensors or arrays of sensors which return a value in $I\!R^d$ for some $d \geq 1$. The acquisition process or the sensors themselves might be subject to some perturbations (miscalibrations of mechanical parts of a $3D$-scanner, electric noise in electrodes, etc). Under the assumption of existence of an underlying structure from which the data are sampled, it therefore seems sensible to introduce a probability measure which models the acquisition process.

[8] In [153], this distance is used to establish the equivalence (according to this notion) of separated nets in certain Hyperbolic metric spaces.

The i-th Voronoi cell of the Voronoi diagram defined by $\{z_1, \ldots, z_n\} \subset Z$ is given by

$$V_i \overset{\triangle}{=} \{z \in Z|\ d_Z(z_i, z) < \min_{j \neq i} d_Z(z_j, z)\} \tag{5.13}$$

We then have $Z = \bigsqcup_{k=1}^{n} \overline{V_k}$.

Lemma 6 *1. If the points $\{z_1, \ldots, z_n\}$ are s-separated, then for any $1 \leq i \leq n$, $B_Z(z_i, \frac{s}{2}) \subset V_i$.*

2. If the points $\{z_1, \ldots, z_n\}$ constitute a R-covering of Z, then $V_i \subseteq B_Z(z_i, R)$ for all $i = 1, \ldots, n$.

Proof:
To prove 1. first note that for any $z \in Z$ and $i \neq j$, $d_Z(z, z_i) + d_Z(z, z_j) \geq s$ by the triangle inequality. Assume in particular that $z \in B_Z(z_i, \frac{s}{2})$, then $d_Z(z, z_i) < \frac{s}{2}$ and $d_Z(z, z_j) > \frac{s}{2}$ for all $j \neq i$, then $z \in V_i$. To prove 2. assume $z \in V_i$ but $z \notin B_Z(z_i, R)$, that is $d_Z(z, z_i) \geq R$. But since $\{z_1, \ldots, z_n\}$ is a R-covering of Z, z must belong to a certain $B_Z(x_k, R)$ for some $k \neq i$, that is $d_Z(z, z_k) < R$. But then z is closer to z_k than to z_i, which contradicts $z \in V_i$. □

We consider $\mathbb{Z}$ to be fixed, and we assume $\mathbb{Z}' = \{z_1', \ldots, z_n'\}$ to be chosen from a set $\mathbb{Z}_m \subset Z$ consisting of $m \gg n$ i.i.d. points sampled uniformly from Z.

We first want to find, amongst points in $\mathbb{Z}_m$, n different points $\{z_{i_1}, \ldots, z_{i_n}\}$ such that each of them belongs to one Voronoi cell, $\{z_{i_k} \in V_k$ for $k = 1, \ldots, n\}$. We provide lower bounds for $\mathbb{P}\left(\#\left(V_k \cap \mathbb{Z}_m\right) \geq 1,\ 1 \leq k \leq n\right)$, the probability of this happening.

We can see the event as if we *collected* points inside all the Voronoi cells, a case of the *Coupon Collecting Problem*, see [70]. We buy merchandise at a coupons-giving store until we have collected all possible types of coupons. The next Lemma presents the basic results we need about this concept. These results are due to Von Schilling ([176]) and Borwein and Hijab ([22]).

Lemma 7 (Coupon Collecting) *If there are n different coupons one wishes to collect, such that the probability of seeing the k-th coupon is $p_k \in (0,1)$, (let $\vec{p} = (p_1, \ldots, p_n)$), and one obtains samples of all of them in an independent way then:*

1. ([176]) The probability $P_{\vec{p}}(n, m)$ of having collected all n coupons after m trials is given by

$$P_{\vec{p}}(n, m) = 1 - S_n \left(\sum_{j=2}^{n} (-1)^j \left(\sum_{k=j}^{n} p_k \right)^m \right) \tag{5.14}$$

where the symbol S_n means that we consider all possible combinations of the n indices in the expression being evaluated.[9]

2. ([22]) The expected value of the number of trials needed to collect all the coupons is given by

$$E_{\vec{p}}(n) = \mathbb{E}\left(\max_{1 \leq i \leq n} \frac{X_i}{p_i} \right) \tag{5.15}$$

where X_i are independent positive random variables satisfying $\mathbb{P}\left(X_i > t\right) = e^{-t}$ for $t \geq 0$ and $1 \leq i \leq n$.

[9] For example $S_3((p_1 + p_2)^k) = (p_1 + p_2)^k + (p_1 + p_3)^k + (p_2 + p_3)^k$.

For $n \in I\!N$ let $h_n := \sum_{i=1}^{n} i^{-1}$.

Corollary 8 $P_{\vec{p}}(n,m) \geq 1 - \frac{h_n}{m \cdot \min_k p_k}$.

Proof:
By Markov's inequality, $1 - P_{\vec{p}}(n,m) \leq \frac{E_{\vec{p}(n)}}{m}$. Now, note that $E_{\vec{p}}(n)$ is decreasing in each p_k for $p_k \geq 0$, then it is clear that $E_{\vec{p}}(n) \leq \frac{E_{\vec{1}}(n)}{\min_k p_k}$. On the other hand, it is easy to compute by direct probabilistic calculation that $E_{\frac{\vec{1}}{n}}(n) = nh_n$. We conclude by noting that, by (5.15), $cE_{c\vec{p}}(n) = E_{\vec{p}}(n)$ for any $c > 0$. □

We now directly use these results to bound the bottleneck distance.

Theorem 10 *Let (Z, d_Z) be a smooth compact sub-manifold of $I\!R^d$. Given a covering $N^{(R,s)}_{Z,n}$ of Z with separation $s > 0$ and a number $p \in (0,1)$, there exists a positive integer $m = m_n(p)$ such that if $\mathbb{Z}_m = \{z_k\}_{k=1}^m$ is a sequence of i.i.d. points sampled uniformly from Z, with probability p one can find a set of n different indices $\{i_1, \dots, i_n\} \subset \{1, \dots, m\}$ with*

$$d^Z_{\mathcal{B}}(N^{(R,s)}_{Z,n}, \{z_{i_1}, \dots, z_{i_n}\}) \leq R \qquad \text{and} \qquad Z = \bigcup_{k=1}^{n} B_Z(z_{i_k}, 2R).$$

Moreover, $m_n(p) \leq \left[\frac{h_n}{\min_z \mathbf{a}\left(B_Z(z, \frac{s}{2})\right)} \frac{\mathbf{a}(Z)}{1-p} \right] + 1$.[10]

This result can also be seen the other way around: For a given m, the probability of finding the aforementioned subset in $\mathbb{Z}_m$ is $P_{\vec{p}_Z}(n,m)$ as given by (5.14), for suitably defined $\vec{p}_Z$. The precise form of $\vec{p}_Z$ can be understood from the proof.

Proof:
Let $N^{(R,s)}_{Z,n} = \{\widehat{z}_1, \dots, \widehat{z}_n\}$. We consider the coupon collecting problem in which the k-th coupon has been acquired at least once if $\#\{\mathbb{Z}_m \cap V_k\} \geq 1$, where V_k is the k-th cell of the Voronoi partition corresponding to the covering net $N^{(R,s)}_{Z,n}$. The components of the probability vector $\vec{p}$ are given by $p_k = \frac{\mathbf{a}(V_k)}{\mathbf{a}(Z)}$ for $k = 1, \dots, n$. Using the fact that (5.14) is increasing in the number of trials m,[11] we see that given p we can find a positive integer M such that for $m \geq M$

$$\mathbb{P}\left(\bigcap_{k=1}^{n} \{\#\{\mathbb{Z}_m \cap V_k\} \geq 1\} \right) \geq p$$

Discarding points when more than one has been found inside the same V_k, we can obtain with probability at least p, exactly one point inside each V_k. Let $i_1, \dots, i_n$ be indices such that $z_{i_k} \in V_k$ for $k = 1, \dots, n$. Then $d^Z_{\mathcal{B}}(\{z_{i_1}, \dots, z_{i_n}\}, N^{(R,s)}_{Z,n}) \leq \max_{z \in \bar{A}_k} d_Z(z, \widehat{z}_k)$, since by Lemma 6, $V_k \subseteq B_Z(\widehat{z}_k, R)$, and this concludes the proof of the first claim. Also, by the very same steps plus the triangle inequality we prove that $\{z_{i_1}, \dots, z_{i_n}\}$ constitutes a $2R$-covering of Z. Finally, note that by Corollary 8, $P_{\vec{p}}(n,m) \geq p$ for $m \geq \frac{h_n}{(1-p)\min_k p_k}$. Since again by Lemma 6, $\mathbf{a}(V_k) \geq \min_{z \in Z} \mathbf{a}\left(B_Z(z, \frac{s}{2})\right)$ the last claim follows. □

Corollary 9 *Let X and Y compact sub-manifolds of $I\!R^d$. Let $N^{(R,s)}_{X,n}$ be a covering of X with separation s such that for some positive constant c, $s - 2d_{\mathcal{GH}}(X,Y) > c$. Then, given any number $p \in (0,1)$, there exists*

[10] For real x, $[x]$ stands for the largest integer not greater that x.

[11] Something obvious for which in, principle, we do not need to know the exact expression (5.14).

a positive integer $m = m_n(p)$ such that if $\mathbb{Y}_m = \{y_k\}_{k=1}^m$ is a sequence of i.i.d. points sampled uniformly from Y, we can find, with probability at least p, a set of n different indices $\{i_1, \ldots, i_n\} \subset \{1, \ldots, m\}$ such that

$$d_{\mathcal{J}}(N_{X,n}^{(R,s)}, \{y_{i_1}, \ldots, y_{i_n}\}) \leq 3\, d_{\mathcal{GH}}(X,Y) + R \qquad \textit{and} \qquad Y = \bigcup_{k=1}^{n} B_Y(y_{i_k}, 2(R + 2d_{\mathcal{GH}}(X,Y))).$$

Moreover, $m_n(p) \leq \left[\frac{h_n}{\min_y \mathbf{a}\left(B_Y(y, \frac{c}{2})\right)} \frac{\mathbf{a}(Y)}{1-p} \right] + 1.$

Proof:
Let $\eta = d_{\mathcal{GH}}(X,Y)$. Following Remark 19, we can find a $(R + 2\eta, s - 2\eta)$ n-covering of Y, which we denote by $N_{Y,n}^{(\tilde{R},\tilde{s})}$, such that $d_{\mathcal{J}}(N_{X,n}^{(R,s)}, N_{Y,n}^{(\tilde{R},\tilde{s})}) \leq \eta$. Let, as in Theorem 10, $m = m_n(p)$ be such that for any i.i.d. set of points $\mathbb{Y}_m = \{y_1, \ldots, y_m\}$ uniformly sampled from Y one has

$$\mathbb{P}\left(\exists\, \{y_{i_1}, \ldots, y_{i_n}\} \subset \mathbb{Y}_m \;:\; d_{\mathcal{B}}^{Y}\left(N_{Y,n}^{(\tilde{R},\tilde{s})}, \{y_{i_1}, \ldots, y_{i_n}\}\right) \leq \tilde{R}\right) \geq p$$

where $i_1, \ldots, i_n$ are different indices. Let $N_{Y,n} \subset Y$ be a set of n different points. Then, using the triangle inequality

$$\begin{aligned} d_{\mathcal{J}}(N_{X,n}^{(R,s)}, N_{Y,n}) &\leq d_{\mathcal{J}}(N_{X,n}^{(R,s)}, N_{Y,n}^{(\tilde{R},\tilde{s})}) + d_{\mathcal{J}}(N_{Y,n}, N_{Y,n}^{(\tilde{R},\tilde{s})}) \\ &\leq \eta + d_{\mathcal{B}}^{Y}(N_{Y,n}, N_{Y,n}^{(\tilde{R},\tilde{s})}) \end{aligned}$$

Hence we obtain, by Theorem 10,

$$\mathbb{P}\left(\exists\, N_{Y,n} \subset \mathbb{Y}_m \;:\; d_{\mathcal{J}}\left(N_{X,n}^{(\tilde{R},\tilde{s})}, N_{Y,n}\right) \leq \eta + \tilde{R}\right) \geq p.$$

The other claims follow just like in the proof of Theorem 10. □

Remark 20 *1. The preceding Corollary deals with the case of positive detection: X and Y are* nearly isometric *and we wish to detect this by only accessing the point clouds. The constant c quantifies this metric proximity as encoded by the phrase "*nearly isometric.*" For instance, for a recognition task where for any two* similar *objects X and Y, $d_{\mathcal{GH}}(X,Y) \leq \eta_{max}$, one could choose $c = s - 2\eta_{max}$.*

2. Note that the probability $P_{\vec{p}_Y}(n,m)$ itself (or $m_n(p)$) depends on $d_{\mathcal{GH}}(X,Y)$ through the constant c, see an example of the application of these ideas in §5.3.4 ahead.

Note also that one can write down the following useful bound

$$P_{\vec{p}_Y}(n,m) \geq 1 - \frac{h_n}{m \cdot \min_{y \in Y} \mathbf{a}\left(B_Y(y, \frac{c}{2})\right)} \mathbf{a}(Y) \tag{5.16}$$

which was implicitly used in the proof of Theorem 10. It is sensible to assume one is interested in performing the recognition/classification task for a number of objects which satisfy certain conditions, that is, tune the framework to a particular class of objects. In particular, suppose the class is characterized, among other conditions, by an upper bound on the sectional curvatures. For small $r > 0$ this allows, via Bishop-Günther's Theorem, to obtain a lower bound on $\min_z \mathbf{a}(B_Z(z,r))$ valid $\underline{\text{for all}}$ objects Z in *the class. This in turn can be used to calibrate the system to provide any prespecified probability p as in Corollary 9 for any two objects within the class, see §5.3.4 and §5.3.2 for a more detailed presentation of this ideas.*

A rougher estimate of the value of $m_n(p)$ alluded to in Corollary 9 can be obtained using the value of $E_{\vec{p}}(n)$ when all the coupons are equally likely: $m \simeq E_{\frac{1}{n}}(n) = n \cdot h_n \simeq n \log n$.

This concludes the main theoretical foundation of our proposed framework. Now, we must devise a computational procedure which allows us to actually find the subset $N_{Y,n}$ inside the given point cloud $\mathbb{Y}_m$ when it exists, or at least find it with a large probability. Note that in practise we can only access metric information, that is, interpoint distances. A stronger result in the same spirit of Theorem 10 should take into account possible self-isometries of X (Y), which would increase the probability of finding a net which achieves small $d_{\mathcal{I}}$ distance to the fixed one. We present such a computational framework next.

5.3 Computational Foundations

There are a number of additional issues that must be dealt with in order to develop an algorithmic procedure from the theoretical results previously presented. These are now addressed.

5.3.1 Initial Considerations

In practise our input consists of two statistically independent point clouds $\mathbb{X}_m$ and $\mathbb{Y}_{m'}$ each of them composed of i.i.d. points sampled uniformly from X and Y, respectively. For a positive integer $n \ll \min(m, m')$ we construct good coverings $N_{X,n}^{(R,s)}$ of X and $N_{Y,n}^{(R',s')}$ of Y, respectively. Actually, R, s, R' and s' all depend on n, and we should choose n such that R and R' are small enough to make our bounds useful, see the additional computations below. Details on how we construct these coverings are provided in Section §5.3.3. We will assume, without loss of generality, that these coverings are statistically independent of $\mathbb{X}_m$ and $\mathbb{Y}_{m'}$.

It is convenient to introduce the following additional notation: For a set of points $\mathbb{Z}_q = \{z_k\}_{k=1}^q$ and for a set of indices $I_u = \{i_1, \ldots, i_u\} \subset \{1, \ldots, q\}$, let $\mathbb{Z}_q[I_u]$ denote the subset $\{z_{i_1}, \ldots, z_{i_u}\}$ of $\mathbb{Z}_q$.

Corollary 9 suggests that in practise we compute the following symmetric expression

$$d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'}) \overset{\triangle}{=} \max\left(\min_{J_n \subset \{1,\ldots,m\}} d_{\mathcal{I}}(N_{X,n}^{(R,s)}, \mathbb{Y}_{m'}[J_n]), \min_{I_n \subset \{1,\ldots,m\}} d_{\mathcal{I}}(N_{Y,n}^{(R',s')}, \mathbb{X}_m[I_n])\right) \tag{5.17}$$

which depends not only on $\mathbb{X}_m$ and $\mathbb{Y}_{m'}$ but also on pre-specified covering nets $N_{X,n}^{(R,s)}$ and $N_{Y,n}^{(R',s')}$. However we prefer to omit this dependence in the list of arguments in order to keep the notation simpler.

Then, $d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'})$ upper bounds $d_{\mathcal{GH}}(\mathbb{X}_m, \mathbb{Y}_{m'})$, something we need to require. In fact, for any $I_n \subset \{1, \ldots, m\}$, using the triangle inequality for $d_{\mathcal{GH}}$ (Property 1 from Proposition 3) and then Property 3 from Proposition 3:

$$\begin{aligned} d_{\mathcal{GH}}(\mathbb{X}_m, \mathbb{Y}_{m'}) &\leq d_{\mathcal{GH}}(\mathbb{X}_m, \mathbb{X}_m[I_n]) + d_{\mathcal{GH}}(\mathbb{X}_m[I_n], \mathbb{Y}_{m'}) \\ &\leq d_{\mathcal{GH}}(\mathbb{X}_m, \mathbb{X}_m[I_n]) + d_{\mathcal{GH}}(\mathbb{X}_m[I_n], N_{Y,n}^{(R',s')}) + d_{\mathcal{GH}}(N_{Y,n}^{(R',s')}, \mathbb{Y}_{m'}) \\ &\leq d_{\mathcal{H}}^X(\mathbb{X}_m, \mathbb{X}_m[I_n]) + d_{\mathcal{I}}(\mathbb{X}_m[I_n], N_{Y,n}^{(R',s')}) + R' \end{aligned}$$

Now, considering I_n^* such that $d_{\mathcal{J}}(\mathbb{X}_m[I_n^*], N_{Y,n}^{(R',s')}) = \min_{I_n\subset\{1,\dots,m\}} d_{\mathcal{J}}(N_{Y,n}^{(R,s)}, \mathbb{X}_m[I_n])$, we find

$$d_{\mathcal{GH}}(\mathbb{X}_m, \mathbb{Y}_{m'}) \leq d_{\mathcal{H}}^X(\mathbb{X}_m, \mathbb{X}_m[I_n^*]) + d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'}) + R'$$

Symmetrically, we also obtain for J_n^* such that $d_{\mathcal{J}}(\mathbb{Y}_m[J_n^*], N_{X,n}^{(R,s)}) = \min_{J_n\subset\{1,\dots,m'\}} d_{\mathcal{J}}(N_{X,n}^{(R,s)}, \mathbb{Y}_{m'}[J_n])$

$$d_{\mathcal{GH}}(\mathbb{X}_m, \mathbb{Y}_{m'}) \leq d_{\mathcal{H}}^Y(\mathbb{Y}_{m'}, \mathbb{Y}_{m'}[J_n^*]) + d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'}) + R.$$

Hence, combining the last two expressions

$$\begin{aligned} d_{\mathcal{GH}}(\mathbb{X}_m, \mathbb{Y}_{m'}) &\leq d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'}) \\ &+ \min\left(d_{\mathcal{H}}^X(\mathbb{X}_m, \mathbb{X}_m[I_n^*]),\, d_{\mathcal{H}}^Y(\mathbb{Y}_{m'}, \mathbb{Y}_{m'}[J_n^*])\right) \\ &+ \max(R, R') \end{aligned} \tag{5.18}$$

what implies (Corollary 6) a similar upper bound for $d_{\mathcal{GH}}(X,Y)$. In fact, let $r_m := d_{\mathcal{H}}^X(X, \mathbb{X}_m)$ and $r_{m'} := d_{\mathcal{H}}^Y(Y, \mathbb{Y}_{m'})$, then

$$\begin{aligned} d_{\mathcal{GH}}(X, Y) &\leq d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'}) \\ &+ \min\left(d_{\mathcal{H}}^X(\mathbb{X}_m, \mathbb{X}_m[I_n^*]),\, d_{\mathcal{H}}^Y(\mathbb{Y}_{m'}, \mathbb{Y}_{m'}[J_n^*])\right) \\ &+ \max(R, R') + r_m + r_{m'} \end{aligned} \tag{5.19}$$

Let $\Delta_X := d_{\mathcal{H}}^X(\mathbb{X}_m, \mathbb{X}_m[I_n^*])$ and $\Delta_Y := d_{\mathcal{H}}^Y(\mathbb{Y}_{m'}, \mathbb{Y}_{m'}[J_n^*])$.

We now deal with the opposite kind of inequality. By Corollary 9 we know that with probability at least $P_{\vec{p}_X}(n,m) \times P_{\vec{p}_Y}(n,m')$ we will have both:[12]

$$d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'}) \leq 3\, d_{\mathcal{GH}}(X,Y) + \max(R, R') \tag{5.20}$$

and

$$\Delta_X \leq 2(R' + 2d_{\mathcal{GH}}(X,Y)) \quad \text{and} \quad \Delta_Y \leq 2(R + 2d_{\mathcal{GH}}(X,Y)) \tag{5.21}$$

and from this it follows in particular that $\min(\Delta_X, \Delta_Y) \leq 2\max(R,R') + 4d_{\mathcal{GH}}(X,Y)$ with the same probability.

Summing up, we have thus obtained:

$$d_{\mathcal{GH}}(X,Y) - \alpha(R,R',m,m') \;\leq\; \mathcal{L}(\mathbb{X}_m, \mathbb{Y}_{m'}) \;\overset{prob}{\leq}\; 7(d_{\mathcal{GH}}(X,Y) + \beta(R,R')) \tag{5.22}$$

where the symbol $\overset{prob}{\leq}$ means that the inequality holds with probability $P_{\vec{p}_X}(n,m) \times P_{\vec{p}_Y}(n,m')$, $\alpha(R,R',m,m') :=$

[12] Because we assumed $\mathbb{X}_m$ to be independent from $\mathbb{Y}_{m'}$.

$\max(R,R') + (r_m + r_{m'})$,[13] $\beta(R,R') := \frac{3}{7}\max(R,R')$, and

$$\mathcal{L}(\mathbb{X}_m, \mathbb{Y}_{m'}) := d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'}) + \min(\Delta_X, \Delta_Y). \tag{5.23}$$

Note, for future reference, that $\mathcal{L}(\mathbb{X}_m, \mathbb{Y}_{m'}) \leq \frac{3}{2}\max(\mathbf{diam}\,(X)\,, \mathbf{diam}\,(Y))$.

Remark 21 *Note that α, β and the probability can be controlled by suitably choosing all the parameters. We have therefore obtained an expression like the one anticipated in Section §5.2.1, Equation (5.7). The main difference is that we have not yet proved that $\mathcal{L}(\mathbb{X}_m, \mathbb{Y}_{m'})$ can be computed exactly or approximately in practise. In §5.3.6 we present a simple algorithm for approximately computing this quantity. We do not provide bounds on the fidelity of the algorithm in this Chapter. Results in this direction are subject of current efforts.*

Remark 22 *By a modification of the ideas here presented it may be possible to provide a framework recognition of partial objects: One might want to check whether one object is a part of another one. Clearly, in that case, one shouldn't but compute one half of $d_{\mathcal{F}}$. The covering net $N_{Y,n}^{(R',s')}$ should represent the object that we want to find inside the one represented by $\mathbb{X}_m$.*

5.3.2 Working with Point Clouds

All we have are finite sets of points (point clouds) sampled from each metric space, and all our computations must be based on *these observations* only. Since we made the assumption of randomness in the sampling (and it also makes sense in general to make a random model of the problem, given that the shapes are acquired by a scanner for example), we must relate the number of acquired data points to the coverage properties we wish to have. In other words, and following our theory above, we would like to say that given a desired probability p_c and a radius r_c, there exists a finite m such that the probability of covering all the metric space with m balls (intrinsic or not) of radius r_c centered at those m random points is at least p_c. This kind of characterizations are easy to deal with in the case of sub-manifolds of $\mathbb{R}^d$, where the *tuning* comes from the curvature bounds available. For this we follow [142]. Let Z be a smooth and compact sub-manifold of $\mathbb{R}^d$ of dimension k. Let $\mathbb{Z}_m = \{z_1, \ldots, z_m\} \subset Z$ consist of m i.i.d. points uniformly sampled from Z. For $r > 0$ define

$$f_Z(r) \stackrel{\triangle}{=} \min_{z \in Z} \mathbf{a}\left(B_Z(z,r)\right) \tag{5.24}$$

Then, for $p \in (0,1)$ and $\delta > 0$ we can prove that if $m \geq -\frac{\log((1-p)f_Z(\delta/4))}{f_Z(\delta/2)}$ then

$$p_{\delta,m} \stackrel{\triangle}{=} \mathbb{P}\left(Z \subseteq \cup_{i=1}^m B_Z(z_i,\delta)\right) \geq p \tag{5.25}$$

The function f_Z can be lower bounded using an upper bound, K, for the sectional curvatures of Z (Bishop-Günther Theorem, see [85]): $f_Z(r) \geq F_{K,k}(r)$ where $F_{K,k}(r)$ denotes the area of a ball of radius r in a space of constant sectional curvature K and dimension k. For example, when $K > 0$, one has $F_{K,k}(r) = \frac{2\pi^{k/2}}{\Gamma(k/2)}\int_0^r \left(\frac{\sin(t\sqrt{K})}{\sqrt{K}}\right)^{k-1} dt$.[14]

[13] Observe that $\alpha(R,R',m,m') \leq 3\max(R,R')$.

[14] $\Gamma(t)$ denotes the usual *Gamma* function: $\Gamma(t) = \int_0^\infty u^{t-1}e^{-u}\,du$.

This relation gives us some guidance about the number points we must sample in order to have a certain covering radius, or to estimate the covering radius in terms of m. An important point to remark is that this kind of relations, (5.25), should hold for the family of shapes we want to work with (in a way similar to the one exposed in §5.3.4), therefore, once given bounds on the curvatures that characterize the family, one can determine a precise probabilistic covering relation for it. We leave the exploitation/application of this idea for future work.

Given the natural number $n \leq m$ (or eventually $s > 0$), we use the oracle described in §5.3.3 below to find n-points from $\mathbb{Z}_m$ which constitute a covering of $\mathbb{Z}_m$ of the given cardinality n (or of the given separation s or given covering radius R) and of a resulting radius R_{res} (or resulting separation s_{res}). We denote this set by $N^{(R_{res},s)}_{\mathbb{Z}_m,n} \subseteq \mathbb{Z}_m$ (or $N^{(R,s_{res})}_{\mathbb{Z}_m,n} \subseteq \mathbb{Z}_m$, respectively).

5.3.3 Finding Coverings

In order to find the coverings $N^{(R,s)}_{Z,n}$, we use the well known Farthest Point Sampling (**FPS**) strategy, which we describe next. Suppose we have a dense sampling $\mathbb{Z}_m$ of the smooth and compact sub-manifold (Z, d_Z) of $\mathbb{R}^d$ as interpreted by the discussion above. We want to simplify our sampling and obtain a well separated covering net of the space. We also want to estimate the covering radius and separation of our covering net. It is important to obtain subsets which retain as best as possible the metric information contained in the initial point cloud in order to make computational tasks more treatable without sacrificing precision.

We first show a procedure to sample the whole of Z. Fix n the number of points we want to have in our simplified point cloud $\mathcal{P}_n$. We build $\mathcal{P}_n$ recursively. Given $\mathcal{P}_{n-1}$, we select $p \in Z$ such that $d_Z(p, \mathcal{P}_n) = \max_{z\in Z} d_Z(z, \mathcal{P}_{n-1})$ (here we consider of course, geodesic distances). There might exist more than one point which achieves the maximum, we either consider all of them or randomly select one and add it to $\mathcal{P}_{n-1}$. This sub-sampling procedure has been studied and efficiently implemented in [150] for the case of surfaces represented as point clouds.

The **FPS** procedure satisfies several useful properties as described below.

Let $M(\mathcal{P}_{n-1}) \subset Z$ denote the set of points z for which $d_Z(\mathcal{P}_{n-1}, z)$ is maximal. We denote by s_n and R_n the separation and covering radius of $\mathcal{P}_n \subset Z$, respectively.

Lemma 8 *Let $\mathcal{P}_n$ be the set obtained for each $n \geq n_0$ according to the* ***FPS*** *strategy starting from $\mathcal{P}_{n_0}$, and let p_{n+1} denote any point in $M(\mathcal{P}_n)$. Then, for $n \geq n_0$,*

1. $d^Z_{\mathcal{H}}(Z, \mathcal{P}_{n+1}) \leq d^Z_{\mathcal{H}}(Z, \mathcal{P}_n)$, *that is* $R_{n+1} \leq R_n$.
2. $d_Z(p_{n+2}, \mathcal{P}_{n+1}) \leq d_Z(p_{n+1}, \mathcal{P}_n)$.
3. $s_n \overset{\triangle}{=} \min_{1\leq i<j\leq n} d_Z(p_i, p_j) \geq \min_{1\leq i<j\leq n+1} d_Z(p_i, p_j) = s_{n+1}$.
4. $s_n = d_Z(p_n, \mathcal{P}_{n-1})$.
5. $d^Z_{\mathcal{H}}(Z, \mathcal{P}_n) = d_Z(p_{n+1}, \mathcal{P}_n)$.
6. $n \leq \frac{\mathbf{a}(Z)}{f_Z(\frac{s_n}{2})}$ *where* $\mathbf{a}$ *is the area measure on* Z *and* f_Z *was defined in (5.24).*[15]

[15] Note that with curvature bounds one can obtain a more explicit relation between s_n and n.

In practise we do not have access to Z but only to a point cloud, $\mathbb{Z}_m$, sampled from it. Anyhow, we can still follow the same algorithmic procedure.

Remark 23 *These properties make it easy to compute s_n and R_n* on the fly *inside the algorithm, something useful when the objective is to to obtain either a pre-specified covering radius or a minimal prespecified separation. However, it turns out to be useful to have an estimate on n depending on a prespecified covering radius, that is, we want to find n such that using the **FPS** we obtain a covering of Z consisting of n points and with radius not greater than ε. Property 6 in Lemma §8 gives us a way. Observe that $s_n = R_{n-1}$, then note that if $n \geq \frac{\mathbf{a}(A)}{f_Z(\varepsilon/2)}$ we must have $f_Z(s_n/2) \leq f_Z(\varepsilon/2)$. But f_Z is obviously non-decreasing, therefore $\varepsilon \geq s_n = R_{n-1} \geq R_n$.*

An interesting problem to solve has to do with the behavior of s_n with n through the **FPS** procedure, in the sense that we would like to know the maximum number of steps of the procedure that can be performed such that the separation remains larger than a prespecified number, check Remark 20. Note, for example, that in the case of the unit sphere S^2, the construction of the **FPS** net goes roughly as follows: The first two points are any antipodal points, call them p_1 and p_2. The third point can be any on the equator with respect to p_1 and p_2. The fourth point will lie still on the equator defined by p_1 and p_2 but will be antipodal to p_3. The next eight points will be the centers of the octants defined by the 3 maximal circumferences passing through $\{p_1, p_2, p_3, p_4\}$, $\{p_1, p_2, p_5, p_6\}$ and $\{p_3, p_4, p_5, p_6\}$. The construction follows a similar pattern for $n > 8$. We can therefore obtain an exact formula for s_n. Now, given a surface $\mathcal{S} \in \mathcal{A}$ ($\mathcal{A}$ is a certain class of compact smooth surfaces), we would like to be able to lower bound $s_n(\mathcal{S})$ by some quantity related to $s_n(S^2)$. Perhaps this will require assuming upper bounds on the Gauss curvature of $\mathcal{S} \in \mathcal{A}$. In any case, knowledge of this lower bounds will let us find, for any given $s > 0$, a $n_s \in I\!N$ such that for all $n \leq n_s$, $s_n(\mathcal{S}) \geq s$ for all $\mathcal{S} \in \mathcal{A}$.

Let us now assume that the discrete metric space $(\mathbb{Z}_m, d_Z)$ is a good random sampling of the underlying (Z, d_Z) in the sense that $d_{\mathcal{H}}(Z, \mathbb{Z}_m) \leq r$ with a certain probability $p_{r,m}$, as discussed in Section §5.3.2. We then want to simplify $\mathbb{Z}_m$ in order to obtain a set $\mathcal{P}_n$ with n points which is both a good sub-sampling and a well separated covering net of X.[16] We want to use our n sampled points in the best possible way. We are then led to using the construction discussed above. For example, choose randomly one point $p_1 \in \mathbb{Z}_m$ and consider $\mathcal{P}_1 = \{p_1\}$.[17] Run the procedure **FPS** until $n-1$ other points have been added to the set of points.[18] Compute now $r_n \overset{\triangle}{=} \max_{q \in \mathbb{X}_m} d(q, \mathcal{P}_n)$. Then, also with probability $p_{r,m}$, $\mathcal{P}_n$ is a $(r + r_n)$-covering net of X with separation s_n as expressed in Lemma 8. Following this, we now use the notation $N_{Z,n}^{((r+r_n),s_n)}$.

Next, we present a simplified example of application of the ideas discussed so far.

5.3.4 An Idealized Example

Suppose, for instance, that we are trying to detect, amongst a finite number of objects $\{X^i\}_{i=1}^L$ belonging to a certain family $\mathcal{A}$, when two objects are isometric. We will assume for simplicity of exposition that we

[16] One more reason for wanting the sub-sampling to be well separated, besides the one given by Corollary 9, is that intuitively, the more separated the covering net, the more efficient the use of the points to cover the metric space.

[17] Another option is choosing p_1 and p_2 in $\mathbb{Z}_m$ at maximal distance and then recurse.

[18] As we mentioned before, the goal can be different: Keep adding points while the separation of the resulting sub-sampling is big enough as measured by some pre-specified constant $s > 0$.

have only two possible cases or hypotheses: either (H1) $d_{\mathcal{GH}}(X^i, X^j) = 0$ or (H2) $d_{\mathcal{GH}}(X^i, X^j) \geq D$ for some $D > 0$ for all $1 \leq i, j \leq L$.

We characterize the family $\mathcal{A}$ as those smooth compact surfaces of $\mathbb{R}^3$ such that their Gaussian curvature is bounded from above by some positive constant K, whose total area is bounded from above by some finite constant A. Then, for any sufficiently small $t > 0$,

$$f_{\mathcal{S}}(t) \overset{\triangle}{=} \min_{x \in \mathcal{S}} \mathbf{a}\left(B_{\mathcal{S}}(x,t)\right) \geq \frac{2\pi}{K}(1 - \cos(t\sqrt{K})) =: F_{K,2}(t)$$

<u>for all</u> $\mathcal{S} \in \mathcal{A}$, by the Bishop-Günther Theorem, see §5.3.2. Note in particular that $F_{K,2}(t) > 0$ for $0 < t < \frac{\pi}{\sqrt{K}}$.

For $1 \leq i \leq L$ we will denote by $\mathbb{X}^i_{m_i}$ the point cloud corresponding to the object X^i and r_i will denote numbers such that $X \subset B(\mathbb{X}^i_{m_i}, r_i)$.[19]

Let X^i and X^j be any two such objects, we will decide, in this example, that X^i and X^j are isometric whenever $\mathcal{L}(\mathbb{X}^i_{m_i}, \mathbb{X}^j_{m_j})$ is smaller than a certain threshold, see Equation (5.22).

Fix $\varepsilon > 0$. For all X^i choose coverings $N^{(R_i,s_i)}_{X^i,n_i}$ such that $\max_{1 \leq i \leq L} R_i \leq \varepsilon$, then n_i will be fixed by the procedure one uses to construct those coverings, see §5.3.3. Let $n := \max_i n_i$ and $R := \min_i R_i \leq \varepsilon$. By adding new points to each of the coverings, if necessary, construct new coverings all with n points, covering radius ε and resulting separation s_i. Let $s := \min_i s_i$. Note that we can estimate n in terms of ε, A and K using the discussion in Remark 23. In fact, $n \geq n_\varepsilon \overset{\triangle}{=} 1 + \left[\frac{A}{F_{K,2}(\varepsilon/2)}\right]$ will do the job.

We are now going to estimate the number of sample points (cardinality of the point clouds) needed for each (all) of the objects in order to be able to detect (H1) with high probability.

According to (5.22), for any $1 \leq i, j \leq L$ we know that:

- Under (H1), with a probability $Q_{ij} := P_{\vec{p}_{X^i}}(n, m_i) \times P_{\vec{p}_{X^j}}(n, m_j)$, we have

$$\mathcal{L}(\mathbb{X}_{m_i}, \mathbb{X}_{m_j}) \leq 3\varepsilon.$$

- Also, under (H2), assuming $\varepsilon \geq r_k$ for $1 \leq k \leq L$,[20]

$$\mathcal{L}(\mathbb{X}_{m_i}, \mathbb{X}_{m_j}) \geq D - 3\varepsilon$$

This tells us how to design ε in relation to D in order to be able to tell both hypotheses apart by computing $\mathcal{L}(\mathbb{X}_{m_i}, \mathbb{X}_{m_j})$. Thus, let $\varepsilon \ll \frac{D}{6}$.[21]

Now, one wants to impose Q_{ij} to be high, that is $Q_{ij} \geq (1-q)^2$ for some small prespecified q. Then, using the comments in Remark 20, we see we can for example fix $c := s$ and estimate the required number of samples for each X_i as $m_i \geq \frac{h_n A}{q \cdot F_{K,2}(\frac{s}{2})}$. In conclusion, one can require that all the point clouds consist of at least $\frac{h_{n_\varepsilon} A}{q \cdot F_{K,2}(\frac{s}{2})}$ points (sampled uniformly) from each of the objects and that all the coverings (constructed using **FPS**) consist of at least n_ε points.

[19] In this example we neglect the fact that this covering relation holds with a certain probability.

[20] This is reasonable since $\mathbb{X}^i_{m_i}$ are supposed to be finer samplings than $N^{(R_i,s_i)}_{X^i,n}$.

[21] What means that all $R_i \ll \frac{D}{6}$.

5.3.5 Computing Geodesic Distances

In our experiments we have always worked with sub-manifolds of $\mathbb{R}^d$. We have used a graph based distance computation following [205], or the exact distance, which can be computed only for certain examples (spheres, planes). We could also use the techniques developed for triangular surfaces in [117], or, being this the optimal candidate, the work on geodesics on (maybe noisy) point clouds developed in [142].

The geodesic computation leads to additional sources of (controllable) errors. We can not compute $d_X(x_i, x_j)$ and $d_Y(y_i, y_j)$ exactly, but rather approximate values $d_X^h(x_i, x_j)$ and $d_Y^{h'}(y_i, y_j)$ for which error bounds are often available [142]. For some suitable function $f(\cdot, \cdot, \cdot, \cdot)$

$$\left|d_X(x_i, x_j) - d_X^h(x_i, x_j)\right| \leq f(h, r, s, n) \tag{5.26}$$

and

$$\left|d_Y(y_i, y_j) - d_Y^{h'}(y_i, y_j)\right| \leq f(h', r', s', n) \tag{5.27}$$

where h and h' control the degrees of approximation. These kind of bounds can be computed for all the approximations we have worked with (see [17], [117]), and also for methods like the one proposed in [142]. We omit in this report the inclusion of this source of errors in our considerations, results in that direction will reported elsewhere.

5.3.6 Additional Implementational Details

In this section we conclude the details on the implementation of the framework here proposed.

The first step of the implementation is the computation of $d_{\mathcal{F}}$ and subsequently $\mathcal{L}$, which from the theory we described before, bounds the Gromov-Hausdorff distance.

We have implemented a simple algorithm.[22] According to the definition of $d_{\mathcal{F}}$, (5.17), given the matrix of pairwise geodesic distances between points of $\mathbb{X}_m$, we need to determine whether there exists a sub-matrix of the whole distance matrix corresponding to $\mathbb{X}_m$ which has a small $d_{\mathcal{I}}$ distance to the corresponding interpoint distance matrix of a given $N_{Y,n}^{(R',s')}$. Since we are free to choose -any coverings-, we select this latter covering net as the result of applying the **FPS** procedure to obtain a subsample consisting of n points, where the first two points are selected to be at maximal distance from each other. We believe that this choice, by the very nature of the **FPS** sampling procedure, produces a set of points with certain particularly interesting metric characteristics. For example, just to motivate our choice, consider the set of the first 7 **FPS** points of a dense point cloud on a crocodile and a dog models shown in Figure 5.2 below.

To fix notation, let $\mathbb{X}_m = \{x_1, \ldots, x_m\}$ and $N_{Y,n}^{(R',s')} = \{y_{j_1}, \ldots, y_{j_n}\}$. We then use the following algorithm.

($k = 1, 2$) Choose x_{i_1} and x_{i_2} such that $|d_X(x_{i_1}, x_{i_2}) - d_Y(y_{j_1}, y_{j_2})|$ is minimized.

($k > 2$) Let $x_{i_{k+1}} \in \mathbb{X}_m$ be such that $e_{k+1}(x_{i_{k+1}}) = \min_{1 \leq i_l \leq m} e_{k+1}(x_{i_l})$ where

$$e_{k+1}(x_{i_l}) = \max_{1 \leq r \leq k} |d_X(x_{i_l}, x_{i_r}) - d_Y(y_{j_{k+1}}, y_{j_r})|$$

[22] This simpler algorithm in turn can be modified to be exhaustive and therefore rigorous, details will provided elsewhere.

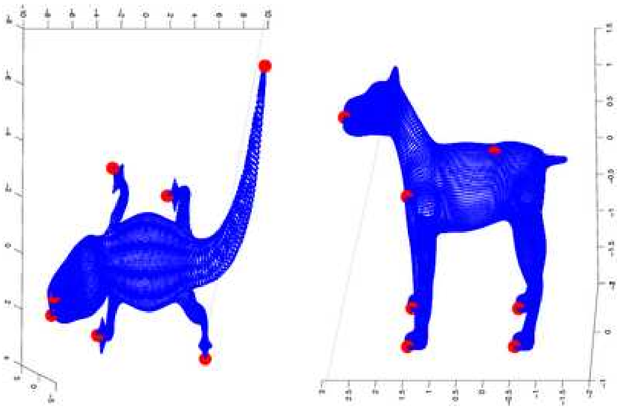


Figure 5.2: **FPS** nets (for $n = 7$) on two point cloud models shown as red balls. Note the perceptual meaningfulness of the point locations automatically chosen by the procedure.

We stop when n points, $\{x_{i_1}, x_{i_2}, \ldots, x_{i_n}\}$ have been selected, and therefore a distance sub-matrix $((d_X(x_{i_u}, x_{i_v})))_{u,v=1}^n$, is obtained.

Since we can write

$$d_{\mathcal{J}}\left(\{x_{i_1}, \ldots, x_{i_n}\}, N_{Y,n}^{(R',s')}\right) \leq \frac{1}{2} \max_{1 \leq k \leq n} \max_{1 \leq t \leq k-1} |d_X(x_{i_k}, x_{i_t}) - d_Y(y_{j_k}, y_{j_t})| = \frac{1}{2} \max_{1 \leq k \leq n} e_k(x_{i_k})$$

we then see that with our algorithm we are trying to decrease the error by working across the sub-diagonal rows of the distance matrix corresponding to $N_{Y,n}^{(R',s')} = \{y_{j_1}, \ldots, y_{j_n}\}$.

Of course, we now use the same algorithm to compute the other half of $d_{\mathcal{F}}$. Note that the set of indices $\{i_1, \ldots, i_n\}$ corresponds to I_n^* as introduced in the definition of Δ_X and subsequently $\mathcal{L}$, Equation (5.23). Therefore, we also obtain an approximation to $\mathcal{L}(\mathbb{X}_m, \mathbb{Y}_{m'})$.

The case $k = 1, 2$ requires visiting all $\frac{m(m-1)}{2}$ interpoint distances between points in $\mathbb{X}_m$. For $k \geq 3$, one must check km different interpoint distances. Then, the complexity of the algorithm is $O(2h_n m + m^2)$. We are currently studying computational improvements along with error bounds for the results provided by the algorithm.

Of course, we still have to prove (or disprove) that the above algorithm, based on **FPS** covering nets, approximates $d_{\mathcal{F}}$ within a reasonable factor. This is subject of current efforts.

5.4 Examples

We now present simple experiments that illustrate the application of the theoretical and computational framework introduced in previous sections. We should note that in these experiments we don't yet exploit the full potential of our theory.[23] For example, to use some of the bounds we need to estimate (or know!) curvature bounds. This estimation could be done using bootstrapping and will be investigated in the future. Also in the future, we plan to make these experiments more rigorous, including concepts of hypothesis testing.

We complemented the more complex data (as presented below) with simple shapes:

[23] For instance, we are not taking into account the probability of covering the shape with the random cloud sampled from it, see (5.25).

[**Plane**] $\Sigma_\pi = [-\frac{\pi}{\sqrt{8}}, \frac{\pi}{\sqrt{8}}]^2$ and $\mathbb{X}_m$ are points sampled uniformly from the square. Note that $\mathbf{diam}\,(X) = \pi$.

[**Sphere**] $S = \{x \in I\!R^d : \|x\| = 1\}$ and $\mathbb{X}_m$ is a set of points uniformly distributed on the sphere. We generated the sample points using the method of Muller, see [207].

5.4.1 Positive Detection

We first test our framework when X and Y are isometric. We consider $X = Y$ and see whether we make the right decision based on the discrete measurements. Let $\mathbb{X}_m$ and $\mathbb{Y}_m$ be two independent sets composed of m independent, uniformly distributed random points on X. We consider X to be either the plane Σ_π or the sphere S as defined above. Given n, using the **FPS** procedure, we construct $N^{(R_X,s_X)}_{\mathbb{X}_m,n}$ and $N^{(R_Y,s_Y)}_{\mathbb{Y}_m,n}$ from $\mathbb{X}_m$ and $\mathbb{Y}_m$, respectively, and look for a metric match inside $\mathbb{X}_m$ and $\mathbb{Y}_m$, respectively, following the algorithm described in §5.3.6 for the computation of $d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_m)$ and subsequently $\mathcal{L}(\mathbb{X}_m, \mathbb{Y}_m)$.[24] For each dataset we tested for values of $m \in M = \{1000, 2000, 2500\}$ and $n \in N = \{20, 40, \ldots, 140\}$, and obtained the results reported below. In Tables 5.1 and 5.2 we show the values obtained for $\mathcal{L}$ for values of $m \in M$ and $n \in N$. As expected, the values of $\mathcal{L}$ are small both when compared to the values reported in next section (for non-isometric shapes, $X = \Sigma_\pi$ and $Y = S$) and when compared to the upper bound $\frac{3\pi}{2} \simeq 4.7124$, as mentioned after Equation (5.22). Note that as a verification, in accordance with formula (5.22) for $d_{\mathcal{GH}}(X,Y) = 0$, we also display the corresponding values of $3\max(R_X, R_Y)$ and the probability, $P_{\vec{p}_X}(n,m) \times P_{\vec{p}_Y}(n,m)$, of having $\mathcal{L} \leq 3\max(R_X, R_Y)$ estimated by using the bound (5.16). Asterisks ($*$) in the tables denote that our lower bound (5.16) was not tight enough to give a positive number. Note that in our experiments we always obtained $\mathcal{L} \leq 3\max(R_X, R_Y)$. This is in some sense a validation of our algorithm for computing $\mathcal{L}$. It also can be interpreted as that there is still room for improvement in the bounds (5.22).

	$m\backslash n$	20	40	60	80	100	120	140
$\mathcal{L}$	1000	0.57077	0.39095	0.32211	0.29971	0.26287	0.24900	0.24900
	2000	0.57216	0.38335	0.31009	0.28149	0.24506	0.24074	0.22770
	2500	0.56942	0.37424	0.30553	0.26125	0.23818	0.24336	0.22468
$3\max(R_X, R_Y)$	1000	1.4929	0.96361	0.71275	0.59935	0.51257	0.48078	0.42079
	2000	1.5607	1.0024	0.75905	0.65428	0.53287	0.49245	0.44269
	2500	1.5626	1.0317	0.76530	0.66509	0.54065	0.50211	0.45791
prob $\geq$	1000	0.83082	0.55117	0.22773	0.044571	$*$	$*$	$*$
	2000	0.91527	0.77526	0.58685	0.44374	0.21043	0.14020	0.039607
	2500	0.93275	0.82145	0.66751	0.56269	0.35018	0.25493	0.13207

Table 5.1: Table with values of $\mathcal{L}$ for $X, Y = P_\pi$ (a plane). See the text for a detailed explanation.

5.4.2 Positive Rejection

We now proceed to compare shapes that are not isometric. We let $X = \Sigma_\pi$ (a plane) and $Y = S$ (a sphere). In this case we expect to be able to detect, based on the finite point clouds, that $\mathcal{L}$ is large, in comparison to the values we obtain in the previous section, when the shapes were actually the same.

[24] Keep in mind that actually $d_{\mathcal{F}}(\mathbb{X}_m, \mathbb{Y}_{m'})$ depends on n, see its definition (5.17).

	$m\backslash n$	20	40	60	80	100	120	140
$\mathfrak{L}$	1000	0.65513	0.50742	0.44578	0.41879	0.38161	0.36817	0.35826
	2000	0.63882	0.49814	0.43230	0.39641	0.38798	0.34721	0.31449
	2500	0.63873	0.51874	0.40902	0.39149	0.34665	0.32906	0.31931
$3\max(R_X, R_Y)$	1000	1.7930	1.2932	0.98264	0.87936	0.78162	0.72061	0.64876
	2000	1.8154	1.3407	1.0688	0.91887	0.83075	0.72182	0.66511
	2500	1.8154	1.3573	1.0264	0.91894	0.79693	0.72379	0.68160
prob $\geq$	1000	0.71395	0.39766	0.085661	0.0017990	$*$	$*$	$*$
	2000	0.82951	0.68052	0.43286	0.31059	0.16894	0.061612	0.0087141
	2500	0.87192	0.71880	0.52749	0.43477	0.27696	0.16941	0.10045

Table 5.2: Table with values of $\mathfrak{L}$ for $X, Y = S$ (a unit sphere). See the text for a detailed explanation.

Table 5.3 shows the results of a simulation in which we compared the sphere S and the plane Σ_π, while varying the covering net sizes and the total number of points uniformly sampled from them ($n \in N$ and $m \in M$ as before). As expected, the values are larger than when comparing plane against plane or sphere against sphere.

	$m\backslash n$	20	40	60	80	100	120	140
$\mathfrak{L}$	1000	1.3191	1.2715	1.1583	1.1739	1.1919	1.1537	1.0849
	2000	1.0751	1.0751	1.0816	1.0991	1.1155	1.1155	1.1305
	2500	1.2059	1.1369	1.1369	1.1471	1.0984	1.1179	1.1179

Table 5.3: Values of $\mathfrak{L}$ for a comparison between Σ_π and S for $n \in N$ and $m \in M$.

5.4.3 $3D$-Shape Recognition

We conclude the experiments with real (more complex) data. We have 4 sets (isometry classes) of shapes,[25] the crocodile $C = \{C_1, C_2\}$, the giraffe $G = \{G_1, G_2\}$, the hand $H = \{H_1, H_2\}$ and the body, $B = \{B_1, B_2\}$. The 2 shapes in each set are bends of each other and therefore isometric. We ran the algorithm with $n = 50$, $m = 2500$ (recall the rough estimate $m \simeq n \log n$ given in Remark 20 and note that with this choices $n \log n \simeq 200 \ll 2500 = m$), using Dijkstra's algorithm to compute geodesic distances.[26] The data description and results are reported in Figure 5.3. Observe that for any fixed shape $X \in C \cup G \cup H \cup B$, the value of $\mathfrak{L}(X, Y)$ is always lower for Y in the same isometry class as X. We note that the technique is not only able to discriminate between different objects but, as expected, doesn't get confused by bends: The distances between a given object and the possible bends of another one are very similar, as it should be the case for isometry invariant recognition.

[25] The datasets were kindly provided to us by Prof. Kimmel and his group at the Technion.

[26] We first considered 10000 points sampled from each of the objects. From each of these these sets we then sub-sampled the 2500 we worked with. For each dataset we used the 10000 points to construct a 15-nearest neighbors graph and then computed the intrinsic distance matrix between the 2500 subset of points using Dijkstra over the whole graph.

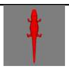
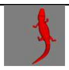
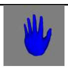
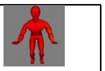
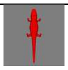
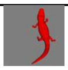
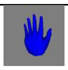
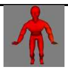
Model								
	1.4071	5.5900	11.769	10.776	11.109	11.111	11.109	11.146
	5.5900	1.4576	12.647	10.196	11.119	11.120	11.123	11.159
	11.769	12.647	1.9357	6.2874	15.169	15.170	15.274	15.207
	10.776	10.196	6.2874	2.0342	14.692	14.693	14.797	14.746
	11.109	11.119	15.169	14.692	0.0045257	0.019845	0.22663	0.19691
	11.111	11.120	15.170	14.693	0.019845	0.0047940	0.23164	0.19715
	11.109	11.123	15.274	14.797	0.22663	0.23164	0.033033	0.096958
	11.146	11.159	15.207	14.746	0.19691	0.19715	0.096958	0.032846
Diameters	22.205	22.222	30.322	29.367	0.040326	0.038832	0.44957	0.37998

Figure 5.3: Comparison results for the complex objects described in §5.4.3. The number of points per model are indicated in the first row under the corresponding figure. The values reported are the estimate of $\mathcal{L}$ between all pairs of objects given by our algorithm. Note that (1) For any object X in this experiment, $\mathcal{L}(X, Y)$ is minimal for Y in the isometry class of X; (2) All objects within the same isometry class have similar values of $\mathcal{L}$ with all other objects belonging to another class.

5.5 Extensions and Conclusions

The comparison framework here introduced opens the doors to extensive research in the area. We conclude this Chapter by presenting some possible directions.

5.5.1 Extensions

The extension to more general metric spaces can be done, in principle, once one agrees upon some definition of uniform probability measure, something that could be done using the Hausdorff Measure, which is defined from the metric.

Another related possible (and easy) extension is that of admitting the points to be sampled from the manifolds with probability measures other than uniform. Actually, in the case of surfaces in $I\!R^3$ acquired by a $3D$-Scanner, the probability measure models the acquisition process itself. In this case, the framework here presented can be extended for a wide family a probability measures, namely those which admit a density function which vanishes at most in sets of 0-uniform measure, i.e., there are no *holes* in the acquisition process.

In other situations it might make more sense to consider the recognition problem for triplets (X, d, μ), where (X, d) is a metric space and μ is a (probability) measure defined on sets of X.

An interesting extension which might make the computational analysis easier would be working with alternative definitions of Hausdorff distance. For example, remembering that the Hausdorff distance between $X, Y \subset Z$, (Z, d, μ) a metric space with metric d (and with probability measure μ) was defined as

$$d_{\mathcal{H}}^{Z}(X,Y) \stackrel{\triangle}{=} \max(\sup_{x\in X} d(x,Y), \sup_{y\in Y} d(y,X)).$$

Then, one can consider substituting each of the supremums inside the $\max(,)$ operation by an L_p-approximation (for $p \geq 1$), for example: $\sup_{x\in X} d(x,Y) \leftrightarrow \left(\int d^p(x,Y)\mu(dx)\right)^{1/p}$, and similarly for the other supremum to obtain, also allowing for a L_q-approximation of the max ($q \geq 1$):

$$d_{\mathcal{H}_{p,q}}^{Z}(X,Y) \stackrel{\triangle}{=} \left(\left(\int d^p(x,Y)\mu(dx)\right)^{q/p} + \left(\int d^p(y,X)\mu(dy)\right)^{q/p}\right)^{1/q}$$

and then the corresponding notion of (p,q)-Gromov-Hausdorff distance is accordingly defined. In particular, it would be interesting to know the corresponding (p,q) version of Property 5 in Proposition 3. Of particular interest in this respect is the very recent work [187].

Scale Dependent Comparisons

In some applications it might be interesting to compare objects in a more local fashion, or in other words, in a scale dependent way. For example, given two objects $\mathcal{S}_1$ and $\mathcal{S}_2$ (with corresponding geodesic distance functions d_1 and d_2) one might wonder whether they resemble each other under the distance $d_{\mathcal{GH}}(,)$ when each of them is endowed with the metric $d_i^{\epsilon} \stackrel{\triangle}{=} \epsilon(1 - e^{-\frac{d_i}{\epsilon}})$, $i = 1, 2$. This choice for the new metrics imposes a scale dependent comparison.

This situation has an important consequence: When ϵ is small enough one might choose to replace d_i by their Euclidean counterparts since, for nearby points x and x' on the sub-manifold $\mathcal{S} \subset \mathbb{R}^k$, $d_{\mathcal{S}}(x,x') \simeq d_{\mathbb{R}^k}(x,x')$. This dispenses with the computational burden of having to approximate the geodesic distance. Also, in a similar vein, in certain applications it may make sense to normalize the distance matrices of all the objects so as to obtain a scale invariant comparison.

5.5.2 Conclusions

A theoretical and computational framework for comparing (smooth, connected and compact) sub-manifolds of $\mathbb{R}^d$ given as point clouds was introduced in this Chapter. The theoretical component is based on the Gromov-Hausdorff distance, which has been embedded in a probabilistic framework to deal with point clouds and computable discrete distances. Examples illustrating this theory were provided.

We are currently working on proving the correctness of the algorithm described in §5.3.6, improving its computational efficiency, performing additional experiments adding hypotheses testing techniques, and in particular, comparing high dimensional point clouds with data from image sciences and neuroscience. These further results and extensions will be reported elsewhere.

Part II

Chapter 6

PDEs and Variational Problems on Implicit Surfaces

6.1 Introduction

In a number of applications in mathematical physics, image processing, computer graphics, and medical imaging, we have to solve variational problems and partial differential equations defined on a general manifold $\mathcal{M}$ (*domain manifold*), which map the data onto another general manifold $\mathcal{N}$ (*target manifold*). That is, we deal with maps from $\mathcal{M}$ to $\mathcal{N}$. When these manifolds are for example three dimensional surfaces, the implementation of the corresponding gradient descent flow or the given PDEs is considerably elaborate. In [134] we have shown how to address this problem for general domain manifolds, while restricting the target manifolds $\mathcal{N}$ to the trivial cases of the Euclidean space or hyper-spheres (this framework has been followed for example in [1]). The key idea was to implicitly represent the domain surface as the (zero) level-set of a higher dimensional function ϕ, and then solve the PDE in the Cartesian coordinate system which contains the domain of this new embedding function. The technique was justified and demonstrated in [134]. It is our goal to show how to work with general target manifolds, and not just hyper-planes or hyper-spheres as previously reported in the literature. Inspired by [134], we also embed the target manifold $\mathcal{N}$ as the (zero) level-set of a higher dimensional function ψ. That is, when solving the gradient descent flow (or in general, the PDE), we guarantee that the map receives its values on the zero level-set of ψ. The map is defined on the whole space, although it never receives values outside of this level-set. Examples of applications of this framework include harmonic maps in liquid crystals ($\mathcal{N}$ is a hypersphere) and 3D surface warping [195]. In this last case, the basic idea is to find a smooth map between two given surfaces. Due to the lack of the new frameworks introduced here and in [134], this problem is generally addressed in the literature after an intermediate mapping of the surfaces onto the plane is performed (see also [113, 216]). With these novel frameworks, direct three dimensional maps can be computed without any intermediate mapping, thereby eliminating their corresponding geometric distortions, see Chapter 8. For this application, as in [195], boundary conditions are needed, this is addressed in Chapter 8.

To introduce the ideas, we concentrate on flat domain manifolds.[1] When combining this framework with

[1] For completeness, we will present the general equations for both generic domain and target manifolds at the end of the

the results on [134], we can of course work with general domains and then completely avoid other popular surface representations, like triangulated surfaces. We are then able to work with intrinsic equations, in Euclidean space and with classical numerics on Cartesian grids, regardless of the geometry of the involved domain and target manifolds. In addition to presenting the general theory, we also address the problem of target submanifolds and open hypersurfaces. A number of theoretical results complement the algorithmic framework here described.

For illustration purposes only, the proposed framework is presented for classical equations from the theory of harmonic maps. The technique could easily be extended to general equations, as it will be clear from the developments below, by using the *dictionary* provided in Appendix B.3. For motivational purposes, in Appendix B.1 we present a simple physical derivation of the Harmonic Maps equation in a simple case.

6.1.1 Why Implicit Representations?

Let us conclude this introduction describing the main reasons and advantages of working with implicit representation when dealing with PDEs and variational problems.

The implicit representation of surfaces, here introduced for solving variational problems and PDEs, is inspired in part by the level-set work of Osher and Sethian [159]. This work, and those that followed it, showed the importance of representing deforming surfaces as level-sets of functions with higher dimensional domains, obtaining more robust and accurate numerical algorithms (and topological freedom). Note that, in contrast with the level-set approach of Osher and Sethian, our target manifold is fixed, what is “deforming” is the dataset being mapped onto it.

Solving PDEs and variational problems with polynomial meshes involves the non-trivial discretization of the equations in general polygonal grids, as well as the difficult numerical computation of other quantities like projections onto the discretized surface (when computing gradients and Laplacians for example). Although the use of triangulated surfaces is quite popular, there is still no consensus on how to compute simple differential characteristics such as tangents, normals, principal directions, and curvatures. On the other hand, it is commonly accepted that computing these objects for iso-surfaces (implicit representations) is simpler and more accurate and robust. This problem becomes even more significant when we not only have to compute these first and second order differential characteristics of the surface, but also have to use them to solve variational problems and PDEs for data defined on the surface. Very little work has been done on the formal analysis of finite difference schemes on non-Cartesian meshes.[2] Note also that working with polygonal representations is dimensionality dependent, and solving these equations for high dimensional (> 2) surfaces becomes even more challenging and significantly less studied. The work here developed is valid for all dimensions of interest (we develop the computational and theoretical framework independently of the manifold dimension). Note that the computational cost of working with implicit representations is not higher than with meshes, since all the work is performed in a narrow band around the level-set(s) of interest.

Our framework of implicit representations enables us to perform all the computations on the Cartesian grid corresponding to the embedding function. These computations are, nevertheless, intrinsic to the surface.

chapter. These equations are easily derived from [134] and the work here presented.

[2] Very important work has been done for finite element approaches, e.g., by the group of Prof. M. Rumpf; as well as for particular equations on particular sub-division representations [14].

Advantages of using Cartesian grid instead of a triangulated mesh include the availability of well studied numerical techniques with accurate error measures and the topological flexibility of the surface, all leading to simple, accurate, robust and elegant implementations. The approach is general (applicable to PDEs and variational problems beyond those derived in this chapter) and dimensionality independent as well. We should note of course that the computational framework here developed is only valid for manifolds which can be represented in implicit form or as intersection of implicit forms. As mentioned above, problems such as 3D shape warping via PDEs could not be addressed (without intermediate projections) without the framework here proposed.

Numerical schemes that solve gradient descent flows and PDEs onto generic target manifolds $\mathcal{N}$ (and spheres or surfaces in particular) will, in general, move the points outside of $\mathcal{N}$ due to numerical errors. The points will then need to be projected back,[3] see for example [4, 171] for the case of $\mathcal{N}$ being a sphere (where the projection is trivial, just a normalization). For general target manifolds, this projection means that for every point $p \in I\!R^d$ ($\mathcal{N} \subset I\!R^d$) we need to know the closest point to p in $\mathcal{N}$. This means knowing the distance from every point $p \in I\!R^d$ to $\mathcal{N}$ (or at least all points in a band of $\mathcal{N}$). This is nothing else than an implicit representation of the target $\mathcal{N}$, being the particular embedding in this case a distance function. This presents additional background for the framework here introduced, that is, if the embedding function for the surface has to be computed anyway for the projection, why not use it from the beginning if it helps in other steps in the computation?

In a number of applications, surfaces are already given in implicit form, e.g., [33], therefore, the framework introduced in this chapter is not only simple and robust, but it is also natural in those applications. Moreover, in the state-of-the-art and most commonly used packages to obtain 3D models from range data, the algorithms output an implicit (distance) function (see for example graphics.stanford.edu/projects/mich/). Therefore, it is very important, if nothing else for completeness, to have the computational framework here developed, so that the surface representation is dictated by the data and the application and not the other way around. On the other hand, not all surfaces (manifolds) are originally represented in implicit form. When the target manifold $\mathcal{N}$ is simple, like hyper-spheres in the case of liquid crystals, the embedding process is trivial. For generic surfaces, we need to apply an algorithm that transforms the given explicit representation into an implicit one. Although this is still a very active area of research, many very good algorithms have been developed, e.g., [63, 174, 122, 215]. Note that this translation needs to be done only once for any surface. Note also that for rendering, the volumetric data can be used directly, without the need for an intermediate mesh representation.

6.2 The Computational Framework

From now on we assume that the target $d-1$ dimensional manifold $\mathcal{N}$ is given as the zero level set of a higher dimensional embedding function $\psi : I\!R^d \to I\!R$, which we consider to be a signed distance function (this mainly simplifies the notation). For the case where $\mathcal{N}$ is a surface in three dimensional space for example, then $\psi : I\!R^3 \to I\!R$. We also assume that the domain manifold $\mathcal{M}$ is flat and open (as mentioned in the introduction, general domain manifolds were addressed in [134]). We illustrate the basic ideas with

[3] For particular flat target manifolds as the whole space $I\!R^d$ or as those in [161], the projection is not needed. Other authors, e.g., [35, 118], have avoided the projection step for particular cases, while in [201] the authors modify the given variational formulation, in some restricted cases, to include the projection step.

a functional from the theory of harmonic maps. This is just a particular example (and a very important one), and from this example it will be clear how the same arguments can be applied to any given variational problem and PDE. In particular, it can be applied to common Navier-Stokes flows used in brain warping.

6.2.1 The Variational Formulation and its Euler-Lagrange

We search for necessary conditions for the functional $E[\vec{u}]$, defined by

$$E[\vec{u}] \stackrel{\triangle}{=} \int_{\mathcal{M}} e[\vec{u}]\, d_{\mathcal{M}} v \tag{6.1}$$

where

$$e[\vec{u}] \stackrel{\triangle}{=} \frac{1}{2} \|\mathbf{J}_{\vec{u}}\|_{\mathcal{F}}^2 \tag{6.2}$$

to achieve a minimum. Here, $\|\cdot\|_{\mathcal{F}}^2 = \sum_{ij} (\cdot)_{ij}^2$ is the norm of Frobenius and $\mathbf{J}_{\vec{u}}$ is the Jacobian of the map $\vec{u} : \mathcal{M} \to \{\psi = 0\}$. Note that here we are already restricting the map to be onto the zero level-set of ψ, that is, onto the surface of interest $\mathcal{N}$ (the target manifold). This is what permits us to work with the embedding function and the whole space, while guaranteeing that the map will always be onto the target manifold, as desired.[4] Once again, this energy will be used throughout this chapter to exemplify our framework. It will be clear after developing this example that the same arguments work for other variational formulations, as well as for generic PDEs defined onto generic surfaces.

Proposition 6 *The Euler-Lagrange of Equation (6.1), with (6.2), is given by*

$$\Delta \vec{u} + \left(\sum_k \mathbf{H}_{\psi} \left[\frac{\partial \vec{u}}{\partial x_k}, \frac{\partial \vec{u}}{\partial x_k} \right] \right) \nabla \psi(\vec{u}) = 0, \tag{6.3}$$

where $\mathbf{H}_{\psi}$ stands for the Hessian of the embedding function ψ (and we used the notation $A\,[\vec{x}, \vec{y}] = \vec{y}^T A\, \vec{x}$). The solution to this equation is a map onto the zero level-set of ψ.

Proof: The proof is based on adding to the classical techniques to compute Euler-Lagrange equations a projection step that guarantees that the perturbation keeps the map onto $\{\psi = 0\}$.

Assume that $\vec{u}$ is a map minimizing $E(\cdot)$. Given $t > 0$, we construct the variation

$$\vec{v}_t \stackrel{\triangle}{=} \vec{u} + t\, \vec{r}$$

where $\vec{r}$ is a compact C^∞ map in $\mathcal{M}$. For an arbitrary $x \in \mathcal{M}$, we will in general not obtain that $\vec{v}_t(x) \in \{\psi = 0\}$ for all t and x. That is, $\psi(\vec{v}_t(x)) \neq 0$ at some (t, x). Therefore, this variation is not admissible. On the other hand, we can from it construct an admissible variation via

$$\vec{w}_t \stackrel{\triangle}{=} \mathbf{\Pi}_{\{\psi=0\}}(\vec{v}_t)$$

where $\mathbf{\Pi}_{\{\psi=0\}} : R^d \to \{\psi = 0\}$ is the *projection operator* onto $\{\psi = 0\}$. Note that since ψ is a signed distance function, we can simply write this projection operator onto $\{\psi = 0\}$ as

[4] We use $\vec{\cdot}$ to note that for the most general case, the function is vectorial.

$$\mathbf{\Pi}_{\{\psi=0\}}(\vec{\alpha}) = \vec{\alpha} - \psi(\vec{\alpha})\ \nabla\psi(\vec{\alpha}).$$

Let's now define

$$\mathcal{E}(t) \triangleq E[\vec{w}_t]$$

Since the energy achieves a minimum for $t = 0$,

$$\dot{\mathcal{E}}_0 \triangleq \left.\frac{dE(t)}{dt}\right|_{(t=0)} = 0.$$

Let's compute this first variation. We have that

$$\dot{\mathcal{E}}_0 = \sum_{ij} \int_{\mathcal{M}} \left.\left(\frac{\partial w_t^i}{\partial x_j} \frac{d\left(\frac{\partial w_t^i}{\partial x_j}\right)}{dt} \right)\right|_{t=0} d_{\mathcal{M}} v \tag{6.4}$$

Moreover (recall that $\mathbf{H}_\psi$ stands for the Hessian of ψ),

$$\begin{aligned} \frac{\partial w_t}{\partial x_j} &= \left(\frac{\partial \vec{u}}{\partial x_j} + t\frac{\partial \vec{r}}{\partial x_j} \right) - \left(\nabla\psi(\vec{w}_t) \cdot \left(\frac{\partial \vec{u}}{\partial x_j} + t\frac{\partial \vec{r}}{\partial x_j} \right) \right) \nabla\psi(\vec{w}_t) \\ &- \psi(\vec{w}_t)\, \mathbf{H}_\psi(\vec{w}_t) \left(\frac{\partial \vec{u}}{\partial x_j} + t\frac{\partial \vec{r}}{\partial x_j} \right) \end{aligned} \tag{6.5}$$

and we observe that

$$\left.\frac{\partial w_t}{\partial x_j}\right|_{(t=0)} = \frac{\partial \vec{u}}{\partial x_j} - \left(\nabla\psi(\vec{u}) \cdot \frac{\partial \vec{u}}{\partial x_j} \right) \nabla\psi(\vec{u})$$

since $\psi(\vec{u}) = 0$. We can further simplify this observing that $0 = \frac{\partial\psi(\vec{u})}{\partial x_j} = \nabla\psi(\vec{u}) \cdot \frac{\partial \vec{u}}{\partial x_j}$. Therefore,

$$\left.\frac{\partial w_t}{\partial x_j}\right|_{(t=0)} = \frac{\partial \vec{u}}{\partial x_j} \tag{6.6}$$

With a bit further work we can compute the additional derivative, $\frac{d\left(\frac{\partial w_t^i}{\partial x_j}\right)}{dt} = \frac{\partial\left(\frac{dw_t^i}{dt}\right)}{\partial x_j}$. This change in the order of derivatives is done in order to immediately evaluate the result at $t = 0$, thereby simplifying the following derivative. Following in an similar fashion, we obtain

$$\frac{dw_t^i}{dt} = \vec{r} - (\nabla\psi(\vec{w}_t) \cdot \vec{r})\, \nabla\psi(\vec{w}_t) - \psi(\vec{w}_t)\ \mathbf{H}_\psi(\vec{w}_t)\vec{r} \tag{6.7}$$

and

$$\left.\frac{dw_t^i}{dt}\right|_{(t=0)} = \vec{r} - (\nabla\psi(\vec{u}) \cdot \vec{r})\, \nabla\psi(\vec{u}). \tag{6.8}$$

Combining the above computations all together we obtain

$$
\begin{aligned}
\left.\frac{d\left(\frac{\partial w_t^i}{\partial x_j}\right)}{dt}\right|_{(t=0)} &= \frac{\partial\left(\left.\frac{dw_t^i}{dt}\right|_{(t=0)}\right)}{\partial x_j} &(6.9)\\
&= \frac{\partial \vec{r}}{\partial x_j} - \nabla\psi(\vec{u})\left\{\frac{\partial \vec{r}}{\partial x_j}\cdot\nabla\psi(\vec{u}) + \mathbf{H}_\psi\left(\vec{r},\frac{\partial \vec{u}}{\partial x_j}\right)\right\} - (\vec{r}\cdot\nabla\psi(\vec{u}))\left(\mathbf{H}_\psi\,\frac{\partial \vec{u}}{\partial x_j}\right).
\end{aligned}
$$

Following from (6.4) we have that[5]

$$
\begin{aligned}
\dot{\mathcal{E}}_0 &= \sum_j \int_{\mathcal{M}} \left.\left(\frac{\partial \vec{w}_t}{\partial x_j}\frac{d\left(\frac{\partial \vec{w}_t}{\partial x_j}\right)}{dt}\right)\right|_{t=0} d_{\mathcal{M}}v &(6.10)\\
&= \sum_j \int_{\mathcal{M}} \left\{\frac{\partial \vec{r}}{\partial x_j}\cdot\frac{\partial \vec{u}}{\partial x_j} - (\vec{r}\cdot\nabla\psi(\vec{u}))\mathbf{H}_\psi\left[\frac{\partial \vec{u}}{\partial x_j},\frac{\partial \vec{u}}{\partial x_j}\right]\right\} d_{\mathcal{M}}v.
\end{aligned}
$$

Now, applying the divergence theorem we conclude the computation. We first write

$$
\sum_{ij}\int_{\mathcal{M}} \frac{\partial \vec{r}}{\partial x_j}\cdot\frac{\partial \vec{u}}{\partial x_j}\; d_{\mathcal{M}}v = \sum_i \int_{\mathcal{M}} \nabla r^i\cdot\nabla u^i\; d_{\mathcal{M}}v
$$

and then apply the fact $\nabla r^i\cdot\nabla u^i = \nabla\cdot\left(r^i\nabla u^i\right) - r^i\,\Delta u^i$, together with the divergence theorem, to obtain ($\mathbf{n}$ stands for the outward unit normal to $\partial\mathcal{M}$).

$$
\sum_{ij}\int_{\mathcal{M}} \frac{\partial \vec{r}}{\partial x_j}\cdot\frac{\partial \vec{u}}{\partial x_j}\; d_{\mathcal{M}}v = \sum_i \int_{\partial\mathcal{M}} r^i\frac{\partial u^i}{\partial \mathbf{n}}dS - \int_{\mathcal{M}} r^i\Delta u^i d_{\mathcal{M}}v \qquad (6.11)
$$

To conclude we put together this last expression with (6.9), and after some algebra we obtain that $\dot{\mathcal{E}}_0$ is equal to

$$
\int_{\partial\mathcal{M}} \vec{r}\cdot\mathbf{J}_{\vec{u}}\mathbf{n}\; dS - \int_{\mathcal{M}} \vec{r}\cdot\left\{\Delta\vec{u} + \left(\sum_k \mathbf{H}_\psi\left[\frac{\partial \vec{u}}{\partial x_k},\frac{\partial \vec{u}}{\partial x_k}\right]\right)\nabla\psi(\vec{u})\right\} d_{\mathcal{M}}v \qquad (6.12)
$$

The boundary condition is eliminated since the support of $\vec{r}$ is compactly included in $\mathcal{M}$. To eliminate the additional term for an arbitrary $\vec{r}$ we must impose

$$
\Delta\vec{u} + \left(\sum_k \mathbf{H}_\psi\left[\frac{\partial \vec{u}}{\partial x_k},\frac{\partial \vec{u}}{\partial x_k}\right]\right)\nabla\psi(\vec{u}) = 0. \qquad (6.13)
$$

□

Equation (6.3) (or (6.13)) then gives the corresponding Euler-Lagrange for the given variational problem. Note, once again from our computations, that despite all the terms "live" in the Euclidean space where the target manifold is embedded, $\vec{u}$ will always map onto the level-set of interest, $\{\psi = 0\}$, and therefore, onto the surface of interest. This is guaranteed by this equation, no additional computations are needed. This

[5] We have used as before the notation $A[\vec{x},\vec{y}] = \vec{y}^T A\,\vec{x}$

is the beauty of the approach, while working freely on the Euclidean space (and therefore with Cartesian numerics), we can guarantee that the equations are intrinsic to the given surfaces of interest. We will further verify this in §6.2.3 to help the reader grasp the intuition behind this framework. In the same section we present a particular example of the above equation for a target surface given by a hypersphere.

The Gradient-Descent Flow

The gradient descent corresponding to (6.13) is given by

$$\frac{\partial u^i}{\partial t} = \Delta u^i + \sum_{k=1}^{d} \mathbf{H}_\psi(\vec{u}) \left[\frac{\partial \vec{u}}{\partial x_k}, \frac{\partial \vec{u}}{\partial x_k} \right] \frac{\partial \psi}{\partial u^i}(\vec{u}), \tag{6.14}$$

where the initial datum $\vec{u}_0$ is given by the vector field we want to process, together with *Neumann* boundary conditions:

$$\begin{cases} \vec{u}(x,0) = \vec{u}_0(x), \;\; x \in \mathcal{M} \\ \mathbf{J}_{\vec{u}}\mathbf{n}|_{\partial\mathcal{M}} = \mathbf{0}. \end{cases} \tag{6.15}$$

To complete the picture, the use of *Neumann* boundary conditions needs to be justified. This is done in Appendix B.2.

6.2.2 Connections with Harmonic Maps

The goal of this section is to illustrate the connections of the equations above with the well known theory of harmonic maps. As it is the case of the proof of Proposition 6, these connections are simple to derive, as we do below. Nevertheless, the derivations themselves present illustrative calculus with implicit surfaces and PDEs on them.

The expressions derived in previous sections come from the theory of harmonic maps, e.g., [92, 213, 50, 136, 64, 65, 137, 98, 166, 186, 185, 12]. In general, harmonic maps are defined as those maps between two manifolds $(\mathcal{M}, g)$ and $(\mathcal{N}, h)$ which minimize the energy

$$E[\vec{u}] \triangleq \int_{\mathcal{M}} e[\vec{u}] \; dV_{\mathcal{M}} \tag{6.16}$$

where, in local coordinates, the *energy density* $e[\vec{u}]$ is given by

$$e[\vec{u}](x) \triangleq \frac{1}{2} \; g^{pq}(x) \; h_{ij}(\vec{u}(x)) \; \frac{\partial u^i}{\partial x_p} \frac{\partial u^j}{\partial x_q}. \tag{6.17}$$

We have used Einstein's summation here, where repeated indices indicate summation with respect to this index, together with the usual notation for tensors.[6] When both the domain and target manifolds are represented explicitly, the classical case, the Euler-Lagrange equation corresponding to this energy is given by (see [186])

$$\Delta_{\mathcal{M}} u^l + \mathbf{\Gamma}^l_{ij}(\vec{u}) \, g^{\alpha\beta} \frac{\partial u^i}{\partial x^\alpha} \frac{\partial u^j}{\partial x^\beta} = 0 \tag{6.18}$$

[6] $(g^{-1})_{ij} \triangleq g^{ij}$.

where $\Delta_{\mathcal{M}}$ is the Laplace-Beltrami operator (reduced to the regular Laplacian for the case of flat domain manifolds) and $\mathbf{\Gamma}^l_{ij}(\vec{u})$ stands for the Christoffel symbols of the target manifold, evaluated at $\vec{u}$. Note that the first component, the Laplace-Beltrami of u, addresses the domain manifold, whereas the second term addresses the target manifold. By embedding the target manifold, we are changing the Christoffel symbols (expressing them in implicit form, see below),[7] while the work in [134] changed the other terms, since the embedding was done to the domain manifold, see §6.4. The framework here introduced can then be seen as the re-writing of given PDEs mapping manifolds to manifolds in such a way that the intrinsic geometric characteristics of the equation are expressed using the embedding functions.

As an example, let's see what happens with the above energy for the Euclidean case. Since both metrics are proportional to the identity,

$$e[\vec{u}](x) = \frac{K}{2} \sum_{ij} \left(\frac{\partial u^i}{\partial x_j} \right)^2$$

which is just a constant multiplying $\|\mathbf{J}_{\vec{u}}\|^2_{\mathcal{F}}$. Therefore, the energy defined in the previous case is just a particular case of harmonic maps. In general, this energy can be used in problems such as color image denoising and directions denoising [12, 13], as a regularization term for ill-possed problems defined on general surfaces [156], for general denoising [152, 198], for models of liquid crystals, and as a component of a system for surface mapping and matching [136, 216].

An(other) Informal Calculation

We now present an additional computation that connects in a deep way the implicit framework with harmonic maps. We consider the harmonic energy density given in (6.17) for the planar domain manifold case ($g_{ij} = \delta_{ij}$). We can simplify things to obtain

$$e[\vec{u}](x) = \frac{1}{2}\; h_{ij}(\vec{u}(x))\; \frac{\partial u^i}{\partial x_p} \frac{\partial u^j}{\partial x_p} = \frac{1}{2} \sum_p \mathbf{h}[\vec{u}_{x_p}, \vec{u}_{x_p}]$$

We know that $\mathbf{\Pi}_{\nabla\psi} = \mathbf{I} - \nabla\psi\nabla\psi^T$ can be thought of as the inverse of the target manifold's *metric tensor*. But since $\nabla\psi$ is a zero eigenvalue eigenvector for $\mathbf{\Pi}_{\nabla\psi}$, it will be a ∞ eigenvalue eigenvector for $\mathbf{\Pi}^{-1}_{\nabla\psi}$. Then, we can't use the identification $\mathbf{h} = (h_{ij}) \leftrightarrow \mathbf{\Pi}^{-1}_{\nabla\psi}$ in the above expression for the energy density. However, we can proceed as follows. Take $\epsilon > 1$ and define the metric[8]

$$\mathbf{h}^\epsilon \overset{\triangle}{=} \left(\epsilon\mathbf{I} - \nabla\psi\nabla\psi^T \right)^{-1}$$

one can then compute the inverse as (it's an elementary formula, see for example [111])

$$\mathbf{h}^\epsilon = \frac{1}{\epsilon} \left(\mathbf{I} + \frac{\nabla\psi\nabla\psi^T}{\epsilon - 1} \right)$$

The energy density can be rewritten as (we will use a subindex ϵ)

[7] Or alternatively, the second fundamental form of the target manifold.

[8] Since $\epsilon > 1$, all the eigenvalues are positive.

$$e_\epsilon[\vec{u}](x) = \frac{1}{2\epsilon}\left(\sum_i \|\vec{u}_{x_i}\|^2 + \frac{1}{\epsilon-1}\sum_i |\vec{u}_{x_i}\cdot\nabla\psi|^2\right)$$

After computing the variational derivative for the functional $\int_{\mathcal{M}} e_\epsilon[\vec{u}](x)\,dx$ we obtain that $\vec{u}$ must satisfy

$$\Delta\vec{u} + \frac{1}{\epsilon-1}\left(\sum_i \mathbf{H}_\psi[\vec{u}_{x_i},\vec{u}_{x_i}] + \Delta\vec{u}\cdot\nabla\psi\right)\nabla\psi = 0$$

By multiplying all the terms in the above equation by $\epsilon-1$ and letting $\epsilon\to 1$ we find that the expression between brackets must vanish. As we will see in §6.2.3, what's between brackets is nothing but $\Delta\nu(x)$ where $\nu(x) = \psi(\vec{u}(x))$. So ν is a harmonic function in $\mathcal{M}$. It is also evident that ν satisfies Dirichlet boundary conditions if $\vec{u}$ does, and since we are trying to map things from $\mathcal{M}$ to $\mathcal{N}$, those boundary conditions for $\vec{u}$ must be such that $dist(\vec{u}(x),\mathcal{N}) = 0$ for $x\in\mathcal{M}$, so $\nu|_{\partial\mathcal{M}} = 0$. Then we conclude that ν must be zero everywhere in $\mathcal{M}$.

6.2.3 Simple Verifications

We now show that the Euler-Lagrange (6.13), and its corresponding gradient descent flow (6.14), are the extension for implicit targets of common equations derived in the literature for explicitly represented manifolds. We also explicitly show that the flow equation guarantees, as expected from the derivations above and in particular from the proof of Proposition 6, that if the initial datum is on the target manifold, it will remain on it. We also express the second fundamental form of a manifold that is implicitly represented. All these results will help to further illustrate the approach and verify its correctness.

Geodesics on Implicit Manifolds

It is well known, see [64, 65, 175], that arc-length parameterized geodesics on the manifold $\mathcal{N}$ satisfy the harmonic maps PDE, and therefore Equation (6.3). If we assume isotropic and homogeneous metric over $\mathcal{N}$, from Equation (6.3) we obtain that (arc-length parameterized) geodesics must satisfy

$$\ddot{\gamma} + \mathbf{H}_\psi\left[\dot{\gamma},\dot{\gamma}\right]\nabla\psi(\gamma) = 0. \tag{6.19}$$

This important equation shows how to obtain geodesic curves on manifolds represented in implicit form.

Liquid Crystals ($\mathcal{N} = S^{d-1}$)

One of the most popular examples of harmonic maps is given when the target manifold $\mathcal{N}$ is a hypersphere. That is, the map is onto S^{d-1}. In this case, the embedding (signed distance) function is simply $\psi(\vec{y}) = \|\vec{y}\| - 1$, $\vec{y}\in I\!R^d$. From this, $\nabla\psi(\vec{y}) = \frac{\vec{y}}{\|\vec{y}\|}$ and $(\mathbf{H}_\psi(\vec{y}))_{ij} = \frac{\delta_{ij}}{\|\vec{y}\|} - \frac{y_i y_j}{\|\vec{y}\|^3}$. We also have that $\mathbf{H}_\psi(\vec{u}(x))\left[\frac{\partial\vec{u}}{\partial x_k},\frac{\partial\vec{u}}{\partial x_k}\right] = \delta_{ij}\frac{\partial u^i}{\partial x_k}\frac{\partial u^j}{\partial x_k} - \frac{\partial u^i}{\partial x_k}\frac{\partial u^j}{\partial x_k}u_i u_j$, since $\|\vec{u}\| = 1$. In addition, $u^i\frac{\partial u^i}{\partial x_k} = 0$, fact simply obtained taking derivatives with respect to x_k. We then obtain that $\frac{\partial u^i}{\partial x_k}\frac{\partial u^j}{\partial x_k}u_i u_j = \left(\frac{\partial u^i}{\partial x_k}u_i\right)^2 = 0$, and $\sum_{k=1}^{d}\mathbf{H}_\psi(\vec{u}(x))\left[\frac{\partial\vec{u}}{\partial x_k},\frac{\partial\vec{u}}{\partial x_k}\right] =$

$\sum_{ik}\left(\frac{\partial u^i}{\partial x_k}\right)^2 = \|\mathbf{J}_{\vec{u}}(x)\|^2_{\mathcal{F}}$. Therefore, the corresponding diffusion equation from (6.14) is

$$\frac{\partial \vec{u}}{\partial t} = \Delta \vec{u} + \|\mathbf{J}_{\vec{u}}\|^2_{\mathcal{F}}\ \vec{u}$$

which is exactly the well known gradient descent flow for this case. We have then verified the correctness of the derivation in Proposition 6 for the case of unit spheres as target manifolds.

Diffusion of Probabilities

In this case, $\mathcal{N} = \{x \in I\!R^d |\ x_i \geq 0,\ \sum_{i=1}^d x_i = 1\}$ which is not a closed manifold. However, by maximum principle arguments, if the initial datum is on $\mathcal{N}$, it will remain there for all time of smooth existence, see §6.2.5 and [161]. Then, we can formally consider $\psi(x) = \frac{\sum_{i=1}^d x_i - 1}{\sqrt{d}}$, the signed distance from a point $x \in I\!R^d$ to the hyperplane $\{z \in I\!R^d |\ \sum_{i=1}^d z_i = 1\}$, where the sign was selected accordingly to our choice of $\frac{(1,\ldots,1)}{\sqrt{d}}$ as the unit normal to the hyperplane. We then obviously obtain $\nabla\psi(x) = \frac{(1,\ldots,1)}{\sqrt{d}}$ and $\mathbf{H}_\psi(x) = 0$ for all x. Consequently, the evolution equation for this case is

$$\vec{u}_t = \Delta \vec{u}$$

as expected, [161].

Mapping Restriction onto the Zero Level-Set

We now explicitly show that if the initial datum belongs to the target surface given by the zero level-set of ψ, then the solution to the diffusion flow (6.14) also belongs to this level-set. This further shows the correctness of our approach.

Proposition 7 *A regular solution to Equation (6.14) holds* $\psi(\vec{u}(x,t)) = 0,\ \forall\, x \in \mathcal{M},\ \forall\, t \geq 0$ *of regularity.*

Proof: If the initial datum is on $\{\psi = 0\}$, then this property is true for $t = 0$. Let's define $\nu(x,t) = \psi(\vec{u}(x,t))$. Then[9]

$$\frac{\partial \nu}{\partial t} = \nabla\psi(\vec{u}) \cdot \frac{\partial \vec{u}}{\partial t} = \Delta\vec{u} \cdot \nabla\psi(\vec{u}) + \sum_{k=1}^d \mathbf{H}_\psi(\vec{u}) \left[\frac{\partial \vec{u}}{\partial x_k}, \frac{\partial \vec{u}}{\partial x_k}\right]$$

since ψ is a distance function. In addition, $\frac{\partial \nu}{\partial x_i} = \nabla\psi(\vec{u}) \cdot \frac{\partial \vec{u}}{\partial x_i}$, and then

$$\frac{\partial^2 \nu}{\partial x_i^2} = \left(\mathbf{H}_\psi(\vec{u}) \frac{\partial \vec{u}}{\partial x_i}\right) \cdot \frac{\partial \vec{u}}{\partial x_i} + \nabla\psi(\vec{u}) \cdot \frac{\partial^2 \vec{u}}{\partial x_i^2}.$$

Adding on $i = 1, ..., d$, it follows that $\frac{\partial \nu}{\partial t} = \Delta\nu$, meaning that ν verifies the heat flow. In addition to this, $\frac{\partial \nu}{\partial \mathbf{n}}|_{\partial\mathcal{M}} = \nabla_x\left(\psi(\vec{u}(x,t))\right) \cdot \mathbf{n} = \mathbf{J}^T_{\vec{u}} \nabla\psi(\vec{u}) \cdot \mathbf{n} = (\nabla\psi(\vec{u}))^T \mathbf{J}_{\vec{u}} \mathbf{n} = (\nabla\psi(\vec{u}))^T \mathbf{0} = 0$, due to the boundary conditions on the evolution of $\vec{u}$.

[9] The calculations that follow in the proof don't take into account that ψ might fail to be differentiable at some points. This could be simply addressed by a regularization argument. Moreover, we use the fact that there exists $T > 0$ such that $\vec{u}$ is regular in $[0, T)$, see below.

We have then obtained that ν verifies the heat flow with *Neumann* boundary conditions and with zero initial data. From the uniqueness of the solution, it follows that $\nu(x,t) = 0 \; \forall x \in \mathcal{M}, \; \forall t \geq 0$.
□

Second Fundamental Form for Implicit Surfaces

If we compare the gradient descent flow (and Euler-Lagrange equation) we have obtained with the classical one from harmonic maps, we see that the main difference is that Christoffel symbols for the target manifold term appearing in the classical formulation have been replaced by a new term that includes the Hessian of the embedding function. We obtained this by first embedding the target manifold and then restricting the search for the minimizing map to the class of maps onto the zero level-set of the embedding function. This approach can be followed to apply this framework to any related variational problem. We now show how the same equation can be obtained by simply substituting the second fundamental form of the explicit target manifold by the corresponding expression for an implicit target manifold. This will illustrate how to apply our framework to more general PDEs, not necessarily gradient descent flow. The basic idea is just to replace all the PDE components concerning the target manifold by their counterparts for implicit representations.

In [130] (page 150) it is shown that the *scalar second fundamental form h* at a point p of a hypersurface $\mathcal{S}$ can be written in the form

$$h(p)(V,W) = \frac{\mathbf{H}_\psi(p)[V,W]}{\|\nabla\psi\|^2}$$

for V, $W \in T_p\mathcal{S}$. According to [130] (page 139) the *vectorial second fundamental form* is given by

$$I\!I(p)(V,W) = h(p)(V,W)\,\frac{\nabla\psi}{\|\nabla\psi\|}$$

From (6.18) and what we have just seen it is obvious that the *implicit* version of the *harmonic map* Euler-Lagrange is (6.13).

As stated before, the implicit representation of the target surface permits then to compute the second fundamental form using differences on Cartesian grids, without the need to develop new numerical techniques on polygonal grids.

From the result just presented, in order to transform a given PDE into its counterpart when the target manifold is represented in implicit form, all that needs to be done is to re-write all the characteristics of the PDE, concerning this target manifold, in implicit form. For completeness, in Appendix B.3 we present basic facts on calculus on implicitly represented hyper-surfaces.

6.2.4 Explicit Derivation of the Diffusion Flow

Here we first proceed in a naïve way to obtain an equivalent formulation of the gradient descent flow that will help in the numerical implementation. We assume we have a family $\{\vec{u}(\vec{x},t)\}_t$ of mappings from Ω to $\mathcal{N}$. For each t we define the *harmonic energy* of a member of the family as

$$E(t) = \frac{1}{2}\int_\Omega \|\mathbf{J}_{\vec{u}(\vec{x},t)}\|_F^2 \; dx$$

We then find a variation of the family such that $E(t)$ decreases. To accomplish this we formally differentiate the energy with respect to t. A simple computation yields

$$\dot{E}(t) = -\int_{\Omega} \vec{u}_t \cdot \Delta\vec{u}\; dx$$

Now, since $\vec{u}(\vec{x},t) \in \mathcal{N}\;\forall\,\vec{x} \in \Omega$ and $\forall\, t$ of smooth existence, one must have $\vec{u}_t(\vec{x},t) \in T_{\vec{u}(\vec{x},t)}\mathcal{N}$. An appropriate choice for $\vec{u}_t$ would be

$$\vec{u}_t \;=\; \mathbf{\Pi}_{T_{\vec{u}(\vec{x},t)}\mathcal{N}}\big(\Delta\vec{u}\big) \tag{6.20}$$

since this makes $\dot{E}(t) = -\int_{\Omega} \|\vec{u}_t\|^2\, dx \;\leq\; 0$.

The projection operator in (6.20), as we already know (see Appendix B.3), can be expressed in a very simple form using ψ (the signed distance function to $\mathcal{N}$),

$$\mathbf{\Pi}_{T_{\vec{p}}\mathcal{N}}\big(\vec{v}\big) = \vec{v} \,-\, \vec{v}\cdot\nabla\psi(p)\,\nabla\psi(p). \tag{6.21}$$

Now, it should happen that (6.20) is *equivalent* to (6.14). We show this in §6.5.

6.2.5 Remarks on the Solutions of the Diffusion Flow

In previous subsections we have derived novel equations for PDEs mapping into target manifolds. We complete the work of this Section with relevant results from the literature on the mathematical correctness of these equations.

The well posedness of these diffusion problem with *Neumann* boundary conditions is addressed in [96, 154], where the following results are obtained, here included for completeness:

Theorem 11 *For a given C^∞ mapping $\vec{u}_0 : \mathcal{M} \to \mathcal{N} \subset I\!R^{n+1}$ with $\frac{\partial\vec{u}_0}{\partial\mathbf{n}} = 0$ on $\partial\mathcal{M}$ and for every $2 + dim(\mathcal{M}) < p < +\infty$ there exists an $\epsilon > 0$ (depending on $\vec{u}_0$) and a mapping $\vec{u} : \mathcal{M} \to \mathcal{N}$ of class $\mathbf{L}^p_2(\mathcal{M}\times[0,\epsilon], I\!R^{n+1})$.* [10] *Moreover, $\vec{u}$ is unique and C^∞ except at the corner $\partial\mathcal{M}\times\{0\}$.*

Theorem 12 *Let $(\mathcal{M}, g)$ and $(\mathcal{N}, h)$ be compact Riemmanian Manifolds with convex boundary. Let $\vec{u} : \mathcal{M}\times[0,\omega) \to \mathcal{N}$ be a maximal solution of the diffusion problem with initial value a C^∞ mapping $\vec{u}_0$,*[11] *with $\chi_0 \overset{\triangle}{=} \|e[\vec{u}_0]\|_{\mathbf{L}^\infty} > 0$. Let $r \in I\!R$ be such that $Ric_{\mathcal{M}} \geq -\frac{r}{2}\, g$,*[12] *and $R \geq 0$ such that all sectional curvatures of $\mathcal{N}$ are not greater than $\frac{R}{4}$. Then,*

1. *In the case $r + R\chi_0 > 0$*

   (a) *if $R > 0$ then* $\begin{cases} \omega \geq \frac{1}{r}\log(1 + \frac{r}{R\chi_0}) & \textit{when } r \neq 0 \\ \omega \geq \frac{1}{R\chi_0} & \textit{when } r = 0 \end{cases}$

   (b) *if $R = 0$ then $\omega = +\infty$.*

2. *In the case $r + R\chi_0 \leq 0$, $\omega = +\infty$.*

[10] $\mathbf{L}^p_2(\mathcal{M}\times[0,\epsilon], I\!R^{n+1})$ is the space of functions $f : \mathcal{M} \to I\!R^{n+1}$ such that for every $i = 1, \ldots, n+1$, $\nabla_{\mathcal{M}} f^i$, $\mathbf{H}^{\mathcal{M}}_{f^i}$ and $\frac{\partial f^i}{\partial t}$ are all in $\mathbf{L}^2(\mathcal{M}\times[0,\epsilon])$.

[11] A solution $\vec{u} : \mathcal{M}\times[0,\omega) \to \mathcal{N}$ of the diffusion problem is *maximal* if it cannot be extended to be a solution on $\mathcal{M}\times[0,\omega+\epsilon)$ for any $\epsilon > 0$ or if $\omega = +\infty$.

[12] $Ric_{\mathcal{M}}$ stands for the Ricci curvature tensor of $\mathcal{M}$.

Maps into Open Surfaces

So far, we have only addressed the case when the target surface is closed (zero level-set). We now briefly deal with open surfaces. We show, following classical results, that when the map $\vec{u}$ is evolving according to the flow in §6.2.1, the set $\mathcal{C}(t) \overset{\triangle}{=} \{\vec{u}(x,t),\ x \in \mathcal{M}\}$ remains inside the initial *convex-hull* of $\mathcal{C}_0 \overset{\triangle}{=} \{\vec{u}_0(x),\ x \in \mathcal{M}\}$, $\forall\, t \geq 0$. This property is basically a consequence of the maximum principle. In the actual computations, this might of course be violated due to numerical errors, and we will later discuss how to correct for this as well.

Let us first motivate the general result presented below for the planar case. Assume that the target manifold $\mathcal{N}$ is flat, for example R^k (we still assume that the domain manifold $\mathcal{M}$ is flat). Let $\vec{u}(x,t)$ solve $\frac{\partial \vec{u}}{\partial t} = \Delta \vec{u}$ for $x \in \mathcal{M}$ and $t \geq 0$, and $\frac{\partial \vec{u}}{\partial \mathbf{n}}|_{\partial \mathcal{M}} = 0$. Let Ξ be a convex set of R^k with smooth boundary (this guarantees that the distance function is also smooth almost everywhere, see [175] for a formal statement), and ξ the signed distance function to this set (positive outside and negative inside). Define $g(x,t) \overset{\triangle}{=} \xi(\vec{u}(x,t))$. Then it follows that $\frac{\partial g}{\partial t} - \Delta g = -\sum_{i=1}^{k} \mathbf{H}_\xi(\frac{\partial \vec{u}}{\partial x_k}, \frac{\partial \vec{u}}{\partial x_k})$.[13] Since Ξ is convex, so it is ξ. Then, the *Hessian* of ξ is *positive semi-definite*, meaning that $\frac{\partial g}{\partial t} - \Delta g \leq 0$. Following the scalar maximum principle, $\max_{\{x \in \mathcal{M},\, t \geq 0\}} g(x,t) = \max_{\{x \in \mathcal{M}\}} g(x,0)$. If $\{\vec{u}_0(x),\, x \in \mathcal{M}\} \subseteq \Xi$, which is equivalent to $0 \geq \xi(\vec{u}_0(x)) = g(x,0)$, we obtain that $g(x,t) \leq 0$, and $\vec{u}(x,t) \in \Xi$, for all $x \in \mathcal{M}$ y $t \geq 0$.

The general result now presented is from [96]. We quote it here for completeness.[14]

Theorem 13 *Let $\vec{u}(x,t)$ be the solution of (6.14) at time t. Let us assume that for $t \leq T$ this solution remains smooth. Let $I_0 = \vec{u}_0(\Omega)$, and $\mathcal{I}_0$ be the* convex hull *of I_0. Then for $(x,t) \in \Omega \times [0,T]$, $\vec{u}(x,t) \in \mathcal{I}_0$.*

6.3 Maps into Implicit Submanifolds

Here we present a modification to the diffusion flow introduced above, which is well suited to diffuse data that belongs to a certain *submanifold* $\mathcal{C}$ of $\mathcal{N} = \{\psi = 0\}$. We specify this submanifold by $\mathcal{C} = \{\psi = 0\} \cap \{\Phi = 0\}$, where we select $\Phi : I\!R^N \to R$ to be the signed *intrinsic* (to $\mathcal{N}$) distance function to $\{\Phi = 0\}$, satisfying (see Appendix B.3 for the notation)

$$1 = \|\nabla_\psi \Phi\| = \sqrt{\|\nabla \Phi\|^2 - |\nabla \psi \cdot \nabla \Phi|^2} \tag{6.22}$$

In addition we specify the condition

$$\Phi(p) = 0 \text{ for } p \in \mathcal{K}_\mathcal{C}$$

where

$$\mathcal{K}_\mathcal{C} = \{x \in I\!R^N \,|\, x = p + \alpha \nabla \psi(p), \text{with } p \in \mathcal{C},\, \alpha \in I\!R\}$$

is the *cone* intersecting $\{\psi = 0\}$ at $\mathcal{C}$ and director rays normal also to $\{\psi = 0\}$.

[13] Note once again that we are omitting details regarding the correct handling of the distance function, since it is not everywhere differentiable. However, by a regularization argument, the same conclusion holds.

[14] The proof of this result has a lot of interest in itself since it can be carried out within the implicit framework introduced in this chapter.

The reason for specifying the submanifold this way is that we cannot proceed as before, simply specifying the submanifold as the zero level set of it's Euclidean distance function. This is because such function would be singular precisely on the submanifold.

As we show in Appendix B.3, the Hessian of Φ, *intrinsic* to $\mathcal{N}$ evaluated at the point p, and restricted to act on vectors that belong to $T_p\mathcal{N}$, can be written in the form

$$\mathbf{H}^{\mathcal{N}}_{\Phi}(p) = \mathbf{H}_{\Phi}(p) - \Lambda(p)\ \mathbf{H}_{\psi}(p) \tag{6.23}$$

where $\Lambda(p) = \nabla\Phi(p)\cdot\nabla\psi(p)$. This expression will be used below.

We now derive the Euler-Lagrange corresponding to this additional mapping restriction. For this, we use a technique slightly different that the one in §6.2.1.

Proposition 8 *The Euler-Lagrange of the functional (6.1), when the solution is restricted to the implicitly represented submanifold $\mathcal{C}$ defined above, is given by*

$$\Delta\vec{u} + \left(\sum_k \mathbf{H}_{\psi}(\vec{u})\left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]\right)\nabla\psi(\vec{u}) + \left(\sum_k \mathbf{H}^{\mathcal{N}}_{\Phi}(\vec{u})\left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]\right)\nabla_{\psi}\Phi(\vec{u}) = 0. \tag{6.24}$$

Proof: Let us assume that $\vec{u}$ achieves a minimum of the energy functional (6.1). We must build a variation of $\vec{u}$ that belongs to $\mathcal{C}$, the intersection of the zero level-sets of two embedding functions (and not just of ψ as before). It is clear that one such variation would be

$$\vec{w}_{\lambda} = \mathbf{\Pi}_{\mathcal{C}}\left(\vec{u} + \lambda\vec{v}\right)$$

We are interested only on those terms of $e[\vec{w}_{\lambda}]$ that do not vanish after the $\sum_{i=1}^{N}\frac{\partial}{\partial x_i}(\bullet)\Big|_{\lambda=0}$ operation, namely those linear in λ. Therefore we only preserve those terms in $\vec{w}_{\lambda}$ which are constant or linear in λ:

$$\vec{w}_{\lambda} \simeq \vec{u} + \lambda\,\mathbf{\Pi}_{T_{\vec{u}}\mathcal{C}}(\vec{v})$$

We write

$$\begin{aligned}\mathbf{\Pi}_{T_{\vec{u}}\mathcal{C}}(\vec{v}) &= \mathbf{\Pi}_{T_{\vec{u}}\{\psi=0\}}\left\{\ \vec{v} - (\vec{v}\cdot\nabla_{\psi}\Phi(\vec{u}))\,\nabla_{\psi}\Phi(\vec{u})\ \right\}\\ &= \vec{v} - (\vec{v}\cdot\nabla_{\psi}\Phi(\vec{u}))\,\nabla_{\psi}\Phi(\vec{u}) - (\vec{v}\cdot\nabla\psi(\vec{u}))\,\nabla\psi(\vec{u})\end{aligned}$$

where $\nabla_{\psi}\Phi(\vec{u}) = \nabla\Phi(\vec{u}) - \Lambda(\vec{u})\ \nabla\psi(\vec{u})$ is the gradient of Φ *intrinsic* to $\{\psi = 0\}$.
In this way we find that (up to a first order in λ):

$$\begin{aligned}e[\vec{w}_{\lambda}] &\simeq e[\vec{u}] + \lambda\sum_{i=1}^{N}\vec{u}_{x_i}\cdot[\vec{v}_{x_i} - \vec{v}_{x_i}\cdot\nabla_{\psi}\Phi(\vec{u})\ \nabla_{\psi}\Phi(\vec{u}) \qquad (6.25)\\ &- \vec{v}\cdot\frac{\partial\nabla_{\psi}\Phi(\vec{u})}{\partial x_i}\ \nabla_{\psi}\Phi(\vec{u}) - \vec{v}\cdot\nabla_{\psi}\Phi(\vec{u})\ \frac{\partial\nabla_{\psi}\Phi(\vec{u})}{\partial x_i}\\ &- \vec{v}_{x_i}\cdot\nabla\psi(\vec{u})\ \nabla\psi(\vec{u}) - \vec{v}\cdot\frac{\partial\nabla\psi(\vec{u})}{\partial x_i}\nabla\psi(\vec{u}) - \vec{v}\cdot\nabla\psi(\vec{u})\frac{\partial\nabla\psi(\vec{u})}{\partial x_i}]\end{aligned}$$

Since $\Phi(\vec{u}) = \psi(\vec{u}) = 0$, differentiating with respect to x_i we obtain that $\nabla\Phi(\vec{u}) \cdot \vec{u}_{x_i} = \nabla\psi(\vec{u}) \cdot \vec{u}_{x_i} = 0$, and therefore

$$\nabla_\psi \Phi(\vec{u}) \cdot \vec{u}_{x_i} = 0$$

The expression (6.25) can be simplified to obtain

$$e[\vec{w}_\lambda] \simeq e[\vec{u}] + \lambda \, \sum_{i=1}^{N} \vec{u}_{x_i} \cdot \left[\vec{v}_{x_i} - \vec{v} \cdot \nabla_\psi \Phi(\vec{u}) \, \frac{\partial \nabla_\psi \Phi(\vec{u})}{\partial x_i} - \vec{v} \cdot \nabla\psi(\vec{u}) \, \frac{\partial \nabla\psi(\vec{u})}{\partial x_i} \right]$$

Moreover, since

$$\frac{\partial \nabla_\psi \Phi(\vec{u})}{\partial x_i} = \mathbf{H}_\Phi \vec{u}_{x_i} - \frac{\partial \Lambda}{\partial x_i}(\vec{u}) \; \nabla\psi(\vec{u}) - \Lambda(\vec{u}) \; \mathbf{H}_\psi \vec{u_{x_i}}$$

we have

$$\frac{\partial \nabla_\psi \Phi(\vec{u})}{\partial x_i} \cdot \vec{u}_{x_i} = \mathbf{H}_\Phi^{\mathcal{N}}[\vec{u}_{x_i}, \vec{u}_{x_i}]$$

With all this in mind we find that (again, up to first order in λ)

$$e[\vec{w}_\lambda] \simeq e[\vec{u}] + \lambda \, \sum_{i=1}^{N} \vec{u}_{x_i} \cdot \left[\vec{v}_{x_i} - \vec{v} \cdot \nabla_\psi \Phi(\vec{u}) \; \mathbf{H}_\Phi^{\mathcal{N}}[\vec{u}_{x_i}, \vec{u}_{x_i}] - \vec{v} \cdot \nabla\psi(\vec{u}) \; \mathbf{H}_\psi(\vec{u})[\vec{u}_{x_i}, \vec{u}_{x_i}] \right]$$

Using this expression, after imposing that $\frac{\partial}{\partial\lambda}\big|_{\lambda=0} \int_\Omega e[\vec{w}_\lambda] \; dv = 0$ for every $\vec{v}$, we find that the Euler-Lagrange is

$$\Delta\vec{u} + \left(\sum_k \mathbf{H}_\psi \left[\frac{\partial \vec{u}}{\partial x_k}, \frac{\partial \vec{u}}{\partial x_k} \right] \right) \; \nabla\psi(\vec{u}) + \left(\sum_k \mathbf{H}_\Phi^{\mathcal{N}} \left[\frac{\partial \vec{u}}{\partial x_k}, \frac{\partial \vec{u}}{\partial x_k} \right] \right) \; \nabla_\psi \Phi(\vec{u}) = 0. \tag{6.26}$$

an expression utterly predictable.

□

6.3.1 Simple Verification

As for the case of closed manifolds, we now verify that in fact the gradient descent corresponding to the Euler-Lagrange (6.26) keeps $\vec{u}$ in $\{\psi = 0\} \cap \{\Phi = 0\}$.

Proposition 9 *If $\vec{u}$ is a solution to the gradient descent flow corresponding to Equation (6.26), then $\vec{u}$ maps into the submanifold $\{\psi = 0\} \cap \{\Phi = 0\}$.*

Proof: We just need to show that both $\nu(x,t) \triangleq \psi(\vec{u}(x,t))$ and $\mu(x,t) \triangleq \Phi(\vec{u}(x,t))$ are always zero. The idea is the same one we used in §6.2.3, it is enough to show that both ν and μ satisfy the *heat equation* with adiabatic *boundary conditions.*

$[\psi]$

We have

$$\nu_t = \nabla\psi \cdot \Delta\vec{u} + \sum_k \mathbf{H}_\psi \left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]$$

since $\nabla\psi \perp \nabla_\psi\Phi$. Also

$$\Delta\nu = \nabla\psi \cdot \Delta\vec{u} + \sum_k \mathbf{H}_\psi \left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]$$

and

$$\nu_t = \Delta\nu$$

[$\mathbf{\Phi}$]

We have

$$\mu_t = \nabla\Phi \cdot \Delta\vec{u} + \nabla\Phi \cdot \nabla_\psi\Phi \left(\sum_k \mathbf{H}_\Phi^{\mathcal{N}} \left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]\right) + \Lambda \left(\sum_k \mathbf{H}_\psi \left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]\right)$$

From $\nabla\Phi \cdot \nabla_\psi\Phi = \nabla_\psi\Phi \cdot \nabla_\psi\Phi = \|\nabla_\psi\Phi\|^2 = 1$, the above equation continues as

$$\begin{aligned} &= \nabla\Phi \cdot \Delta\vec{u} + \left(\sum_k \mathbf{H}_\Phi^{\mathcal{N}} \left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]\right) + \Lambda \left(\sum_k \mathbf{H}_\psi \left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]\right) \\ &= \nabla\Phi \cdot \Delta\vec{u} + \left(\sum_k \mathbf{H}_\Phi \left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]\right) \end{aligned}$$

Also

$$\Delta\mu = \nabla\Phi \cdot \Delta\vec{u} + \left(\sum_k \mathbf{H}_\Phi \left[\frac{\partial\vec{u}}{\partial x_k}, \frac{\partial\vec{u}}{\partial x_k}\right]\right)$$

and then

$$\mu_t = \Delta\mu$$

Finally, it is easy to see that both ν and μ satisfy *Neumann boundary conditions*. Since at $t = 0$ both functions are zero, we must have that they are identically zero.

□

6.3.2 Example

We now present an example of the evolution corresponding to the above equation, where the target manifold the circle $S^1 \subset I\!R^3$. We will prove, by direct calculation, that the evolution PDE corresponding to (6.26) reduces to the expected one.

Let $\mathcal{C} = \{(x, y, z) \in I\!R^3 |\ x^2 + y^2 = 1,\ z = 0\}$. We will then choose the representation $\mathcal{C} = \{(x, y, z) \in I\!R^3 |\ z = 0\} \cap \{(x, y, z) |\ x^2 + y^2 = 1\}$. We select $\mathcal{N} = \{(x, y, z) \in I\!R^3 |\ z = 0\}$, that is, $\psi(x, y, z) = z$ and

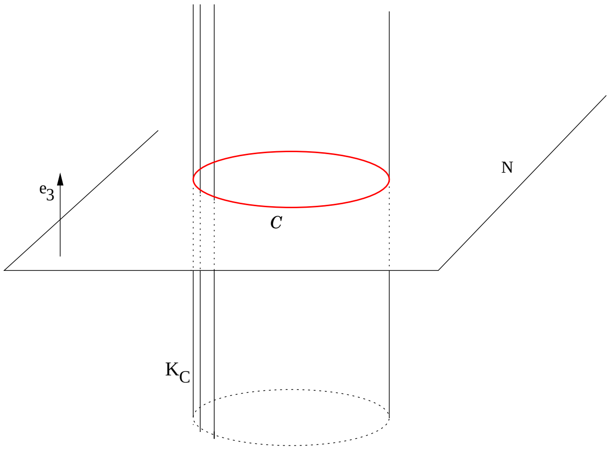


Figure 6.1: Example of a mapping into $S^1 \subset I\!R^3$.

therefore $\nabla\psi = \vec{e}_3 = (0,0,1)$. The set $\mathcal{K}_{\mathcal{C}}$ is then given by $\{(x,y,z)|\ x^2+y^2=1\}$. In Figure 6.1 we depict the situation. Now we solve (6.22) with the condition $\Phi\mid_{\mathcal{K}_{\mathcal{C}}} = 0$. Observe that $\nabla_\psi\Phi = (\Phi_x, \Phi_y, 0)$, then the PDE we must solve reads $\Phi_x^2 + \Phi_y^2 = 1$, and then the solution that satisfies the boundary condition above is $\Phi(x,y,z) = \sqrt{x^2+y^2} - 1$. Let $\rho = \sqrt{x^2+y^2}$. One computes that $\nabla\Phi(x,y,z) = (x,y,0)\rho^{-1}$. We can now find the components of $\mathbf{H}_\Phi$, the Hessian matrix of Φ at the point (x,y,z) to obtain $\Phi_{xx} = \rho^{-1} - x^2\rho^{-3}$, $\Phi_{yy} = \rho^{-1} - y^2\rho^{-3}$, $\Phi_{xy} = \Phi_{yx} = -xy\rho^{-3}$, and $\Phi_{zx} = \Phi_{xz} = \Phi_{zy} = \Phi_{yz} = \Phi_{zz} = 0$. Since $\mathbf{H}_\psi = 0$ we obtain $\mathbf{H}_\Phi^{\mathcal{N}}(x,y,z) = \mathbf{H}_\Phi(x,y,z)$.

The next step is to write equation (6.26) in this specific case. The first observation is that,again, since $\mathbf{H}_\psi = 0$, the time evolution corresponding to (6.26) simplifies to

$$\vec{u}_t = \Delta\vec{u} + \left(\sum_k \mathbf{H}_\Phi(\vec{u})\left[\vec{u}_{x_k}, \vec{u}_{x_k}\right]\right)\nabla\Phi(\vec{u})$$

For any vector $\vec{v} = (v_x, v_y, v_z) \in I\!R^3$, one has at the point (x,y,z), $\mathbf{H}_\Phi[\vec{v},\vec{v}] = v_x^2(1-x^2) + v_x^2(1-x^2) - 2xyv_xv_y = v_x^2 + v_y^2 - (xv_x + yv_y)^2$.

It is also of great help knowing, from §6.3.1, that along the time evolution, both $\psi(\vec{u}) = 0$ and $\Phi(\vec{u}) = 0$ if the initial datum is on $\mathcal{C}$. This translates into that $\rho = 1$ everywhere in our expressions for $\mathbf{H}_\Phi$ and $\nabla\Phi$. Let's write $\vec{u} = (U,V,W)$, then, since $U^2+V^2 = 1$, differentiating with respect to x_k we find $UU_{x_k} + VV_{x_k} = 0$. We also have $W = 0$.

Hence, $\mathbf{H}_\Phi(\vec{u})[\vec{u}_{x_k}, \vec{u}_{x_k}] = U_{x_k}^2 + V_{x_k}^2 - (UU_{x_k} + VV_{x_k})^2 = U_{x_k}^2 + V_{x_k}^2$, and the time evolution equation reads

$$\begin{cases} U_t = \Delta U + (\|\nabla U\|^2 + \|\nabla V\|^2)U \\ V_t = \Delta V + (\|\nabla U\|^2 + \|\nabla V\|^2)V \\ W = 0 \end{cases} \tag{6.27}$$

which we immediately recognize as the one corresponding to diffusion of maps into S^1 ($\subset I\!R^2$, if we discard the superfluous component W), see Appendix B.2, equation (B.2).

6.4 Implicit Domain Manifolds and p-Harmonic Maps

For completeness, we present now the formulas corresponding to the case where both the domain and target manifolds are represented in implicit form (with the embedding functions being the corresponding signed distance ones). Deriving these formulas is straightforward using the framework here presented when combined with the work in [134]. We also show the corresponding flows for p-harmonic maps.

6.4.1 p-Harmonic Maps

We still assume $\mathcal{M}$ to be planar. The energy density (6.2) (but not the dependence of the energy on its density) is redefined as follows. For every $p \in [1, +\infty)$ let

$$e_p[\vec{u}] \triangleq \frac{1}{p} \|\mathbf{J}_{\vec{u}}\|_{\mathcal{F}}^p$$

A simple application of variational calculus leads to conclude that[15]

$$\vec{u}_t = p^{1-\frac{2}{p}} \, \mathbf{\Pi}_{\nabla\psi(\vec{u})} \left(\nabla \cdot \left((e_p[\vec{u}])^{1-\frac{2}{p}} \, \mathbf{J}_{\vec{u}}^T \right) \right) \tag{6.28}$$

Note that if $p < 2$ difficulties are expected to arise, see [12] and the references therein.

6.4.2 Generic (Implicit) Domain Manifolds

Let $\mathcal{M} = \{x \in I\!R^m : \phi(x) = 0\}$, where $\phi(\cdot)$ is the signed distance function to $\mathcal{M}$, then the diffusion is given by:

$$\vec{u}_t = \nabla \cdot \left(\mathbf{\Pi}_{\nabla\phi} \mathbf{J}_{\vec{u}}^T \right) \; + \; \left(\sum_{k,r} \mathbf{H}_\psi[\vec{u}_{x_r}, \vec{u}_{x_k}] \, (\mathbf{\Pi}_{\nabla\phi})_{kr} \right) \nabla\psi \tag{6.29}$$

The whole deduction rests upon the redefinition of the energy (6.1) and its density (6.2). Now we should define the energy density to be

$$e_\phi[\vec{u}] \triangleq \frac{1}{2} \|\mathbf{J}_{\vec{u}}^\phi\|_{\mathcal{F}}^2$$

where the *intrinsic Jacobian* of $\vec{u}$ can be written as (see Appendix B.3 for more details) $\mathbf{J}_{\vec{u}}^\phi = \mathbf{J}_{\vec{u}} \, \mathbf{\Pi}_{\nabla\phi}$.

The new definition for the energy should be:[16]

$$E[\vec{u}] \triangleq \int_{I\!R^m} e_\phi[\vec{u}] \, \delta(\phi(x)) \, dx \tag{6.30}$$

Comparing (6.29) with (6.18), we can infer the implicit form of the Christoffel symbols:[17]

[15] The divergence operator convention (for a matrix A) we have used is $\nabla \cdot A = \left(\nabla \cdot \vec{A}_{v1} \middle| \dots \middle| \nabla \cdot \vec{A}_{vr} \right)$, where $\vec{A}_{vi}$ stands for the i-th column of A. That is, we apply a columnwise divergence.

[16] We have already taken into account that $\|\nabla\phi\| = 1$.

[17] Of course $g^{ij} = \left(\mathbf{\Pi}_{\nabla\phi}\right)_{ij}$ $(= g_{ij}^{-1})$. Then, it is nice to observe (although formally incorrect) that since $\mathbf{\Pi}_{\nabla\phi}\nabla\phi = 0$, then the metric $g : I\!R^d \to I\!R^{d\times d}$ has eigenvalue $+\infty$ in the direction given by $\nabla\phi$ thus prohibiting intermingling of information between adjacent level sets of ϕ.

$$\mathbf{\Gamma}^l_{ij}(\vec{u}) = \frac{\partial^2 \psi}{\partial u^i \partial u^j}(\vec{u})\ \frac{\partial \psi}{\partial u^l}(\vec{u})$$

6.4.3 Generic (Implicit) Domain Manifold and p-Harmonic Maps

Using both generalizations presented above, we arrive at the following formula with a bit more computational effort

$$\vec{u}_t = p^{1-\frac{2}{p}}\ \mathbf{\Pi}_{\nabla\psi(\vec{u})} \left(\nabla \cdot \left(\left(e_{\phi,p}[\vec{u}] \right)^{1-\frac{2}{p}}\ \mathbf{\Pi}_{\nabla\phi} \mathbf{J}^T_{\vec{u}} \right) \right) \tag{6.31}$$

where

$$e_{\phi,p}[\vec{u}] \stackrel{\triangle}{=} \frac{1}{p} \|\mathbf{J}^{\phi}_{\vec{u}}\|^p_{\mathcal{F}}$$

6.4.4 Diffusion of Tangent and Normal Directions

Throughout this section we will assume $dim(\mathcal{M}) = dim(\mathcal{N})$. Assume we want to diffuse intrinsic vectorial data constrained to be a direction (unit norm) and to be either normal or tangent to the domain manifold, e.g., [134]. This is an extremely important case, for example to denoise principal directions and normal vectors. We now derive these equations, which to the best of our knowledge have not been reported before even for explicit manifolds.

To achieve this goal, we minimize the functional (6.30) taking a variation of the form (assume $\vec{u}$ minimizes the energy functional while satisfying both $\|\vec{u}\| = 1$ and $\Pi(\vec{u}) = \vec{u}$)

$$\vec{u}_\lambda(x) \stackrel{\triangle}{=} \frac{\vec{u} + \lambda \Pi(\vec{v})}{\|\vec{u} + \lambda \Pi(\vec{v})\|}$$

where $\vec{v} : \mathcal{M} \to I\!R^d$ is smooth and Π is either $\mathbf{\Pi}_{T_x\mathcal{M}}$ or $\mathbf{\Pi}_{N_x\mathcal{M}}$ (projection onto the tangent or normal space respectively). Let $\vec{w} = \Pi(\vec{v})$, then it follows easily that

$$\left.\frac{dE[\vec{u}_\lambda]}{dt}\right|_{\lambda=0} = -\int_{I\!R^m} \{\Delta_\phi \vec{u}\ +\ 2\, e_\phi[\vec{u}]\ \vec{u}\} \cdot \vec{w}\ \ \delta(\phi(x))\, dx$$

Imposing $\left.\frac{dE[\vec{u}_\lambda]}{dt}\right|_{\lambda=0} = 0$ for all v implies

$$\Pi\left(\Delta_\phi \vec{u}\ +\ 2\, e_\phi[\vec{u}]\ \vec{u}\right) = \Pi\left(\Delta_\phi \vec{u}\right)\ +\ 2\, e_\phi[\vec{u}]\ \vec{u} = 0$$

Finally, the diffusion flow obtained is

$$\frac{\partial \vec{u}}{\partial t}(x,t) = \Pi_x\left(\Delta_\phi \vec{u}(x,t)\right)\ +\ 2\, e_\phi[\vec{u}](x,t)\ \vec{u}(x,t) \tag{6.32}$$

Note that if the PDE (6.32) admits a smooth solution until time T,[18] and if (for instance) we are dealing with tangent directions diffusion, the function $f(x,t) \stackrel{\triangle}{=} \nabla\phi(x) \cdot \vec{u}(x,t)$ satisfies $f_t(x,t) = 2e_\phi[\vec{u}]f(x,t)$.

[18] Note that we might be subject to the topological obstruction given by the Hairy Ball Theorem when $dim(\mathcal{M}) = dim(\mathcal{N})$ is odd.

Therefore

$$f(x,t) = f(x,0)e^{2\int_0^t e[\vec{u}](x,t)\,dt}$$

thus verifying that if $\nabla\phi(x)\cdot\vec{u}(x,0) = 0$ then $\nabla\phi(x)\cdot\vec{u}(x,t) = 0$ for $t \leq T$. We also want to check whether $\|\vec{u}(x,t)\| = 1 \;\forall\; (x,t)$. Let

$$F_\phi[\vec{u}](t) \stackrel{\triangle}{=} \frac{1}{2}\int_{I\!\!R^d} \|\mathbf{J}_{\vec{u}}\|^2_{\mathcal{F}}\;\delta(\phi(x))\;dx$$

then $\dot{F}_\phi[\vec{u}](t) = -\int_{I\!\!R^d} \vec{u}_t\cdot\Delta_\phi\vec{u}\;\;\delta(\phi(x))\,dx$. Since both $\|\vec{u}\| = 1$ and $\Pi(\vec{u}) = \vec{u}$ (so $\Pi(\vec{u}_t) = \vec{u}_t$ since Π does not depend on t) must hold, and in order to make $\dot{F}_\phi[\vec{u}](t)$ non-positive we choose

$$\vec{u}_t = \Pi\,\mathbf{\Pi}_{T_{\vec{u}}\{\|\vec{u}\|=c\}}\Delta_\phi\vec{u} \tag{6.33}$$

where $\mathbf{\Pi}_{T_{\vec{u}}\{\|\vec{u}\|=c\}} = \mathbf{I} - \frac{\vec{u}}{\|\vec{u}\|}\frac{\vec{u}^T}{\|\vec{u}\|}$ for any $c > 0$.

Note that the above evolution indeed forces $\vec{u}(x,t)$ to satisfy both imposed conditions. Let $\vec{v} : I\!\!R^d \to I\!\!R^d$ be such that $\Pi(\vec{v}) = \vec{0}$ then $(\vec{v}\cdot\vec{u})_t = \vec{v}\cdot\vec{u}_t = \vec{v}\cdot\Pi\,\mathbf{\Pi}_{T_{\vec{u}}S^{d-1}}\Delta_\phi\vec{u} = \Pi^T\vec{v}\cdot\mathbf{\Pi}_{T_{\vec{u}}\{\|\vec{u}\|=c\}}\Delta_\phi\vec{u} = 0$, since the projection matrix is symmetric, and just using this we have $\left(\frac{1}{2}\|\vec{u}\|^2\right)_t = \vec{u}\cdot\vec{u}_t = \vec{u}\cdot\mathbf{\Pi}_{T_{\vec{u}}\{\|\vec{u}\|=c\}} = 0$ trivially. Finally, using $\|\vec{u}\| = 1$ and carrying out some computations in a way similar to §6.5 below,[19] one can prove that (6.33) reduces to (6.32).

6.5 Numerical Implementation and Examples

We now discuss the numerical implementation of the flows previously introduced. Since the target manifold is now implicitly represented, we can basically use classical, well studied, numerical techniques on Cartesian grids. In other words, the framework here introduced permits the use of already existing numerical techniques, thereby enjoying their available analysis results. This is a key concept, instead of working on the development of new numerical schemes for meshes, the use of implicit representations following our framework brings us back to classical schemes. Moreover, examples like those in Figure 6.5 have not been reported in the literature yet, since prior to our approach all PDEs for mapping 3D meshes used projections as intermediate steps. Therefore, the work here proposed, when combined with [134], not only permits to use classical numerical schemes to solve PDEs and variational problems for surfaces, it is also an enabling technology for general maps.

Note that although the flows derived in this chapter guarantee that the map remains on the target (submanifold), numerical errors can move it away from it, requiring a simple projection step (see the projection equations presented before in this chapter). In particular, when dealing with submanifolds, although the evolution equations also guarantee that the solution will remain inside the convex hull, due to numerical discretization, $\vec{u}$ could be taken outside of it during the evolution. In order to numerically project it back, we need to have a distance function to this convex hull defined on the implicitly defined target manifold. In [145] we have shown how to computationally optimal compute such a distance function on implicitly defined manifolds, and this is the technique used for this projection into the convex hull.

[19] The main difference is that now one must take into account the Laplace-Beltrami expressed "implicitly," see Appendix B.3 for more details on intrinsic differential operators within the implicit framework.

An explicit scheme can be devised to implement (6.29) (recall that this is the extension, for general domain manifolds, of the Equation (6.14) derived in §6.2). However, following [62], it turns out that it is more convenient to implement the *mathematically equivalent* evolution derived in §6.2.4. More specifically, the *equivalent* evolution is (see equation (6.21))

$$\frac{\partial u}{\partial t} = \Delta u \,-\, \big(\Delta u \cdot \nabla \psi\big) \nabla \psi \tag{6.34}$$

That both evolutions are equivalent is easy to see:

Proposition 10 *Equation (6.34) is equivalent to the mapping into implicit surfaces flow (6.14).*

Proof: One has that $f(x,t) \overset{\triangle}{=} \psi(\vec{u}(x,t)) = 0 \; \forall (x,t) \in \Omega \times I\!R^+ \cup \{0\}$ for $\vec{u}(\cdot,\cdot)$ satisfying (6.14). Now, differentiating f with respect to x_i we obtain

$$\nabla\psi(\vec{u}) \cdot \vec{u}_{x_i} = 0$$

Differentiating again with respect to x_i,

$$\mathbf{H}_\psi[\vec{u}_{x_i}, \vec{u}_{x_i}] \,+\, \nabla\psi \cdot \vec{u}_{x_i x_i} = 0$$

Summing for all i,

$$\sum_i \mathbf{H}_\psi[\vec{u}_{x_i}, \vec{u}_{x_i}] \,+\, \nabla\psi \cdot \Delta\vec{u} = 0$$

and using the previous expression we derive (6.34) from (6.14).
□

6.5.1 Numerical Scheme

All the coding was done using **Flujos** as the main core (see [74]) and **VTK** (see [202]) for visualization purposes. Note that for visualization purposes only, the surfaces are triangulated at the end, via marching cubes as implemented in [202]. This is not at all an intrinsic component of our framework, and many applications (e.g., brain warping and regularization problems) are interested in the values of the solution $\vec{u}$, without the need for visualization of the target surface.

All the examples below were carried based in equation (6.31). Once again, the numerical implementation is straightforward (at least when $p = 2$), since it is basic Cartesian numerics, and full details and analysis can be found in the standard literature in numerical analysis. We select a particular efficient scheme from the literature, while others (including implicit or semi-implicit schemes) could be used as well.

We use forward time discretization (explicit scheme), and for the spatial discretization, we used the following well known recipe. To spatially discretize

$$f_t(x,t) = \nabla \cdot (\mathbf{K}(x) \nabla f(x,t)) \tag{6.35}$$

($\mathbf{K}(x)$ is a symmetric positive semi-definite matrix), we consider *backward* approximation of the divergence and a *forward* approximation of the gradient. Let's explain how this applies in our situation, and for that we assume $p = 2$ in (6.31). Then the equation we have to implement is

$$\vec{u}_t(x,t) = \mathbf{\Pi}_{\nabla\psi(\vec{u}(x,t))}\left(\nabla \cdot \left(\mathbf{\Pi}_{\nabla\phi(x)}\left(\mathbf{J}^T_{\vec{u}}(x,t)\right)\right)\right) \tag{6.36}$$

If we don't take into account the outer projection matrix, every coordinate of $\vec{u}$ evolves according to

$$u^i_t(x,t) = \nabla \cdot \left(\mathbf{\Pi}_{\nabla\phi(x)}\nabla u^i(x,t)\right)$$

having for each component the same structure than the model evolution (6.35). We then borrow the above discretization for our evolution. If we consider the coupling among different u^is imposed by the projection matrix $\mathbf{\Pi}_{\nabla\psi(\cdot)}$, we see that we still preserve numerical stability since this matrix is positive semidefinite and has spectral radius not greater than 1.[20] In more detail, it can be shown after some calculations (see [91, 191]) that for the scheme (p now denotes a position over the grid)

$$\vec{v}^{n+1}_p = \vec{v}^n_p + \Delta t\ \mathbf{P}(\vec{v}^n_p)\left(\nabla^- \cdot \left(\mathbf{Q}(p)\nabla^+\vec{v}^n_p\right)\right)$$

the stability condition is of the form ($\lambda = \frac{\Delta t}{(\Delta x)^2}$)

$$\lambda \leq \min_{p,u} \frac{S(p)}{\rho(\mathbf{P}(u)) \max\{S^2(p), D^2(p)\}}$$

or

$$\lambda \leq \frac{1}{\max_u \rho(\mathbf{P}(u))}\ \min_p \left\{\frac{S(p)}{\max\{S^2(p), D^2(p)\}}\right\}$$

where $\rho(\mathbf{P}(p))$ stands for the spectral radius of the matrix $\mathbf{P}(p)$, $S(p) = \sum_{ij}\left(q_{ij}(p) + q_{ij}(p - \Delta x\vec{e}_i)\right)$, and $D(p) = \sum_{ij}\left(q_{ij}(p) - q_{ij}(p - \Delta x\vec{e}_i)\right)$. In our case we may admit $D(\cdot)$ to be small compared with $S(\cdot)$ (given the identification $\mathbf{Q} \leftrightarrow \mathbf{\Pi}_{\nabla\phi}$) when Δx is small. This can be easily related to the curvatures of $\{\phi = 0\}$ giving a condition on the sampling of the distance function (ϕ) representing the domain manifold. This condition mainly means that we require a fine enough sampling as to guarantee that the change in the normals to the level surfaces of ϕ is small between adjacent grid points. This condition is obviated when the domain manifold is planar. So the stability condition becomes

$$\lambda \leq \frac{1}{\max_u \rho(\mathbf{P}(u))\ \max_p S(p)}$$

Since by Cauchy-Schwartz's inequality (and the aforementioned assumption on the change of $\nabla\phi$ between adjacent grid points) $2\,d$ (in practise) upper-bounds $S(p)$, remembering the fact that $\rho(\mathbf{P}(p)) \leq 1$, we arrive at $\lambda \leq \frac{1}{2\,d}$. Note that if a more careful implementation is desired, good choices are ADI or AOS schemes, see [206].

All derivatives in $\mathbf{\Pi}_{\nabla\psi(\cdot)}$ and $\mathbf{\Pi}_{\nabla\phi(\cdot)}$ were approximated by central differences. An interpolation scheme had to be used since the evaluations of $\mathbf{\Pi}_{\nabla\psi(\cdot)}$ in the above equation are at positions given by $\vec{u}(x,t)$, positions not necessarily on the underlying grid. We used linear interpolation for this purpose.

Note that as done in [134], when the domain manifold is also implicitly represented, the values of the map on it are, from time to time (every 5 iterations, for example), extended to its surrounding offset due

[20] Note that $\|\vec{v}\|^2 \geq \mathbf{\Pi}_{\nabla\psi}[\vec{v},\vec{v}] = \|v\|^2 - |\nabla\psi \cdot \vec{v}|^2 \geq 0$ for all $\vec{v}$. We have used that ϕ is a distance function.

to stability considerations, we call this process "extension evolution". This process is well known in the area of Implicit Surfaces. Also, as explained before, due to numerical discretization, the discretely computed solution map can be taken out of the target manifold during the evolution. In this chapter, we simply project it back at every iteration. We have seen that this projection is a trivial step due to the fact that the embedding is a distance function. It is quite straightforward to show that the results reported in [4] can be extended for our equations as well, at least for convex hyper-surfaces (additional numerical work in this area has been performed by Prof. W. E, [62]). This guarantees then that the projection step does not introduce numerical problems. Further analysis of this projection step will be reported elsewhere.

This provides the whole numerical scheme for this particular equation using our framework. To resume, we implement (6.36) with simple finite differences schemes (central, forward, and backward differences). At every numerical iteration, the values of $\vec{u}$ are projected to the zero level-set to correct for possible numerical errors (projection which becomes trivial since the embedding function is a distance function). If the domain manifold is not planar, every k ($k = 5$ in our experiments) iterations we run a certain number of iterations of the extension evolution, [134]. When needed, we interpolate the values of the grid onto the underlying surface by simple linear interpolation. All these steps are widely known, simple to implement, are based on well known numerical schemes, and are generic and not designed just for a particular flow.

6.5.2 Examples

In all the examples below, the domain manifold $\mathcal{M}$ is either the Euclidean space $I\!R^2$ or an implicit torus. The target manifold $\mathcal{N}$ is an implicit surface in $I\!R^3$, that is, the zero level-set of $\psi : I\!R^3 \to I\!R$, ψ being a signed distance function (this is of course also the case when the surface is a sphere, ψ being as in §6.2.3).

In order to present interesting examples we construct texture maps, add noise to them, and then diffuse them using our framework. Let $\mathcal{S}$ be the surface onto which we want to map a given (planar) image defined in a subset $D \subset I\!R^2$. Then the *texture map* is a map $\vec{T} : \mathcal{S} \to D$. Once the map is known, we inverted it to find a map $\vec{u}_0 : D \to \mathcal{S}$. Then, we built up the noisy map $\vec{u} : D \to \mathcal{S}$ defined by

$$\vec{u}(x) = \mathbf{\Pi}_{\mathcal{S}} \left(\vec{u}_0(x) + \vec{\mathbf{n}}(x)\right)$$

where $\vec{\mathbf{n}} : D \to \mathcal{S}$ is random map with small prescribed power σ. We then feed the evolution (6.14) with $\vec{u}$ as initial condition, and Neumann boundary conditions. After a certain number of steps, we stop the evolution, invert the resulting map, and use it as a texture map to paint the surface with a certain texture.[21]

As a means of finding a suitable $\vec{T}$ we have implemented the work in [217] (a multidimensional scaling approach), combined with the technique developed in [145] for computing distances on implicit surfaces. In all the steps just described there are some minor implementation details, mainly regarding interpolation tasks, that we omit for the sake of clarity.

In Figures 6.2, 6.3 and 6.4 we then denoise vectors from the plane $I\!R^2$ to a 3D surface defined as the zero level-set of $\psi : I\!R^3 \to I\!R$ and map a texture image to the surface using the obtained map. Note that the map is the one being processed, not the image itself.

We also show an example of diffusion of random maps from an implicit torus to the implicit bunny model, see Figure 6.5. As expected from the theory, when evolving this set with the harmonic flow, the set

[21] Note that we are not proposing this as a complete texture mapping alternative, it is just to provide an illustrative example.

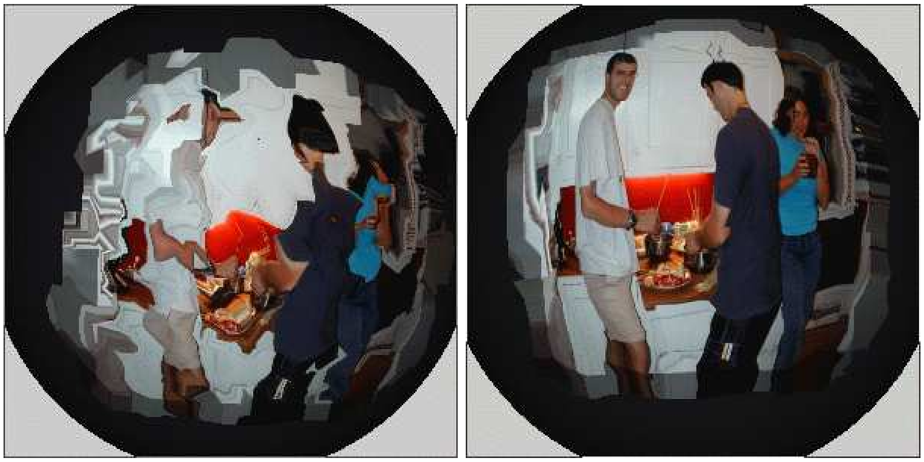


Figure 6.2: Diffusion of a noisy texture map (left) onto an implicit sphere (right).

converges to a unique point. This particular example of mapping a given 3D surface to another one was previously addressed via artificial, distortion introducing, projections to the plane or sphere when the surface was represented as meshes [195].

6.6 Conclusions

In this chapter we have shown how to implement variational problems and partial differential equations onto general target surfaces. We have also addressed the case of open target surfaces and sub-manifolds. The key concept is to represent the target (sub-)manifolds in implicit form, and then implement the equations in the corresponding embedding space. This framework completes the work with general domain manifolds reported in [134], thereby providing a complete solution to the computation of maps between generic manifolds.

We are currently using this framework to map two generic surfaces for warping (without intermediate projections onto the plane), and to develop numerical techniques for high order flows on and onto surfaces. To complete the general computational framework here introduced, a detailed numerical analysis on comparison with mesh beased techniques is to be performed. For the work on implicit domain manifolds introduced in [134], some of this analysis was recently performed in [1]. We plan to perform similar tests for implicit target manifolds and results will be reported elsewhere.

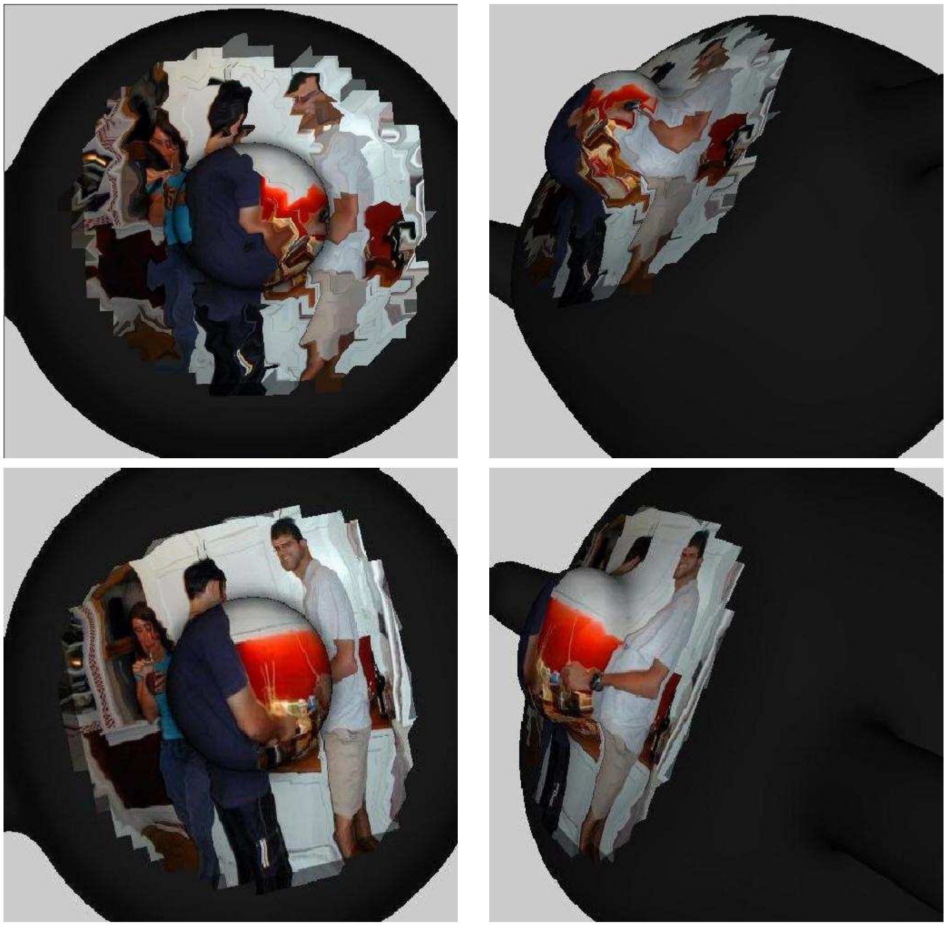


Figure 6.3: Diffusion of a noisy texture map onto an implicit teapot. We show two different views (noisy on the top and regularized on the bottom).

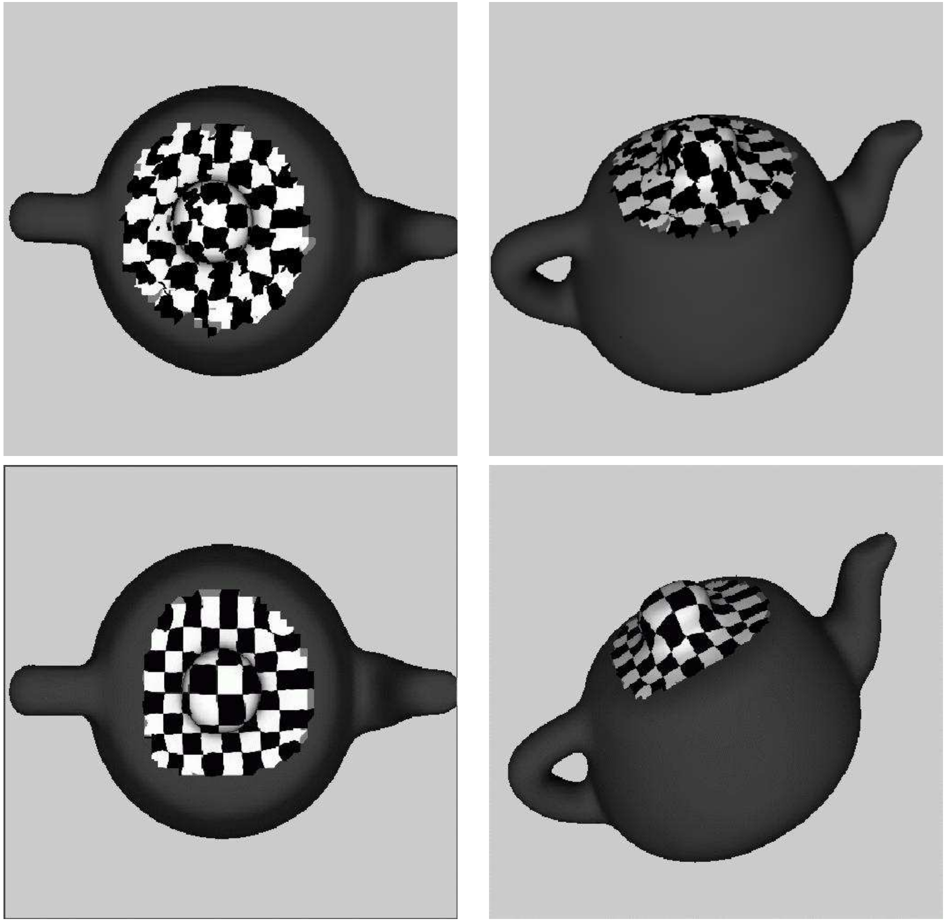


Figure 6.4: Diffusion of a texture map for an implicit teapot (noisy on he top and regularized on the bottom). A chess board texture is mapped.

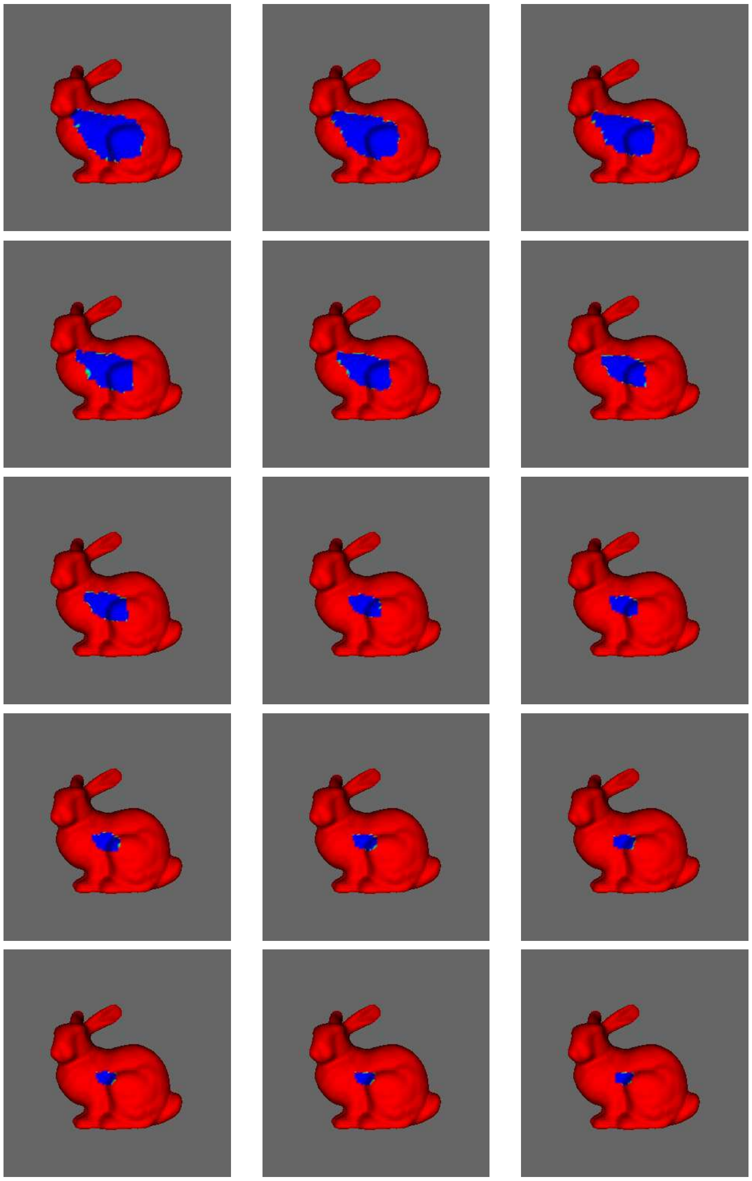


Figure 6.5: Diffusion of a random map from an implicit torus to the implicit bunny. In blue are marked those points of the bunny's surface pointed by the map at every instant. Different figures correspond to increasing instances of the evolution, from top to bottom and left to tight. We show the map at 17 of 100 iterations performed to the initial map with a time step of .01. We used the 2-harmonic heat flow with adiabatic conditions.

Chapter 7

Two applications to Brain Imaging

7.1 Introduction

Some of the most fundamental brain imaging analysis processes can be seen as finding appropriate maps from a general manifold $\mathcal{M}$ (*domain manifold*) onto another general manifold $\mathcal{N}$ (*target manifold*). For example, to compare brain data across subjects, individual brain datasets ($\mathcal{M}$) are often mapped to a neuroanatomical template or brain atlas ($\mathcal{N}$), using a spatial transformation that deforms brain surfaces or volumes to match their counterparts in the atlas, e.g., [11, 48, 195]. Additional examples are finding special curves such as sulcal beds ($\mathcal{M}$ being a segment of the real line and $\mathcal{N}$ the 3D brain surface, i.e., the cortex), regularizing or smoothing maps of surface-based signals, such as fMRI data or cortical thickness ($\mathcal{M}$ being the 3D brain surface and $\mathcal{N}$ the positive real line), and brain warping (both $\mathcal{M}$ and $\mathcal{N}$ are 3D brains), e.g., [42, 146, 195]. In other words, it is fundamental for brain imaging research to efficiently compute maps between manifolds, from a generic $\mathcal{M}$ to a generic $\mathcal{N}$. Moreover, all these important tasks can be addressed with partial differential equations (PDEs) or variational formulations between the two manifolds. PDEs have been widely used in neuroimaging, for example in segmenting anatomy using deformable surfaces and probability diffusion [33, 55, 116, 190], in denoising or enhancing brain-derived signals using anisotropic diffusion or scale-spaces [212], and in computing structural brain changes in development or dementia, e.g., [78, 77, 192] and references therein. In this chapter we show a couple of applications of the ideas explained before in this thesis to problems in Brain Imaging. In §7.2 we follow [145] and show how to compute interesting geodesics on implicit cortical surfaces, namely sulcal fundi in the cortex. Based on Chapter 6, in §7.3 we show how to find smooth curves constrained by landmarks on a cortical surface.

7.2 Geodesic computations

Computing distance functions and geodesics on surfaces has a number of applications in brain imaging. For example, it can be used for finding cortical features such as sulci, for cortical surface flattening, for visualization, and for brain warping [115, 195, 200, 203], for correctly estimating spatial correlations in cortical fMRI signals [211], and for estimating the variability and distance between functional loci in the cortex. In Figure 7.1 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 [15]

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

where $\mathbf{M}$ stands for the mean curvature of the level sets of ψ, so it is computed simply as $\mathbf{M}(x) = \Delta\psi(x)$. We then used the procedure described in §2.4 for tracking intrinsic geodesics. In the example here 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 [15] and in the references therein.

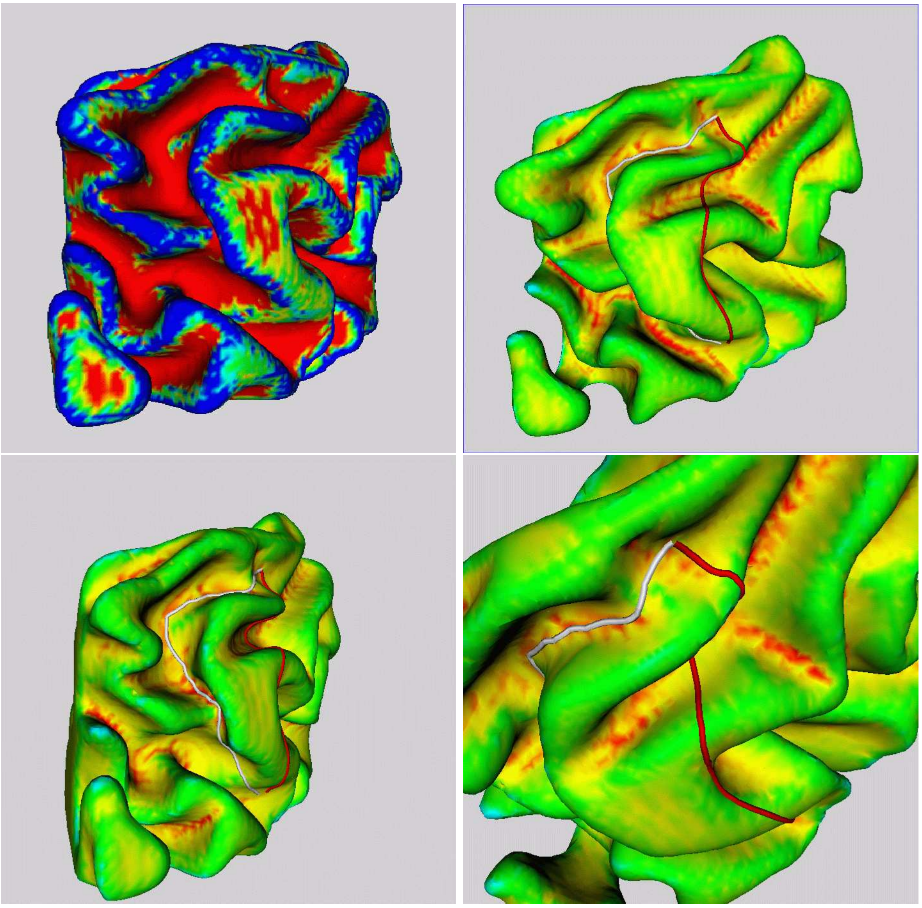


Figure 7.1: These four figures show the detection of valleys over implicit surfaces representing a portion of the human cortex. We use a mean curvature based weighted distance. In the left-upper corner we show the mean curvature of the brain surface (clipped to improve visualization). It is quite convincing 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.

7.3 Curve Smoothing via Heat Flow

We now present two examples of the formulation in Chapter 6 for smoothing curves on an implicit cortical surface $\mathcal{M}$ using Cartesian numerics. Our goal is to compute a smooth curve that passes through a number of previously marked points (landmarks). Without the constraints given by the points, the solution will be a geodesic. The curve is forced to go through the points (or close to them) simply by adding a term that penalizes its distance to the landmarks $\{p_1, \dots, p_N\} \subset \mathcal{M}$.

We consider the functional

$$\mathbf{J}(\gamma) := \mathbf{J}_s(\gamma) + \lambda \mathbf{J}_l(\gamma)$$

where $\mathbf{J}_s(\gamma) := \frac{1}{2}\int_o^1 \|\gamma_x\|^2 dx$ and $\mathbf{J}_l(\gamma) = \int_0^1 g_l(\gamma(x))\, dx$ and $g_l(y) := \sum_{i=1}^N \rho\left(d_{\mathcal{M}}(p_i, y)\right)$ for some convenient ρ. In our experiments $\rho(z) = 1 - e^{-\frac{z^2}{\sigma}}$, where σ measures the *area of atraction* of each landmark point.

The idea is that we are willing to exchange smoothness, as measured by $\mathbf{J}_s$ for proximity to the landmarks.

Let $\gamma : [0,1] \times (0,T) \to \mathcal{M} = \{x : \psi(x) = 0\}$ be a family of curves.

The resulting gradient descent flow for the functional $\mathbf{J}$ is

$$\gamma_t = \gamma_{xx} + \mathbf{H}_\psi(\gamma)[\gamma_x, \gamma_x]\nabla\psi(\gamma) - \lambda \vec{V_l}(\gamma)$$

where, for $m \in \mathcal{M}$, $\vec{V}(m)$ corresponds to the Gateâux derivative of $\mathbf{J}_l$.

This is solved with time-implicit numerics to speed up convergence. Figure 7.3, first row, presents a first example, where we clearly see the evolution of the yellow curve to the constrained black one. On the second and third rows we extend this for the computation of the sulcal bed, constrained by a few given points (regular view on the left and zoomed in one on the right). For this, we add a third term that penalizes paths that move away from the sulcal bed defined as follows

$$\mathbf{J}_{sb}(\gamma) := \int_0^1 g_{valley}(\gamma(x))\, dx$$

where g_{valley} is the weight defined in §7.2.

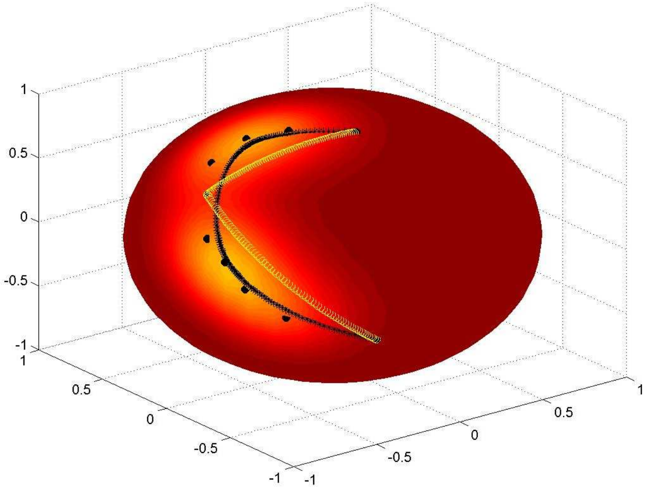


Figure 7.2: In this example, the yellow curve is the initial one and the black one is the final one. Also in black are shown the landmark points for these example. The color on the surface of the sphere at the point y is given by $g_l(y)$.

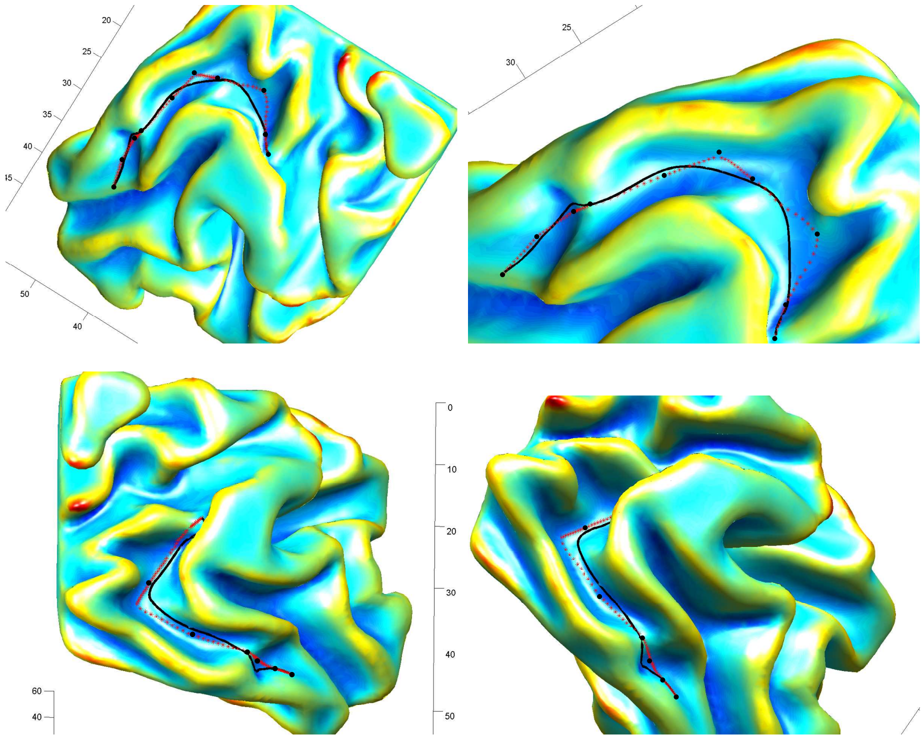


Figure 7.3: Finding special curves on implicit target surfaces. From an initial curve, adding constraints to the classical harmonic energy, we obtain curves that are attracted to marked points (landmarks). Here we show the procedure applied to a curve on a piece of cortical surface.

Chapter 8

Minimizing Lipschitz extensions for Surface Warping

8.1 Introduction

Brain warping, a form of brain image registration and geometric pattern matching, is one of the most fundamental and thereby most studied problems in computational brain imaging [195]. Brain images are commonly warped, using 3-dimensional deformation fields, onto a common neuroanatomic template prior to cross-subject comparison and integration of functional and anatomical data. Images of the same subject may be warped into correspondence over time, to help analyze shape changes during development or degenerative diseases. Almost all the active research groups in this area have developed and/or have their favorite brain warping technique.[1] A few representative works can be found at [42, 44, 48, 71, 78, 90, 146, 192, 194, 195, 67, 200], this list being far from complete. In spite of this, the problem is still open and widely studied, since there is not a "ground truth" method to obtain a map between brains. The criteria for matching different features (e.g., geometry or intensity) may also depend on the applications, which range from recovering intraoperative brain change to mapping brain growth, or reducing cross-subject anatomical differences in group functional MRI studies.

The way the brain warping problem is addressed is critical for studies of brain diseases that are based on population comparisons. Examples of this application can be found at [78, 192], although these are a very non-exhaustive account of the rich literature on the subject. The interested reader may also check [193] for numerous applications of brain warping and population studies. As detailed in [195], brain warping approaches can be divided into two classes, those based on volume-to-volume matching and those based on surface-to-surface matching. Our work belongs to the latter of the categories. Surface matching has recently received increasing attention as most functional brain imaging studies focus on the cortex, which varies widely in geometry across subjects. The power of these studies depends on the degree to which the functional anatomy of the cortex can be aligned across subjects, so improved cortical surface registration has become a major goal. In contrast with flow based works such as those in [42, 146, 192], our motivation is as

[1] This includes groups at JHU, UCLA, U. Penn., INRIA, MGH, GATECH, Harvard-BW, and the University of Florida, to name just a few.

in [8, 90, 94, 102, 108, 216, 217, 204]. That is, we aim to compute a map that preserves certain pre-defined geometric characteristics of the surfaces. While the literature has mainly attempted to preserve angles and areas, we work with geodesic distances (see also [177]). Our work is inspired by the literature on Lipschitz minimizing maps and in its connection to the infinite Laplacian. The motivation for using these frameworks will be presented after some brief mathematical introduction below.

In this chapter we therefore introduce the use of Lipschitz minimizing maps into the area of computational brain imaging, presenting a theoretical and computational framework complemented by examples with artificial and real data. An additional critical contribution of the work here presented is that intermediate distorting maps to the plane or sphere are avoided – these intermediate mappings are common practice in the brain warping literature.

8.2 Formal statement of the problem

Let $\mathcal{B}_1$ and $\mathcal{B}_2$ be two cortical surfaces (2D surfaces in the three dimensional Euclidean space) which we consider smooth and endowed with the metric inherited from $I\!R^3$ so that $d_{\mathcal{B}_1}$ and $d_{\mathcal{B}_2}$ are the geodesic distances measured on $\mathcal{B}_1$ and $\mathcal{B}_2$, respectively. Let $\Gamma_1 \subset \mathcal{B}_1$ and $\Gamma_2 \subset \mathcal{B}_2$ be subsets which represent features for which a correspondence is already known. In general, the sets Γ_i are the union of smooth curves traced on the surfaces, e.g., sulcal beds lying between gyri, and/or a union of isolated points. A set of anatomical landmarks that occur consistently in all subjects can be reliably identified using standardized anatomical protocols or automated sulcal labeling techniques (see for example Brain VISA by Mangin and Riviere and SEAL by Le Goualher).

Functional anatomy also varies with respect to sulcal landmarks, but sulci typically lie at the interfaces of functionally different cortical regions so aligning them improves the registration of functionally homologous areas. As commonly done in brain warping [195], we assume that a correspondence between Γ_1 and Γ_2 is pre-specified to the map (boundary conditions of the map). In this correspondence, internal point correspondences may be allowed to relax along landmark curves in the final mappings, e.g., [132].

To fix ideas let's assume that $\Gamma_1 = \cup_{k=1}^N x_i$ and $\Gamma_2 = \cup_{k=1}^N y_i$, and that the correspondence is given by $x_i \mapsto y_i$ for $1 \le i \le N$.

We want to find a (at least continuous) map $\phi : \mathcal{B}_1 \to \mathcal{B}_2$ such that $\phi(x_i) = y_i$ for $1 \le i \le N$ and such that ϕ produces minimal distortion according to some functional $\mathbf{J}$. One possible way of interpreting this problem is that we are trying to extrapolate or extend the correspondence from Γ_1 to the whole of $\mathcal{B}_1$ in such a way that we achieve small distortion.

A possible way to measure the distortion produced by a map ϕ is by computing the functionals ($1 \le p < \infty$)

$$\mathbf{J}_p(\phi) = \left(\frac{1}{\mu(\mathcal{B}_1)} \int_{\mathcal{B}_1} \|D_{\mathcal{B}_1}\phi\|_2^p \, \mu(dx) \right)^{1/p} \tag{8.1}$$

where $D_{\mathcal{B}_1}$ denotes differentiation intrinsic to the surface $\mathcal{B}_1$ and μ is the area measure on $\mathcal{B}_1$. One immediate idea is then to consider, for a fixed $p \in (1, \infty)$, the following variational problem:

Problem 1 (minimize $\mathbf{J}_p$) *Find $\phi \in \mathcal{S}$ such that $\mathbf{J}_p(\phi) = \inf_{\psi \in \mathcal{S}} \mathbf{J}_p(\psi)$, where $\mathcal{S}$ is a certain smoothness class of maps ϕ from $\mathcal{B}_1$ to $\mathcal{B}_2$ such that they respect the given boundary conditions $\phi(x_i) = y_i$ for all $x_i \in \Gamma_1$.*

The case $p = 2$ corresponds to the Dirichlet functional and has connections with the theory of (standard) Harmonic Maps. In more generality, it is customary to call the solutions to Problem 1 p-Harmonic Maps, see for example [64, 107, 68]. It is easy to show, under mild regularity assumptions, that for a fixed ϕ, $\mathbf{J}_p(\phi)$ is nondecreasing as a function of p, and that [82]

$$\mathbf{J}_\infty(\phi) := \lim_{p\uparrow\infty} \mathbf{J}_p(\phi) = \text{essup}_{x\in\mathcal{B}_1}\|D_{\mathcal{B}_1}\phi(x)\|_2, \tag{8.2}$$

which is the Lipschitz constant of ϕ.

In this chapter we propose to use the functional $\mathbf{J}_\infty$ as a measure of distortion for maps between cortical surfaces and to solve the associated variational problem in order to find a candidate mapping between the cortical surfaces (constrained by the provided boundary conditions).

Let $\mathcal{L}$ denote the space of all Lipschitz continuous maps $\psi : \mathcal{B}_1 \to \mathcal{B}_2$ such that $\psi(x_i) = y_i$ for $1 \leq i \leq N$. We then propose to solve the following problem:

Problem 2 (minimize $\mathbf{J}_\infty$) *Find $\phi \in \mathcal{L}$ such that $\mathbf{J}_\infty(\phi) = \inf_{\psi\in\mathcal{L}} \mathbf{J}_\infty(\psi)$.*

We now argue in favor of this functional.

8.2.1 Why use $\mathbf{J}_\infty$?

Our first argument is that $\mathbf{J}_\infty$ measures distortion in a more global way than any of the $\mathbf{J}_p$ for $p \in (1,\infty)$, since instead of computing an averaged integral quantity, we are looking at the supremum of the local distortions, $\|D_{\mathcal{B}_1}\phi(x)\|_2$. Note also that as stated above, $\mathbf{J}_\infty$ upper-bounds $\mathbf{J}_p$ under mild regularity assumptions.

Another element to consider is that this problem is well posed for the kind of general boundary data we want to respect, provided both at curves and isolated points on the surfaces. At least for the case $p \leq 2$, this is not true in general, see [34].

We are then looking for a Lipschitz extension of the map given at Γ_1 whose Lipschitz constant is as small as possible. Let

$$L(\Gamma_1, \Gamma_2) := \max_{x_i,x_j\in\Gamma_1} \frac{d_{\mathcal{B}_2}(y_i, y_j)}{d_{\mathcal{B}_1}(x_i, x_j)},$$

that is, the Lipschitz constant of the boundary data. In general, we will have $\inf_{\psi\in\mathcal{L}} \mathbf{J}_\infty > L(\Gamma_1, \Gamma_2)$. This is related to Kirszbraun's Theorem, which in one of its many guises states that a Lipschitz map $f : S \to I\!R^D$, $S \subset I\!R^d$, has an extension $\bar{f} : I\!R^d \to I\!R^D$ with the same Lipschitz constant as f, see [69]. In the same vein, one has Whitney and McShane extensions which apply to the case when the domain is any metric space X and the target is $I\!R$. These extensions provide functions that agree with f where boundary conditions are given and preserve the Lipschitz constant throughout X, see for example [10, 109]. The more general problem of extending $f : S \to Y$ ($S \subset X$, X and Y any metric spaces) to all X with the same Lipschitz constant is not so well understood and only partial results are known, see for example [128, 126, 127].

The idea then is to keep the distortion at the same order as that of the provided boundary conditions. In general there might be many solutions for the Problem (2). One particular class of minimizers which has recently received a lot of attention is that of *absolute* minimizers, or *absolutely minimizing Lipschitz extensions* (AMLE). Roughly speaking, the idea here is to single out those solutions of Problem (2) that also possess minimal local Lipschitz constant, again, see [10] for a general exposition, and [109] for a treatment of the case when the domain is any reasonable metric space and the target is the real line.

8.3 Proposed computational approach

If we take for example the case of p-Harmonic maps, one way of dealing with the computation of the optimal map ϕ_p is by implementing the geometric p-heat flow associated with the Euler-Lagrange equation of the functional $\mathbf{J}_p$, starting from a certain initial condition. As was explained in Chapter 6, using an implicit representation for both $\mathcal{B}_1$ and $\mathcal{B}_2$, we could obtain the partial differential equation $\mathbf{PDE}_p$ we need to solve in order to find ϕ_p. By taking the formal limit as $p \uparrow \infty$ we would find $\mathbf{PDE}_\infty$, the PDE that characterizes the solution ϕ_∞ of the (variational) Problem (2).[2] All of this might work if we had a notion of solution for the resulting PDEs. Whereas this is feasible in the case of $\mathbf{PDE}_p$ for $1 < p < \infty$, to the best of our knowledge, there is no such notion of a solution for $\mathbf{PDE}_\infty$. One could of course still persist and try to solve these equations without the necessary theoretical foundations and call these plausible solutions ∞-Harmonic Maps. Nonetheless, this is certainly an interesting line of research.

A different direction is considered in this work. As a guiding example, we first concentrate on the case where $\mathcal{B}_1$ is any closed smooth manifold and $\mathcal{B}_2$ is replaced by $I\!\!R$, as considered in [32] (for scalar data interpolation on surfaces), and in [157]. In [32], the authors propose to follow a similar path to the one we have just described, and they do not obtain a convergent numerical discretization for the resulting PDE. Meanwhile, in [157], the author proposes a convergent discretization of the PDE, basing his construction on the original variational problem. We choose to follow this idea as our guiding principle.

We now explain this alternative approach. The basic idea is simple, instead of first obtaining the Euler-Lagrange equations for $\mathbf{J}_\infty$ and then discretizing them, we will first discretize $\mathbf{J}_\infty$ and then proceed to solve the resulting discrete problem. Consider that the domain $\mathcal{B}_1$ is given discretely as a set of (different) points $\mathbb{B}_1 = \{x_1, \dots, x_m\}$ together with a neighborhood relation (i.e., a graph). To fix ideas let's assume the neighborhood relation is a k-nearest neighbors one. Denote, for each $1 \le i \le m$, by $N_i = \{x_{j_1}, \dots, x_{j_k}\} \in \mathbb{B}_1$ the set of k neighbors of the point x_i. We consider the discrete local Lipschitz constant of the map ϕ at x_i:

$$L_i(\phi) := \max_{x_j \in N_i} \frac{d_{\mathcal{B}_2}(\phi(x_i), \phi(x_j))}{d_{\mathcal{B}_1}(x_i, x_j)} \tag{8.3}$$

Upon noting that $L_i(\phi)$ serves as a discrete approximation to $\|D_{\mathcal{B}_1}\phi(x_i)\|_2$, we see that a possible discretization of the functional $\mathbf{J}_\infty(\phi)$ is given by the discrete global Lipschitz constant of ϕ given by $\max_{1 \le i \le m} L_i(\phi)$. The author of [157] proposed, in the case when $\mathcal{B}_2$ is replaced by $I\!\!R$, solving the discrete version of Problem (2) by following the following iterative procedure (here described for $\mathcal{B}_2$ a surface as in our problem):

- Let ϕ_0 be an initial guess of the map.
- For each $n \ge 1$, if $x_i \notin \Gamma_1$, let

$$\phi_n(x_i) = \arg\min_{y \in \mathcal{B}_2} \max_{j \in N_i} \frac{d_{\mathcal{B}_2}(y, \phi_{n-1}(x_j))}{d_{\mathcal{B}_1}(x_i, x_j)} \tag{8.4}$$

- $\phi_n(x_i) = y_i$ for all $n \ge 0$ for $x_i \in \Gamma_1$.

[2] The case when the domain is a subset of $I\!\!R^d$ and the target is the real line leads to the so called infinity Laplacian, see [10, 106, 52].

With computational efficiency related modifications described below, this is the approach we follow in general. The intuition behind this iterative procedure is that, at each point of the domain, we are changing the value of the map in order to minimize the local Lipschitz constant, that is, the local distortion produced by the map. This is in agreement with the notion of AMLEs briefly explained in §8.2.1. We should remark that since we are using intrinsic distances for the matching, we can let $L_i(\phi)$ play the role of (the norm of) the displacement field for analyzing the deformation,[3] see §8.5 ahead.

8.4 Implementation details

In addition to discretizing the domain $\mathbb{B}_1$, we also use a discretization $\mathbb{B}_2 = \{y_1, \ldots, y_{m'}\}$ of the target space $\mathcal{B}_2$ for our implementation. We endow $\mathbb{B}_2$ with a neighborhood relation given by the k-nearest neighbors of each point. For computational efficiency, we work at all times with two different scales in the discrete domain $\mathbb{B}_1$. We choose a subset F_1 of $\mathbb{B}_1$ such that $\#F_1 \ll m$ but still F_1 is an efficient (well separated) covering of $\mathbb{B}_1$ with small covering radius. We do this by using the well known (geodesic) Farthest Point Sampling (FPS) procedure, see [143, 150], which can be efficiently constructed based on optimal computational techniques. Roughly speaking, we apply the iterative procedure on this subset of points only and then extend the map to the rest of the points in the domain $\mathbb{B}_1$. We now show how to obtain a reasonable initial condition ϕ_0 and then discuss additional details regarding the implementation of the iterative procedure described in the previous section.

Building the initial condition: We compute, for all $x_r \in F_1 \backslash \Gamma_1$, $\phi_0(x_r) = \arg\min_{y \in \mathcal{B}_2} \max_{x_i \in \Gamma_1} \frac{d_{\mathcal{B}_2}(y, y_i)}{d_{\mathcal{B}_1}(x_r, x_i)}$. For this step we use the classical Dijkstra's algorithm for approximating the distances $d_{\mathcal{B}_1}$ and $d_{\mathcal{B}_2}$ since they might be evaluated at faraway points. This is of course run on the graphs obtained from connecting each point to its k-nearest neighbors.

The iterative procedure: After ϕ_0 is computed for all points in the set F_1, we run the iterative procedure from §8.3 on this set of points. The main modification here is that whereas we still use Dijkstra's algorithm for approximating $d_{\mathcal{B}_2}$ in the target surface, since in the domain we must compute $d_{\mathcal{B}_1}$ only for neighboring points (F_1 was chosen to be dense enough), for computationally efficiency we can approximate $d_{\mathcal{B}_1}(x_i, x_j) \simeq \|x_i - x_j\|$ for $x_j \in N_i$. We should also point out that for points in F_1, the neighborhood relation is defined to be that of k-nearest neighbors with respect to the metric on $\mathbb{B}_1$ defined by the adjacency matrix of $\mathbb{B}_1$. Let $\phi_* : F_1 \to \mathbb{B}_2$ denote the map obtained as the output of this stage.

Extension to the whole domain: After we have iterated over points in F_1 until convergence, we extend the map ϕ_* to all points x_i in $\mathbb{B}_1 \backslash \{F_1 \cup \Gamma_1\}$. This is done by computing $\phi_*(x_i) = \arg\min_{y \in \mathcal{B}_2} \max_{x \in F_1 \cup \Gamma_1} \frac{d_{\mathcal{B}_2}(y, \phi_*(x))}{d_{\mathcal{B}_1}(x_i, x)}$. For this step, and since we have already obtained the map for a relatively dense subset, we approximate both $d_{\mathcal{B}_1}$ and $d_{\mathcal{B}_2}$ by the Euclidean distance. Once again, the motivation for this is just computational efficiency.

[3] One can a situation in which two isometric surfaces are matched by our algorithm such that $L_i(\phi) = 1$ for all i, but the displacement field $\|x_i - \phi(x_i)\|$ is large since there may be no rigid motion that aligns the two surfaces. One simple example is a flat sheet of paper and the same sheet slightly bent.

8.5 Examples

In this section we present some computational examples of the ideas presented in previous sections. First, in Figure 8.1 the domain $\mathcal{B}_1$ is a *cube* ($m = 10086$) and the target $\mathcal{B}_2$ is a *sphere* ($m' = 17982$). For the purposes of visualizing the map, we assigned the clown texture (which can be thought of as a function $I : \mathbb{B}_2 \to I\!\!R$) to the sphere, which can be seen on the bottom-right corner of the figure. The sphere and the cube were concentric and of approximately the same size. We selected F_1 on the cube consisting of 1000 well separated points using the FPS procedure alluded to in §8.4. Also, we set $k = 6$ (number of neighbors). We then chose Γ_1 to be the first 100 points of the set and then projected them onto the sphere, obtaining in this way, the corresponding set Γ_2 to use as boundary conditions. We then followed the computational procedure detailed before. The top-left figure shows the composition $I \circ \phi_* : \text{cube} \to I\!\!R$ as a texture on the cube. Finally, the top-right and the bottom-left images show the histogram of $L_i(\phi_*)$ and its spatial distribution in the domain (we paint the cube at each point x_i with the color corresponding to $L_i(\phi_*)$), respectively. Ideally, we would like to obtain a δ-type histogram, meaning that the distances have been constantly scaled. Of course, this is not possible (unless one of the surfaces is isometric to a scaled version of the other), and we attempt to obtain histograms as concentrated as possible. This is quite nicely obtained for this and the additional examples in this chapter.

Figure 8.2 shows the construction of a map from the unit sphere S^2 into a cortical hemisphere $\mathbb{B}$ ($\mathcal{B}$). The boundary conditions consisted of 6 pairs of points. We first took the following 6 points on the sphere $\Gamma_1 = \{(\pm 1, 0, 0), (0, \pm 1, 0), (0, 0, \pm 1)\}$. We then constructed the intrinsic distance matrix $[d_{S^2}(p_i, p_j)]$ for all $p_i, p_j \in \Gamma_1$. Finally, we chose 6 points $\{q_1, \ldots, q_6\} = \Gamma_2$ in $\mathbb{B}$ such that $\max_{i \neq j} \frac{d_{\mathbb{B}}(q_i, q_j)}{d_{S^2}(p_i, p_j)}$ was as close as possible to $\frac{1}{\pi}\text{diam}(\mathbb{B})$. We painted $\mathbb{B}$ with a texture I_H depending on its mean curvature so as to more easily visualize the sulci/crests: If $H(x)$ stands for mean curvature of $\mathcal{B}$ at x, then $I_H(x) = (H(x) - \min_x H(x))^2$. See the caption for more details.

The example in Figure 8.3 is about computing a map Φ from a subject's left hemisphere $\mathbb{B}_1$ to another subject's left hemisphere $\mathbb{B}_2$. The boundary conditions were constructed in a way similar to the one used for the previous example, but in this case, 300 points were chosen. Note that, if available, hand traced curves could be used as commonly done in the literature (in other words, more anatomical/functional oriented boundary conditions). In the first two rows we show 4 different views of each cortical surface, and in the third row we show $\mathbb{B}_1$ colored with the values of $L_i(\Phi)$ which we interpret as a measure of the local deformation of the map needed to match $\mathbb{B}_1$ to $\mathbb{B}_2$. See the caption for more details.

8.6 Concluding remarks

In this chapter we have introduced the notions of minimizing Lipschitz extensions into the area of surface and brain warping. These maps provide a more global constraint than ordinary p-harmonic ones, and allow for more general boundary conditions. The proposed computational framework leads to an efficient surface-to-surface warping algorithm that avoids distorting intermediate steps that are common in the brain warping literature. We are currently investigating the use of this new warping technique for creating population averages and applying it to disease and growth studies. In earlier work, the Jacobian of a deformation mapping over time has been used to map the profile of brain tissue growth and loss in a subject scanned serially

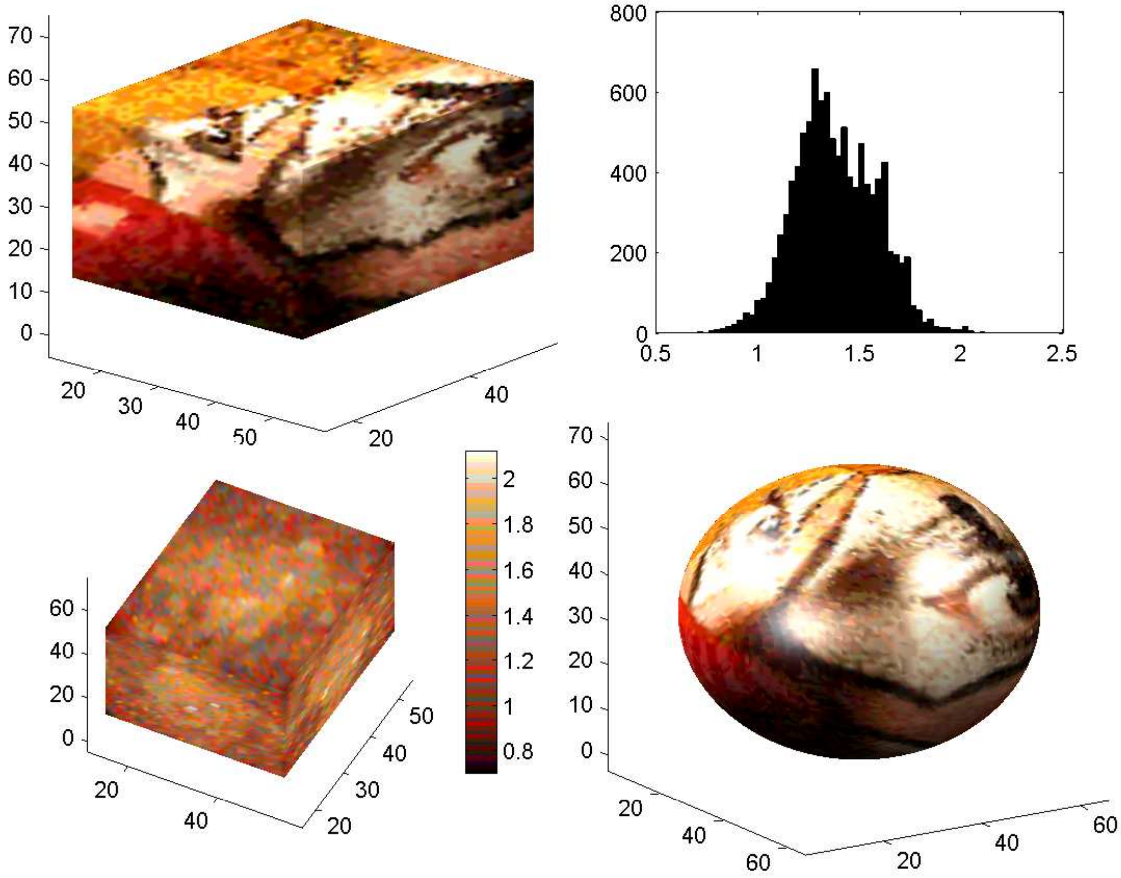


Figure 8.1: Artificial example of the proposed warping algorithm. From top to bottom and left to right: The domain surface, with a picture painted on it to help in visualizing the computed map; histogram of the Lipschitz constant (note how it is concentrated around a single value); color coded distribution of the Lipschitz constant for the computed map; and mapped texture following the computed map.

(tensor-based morphometry [192, 43]). The discrete local Lipschitz constants of our computed mappings also provide a useful index of deformation that can be analyzed statistically across subjects. The framework here introduced can also be applied in 3D for volumetric warping and with weighted geodesic distances instead of natural ones to include additional geometric characteristics in the matching. Results in these directions will be reported elsewhere.

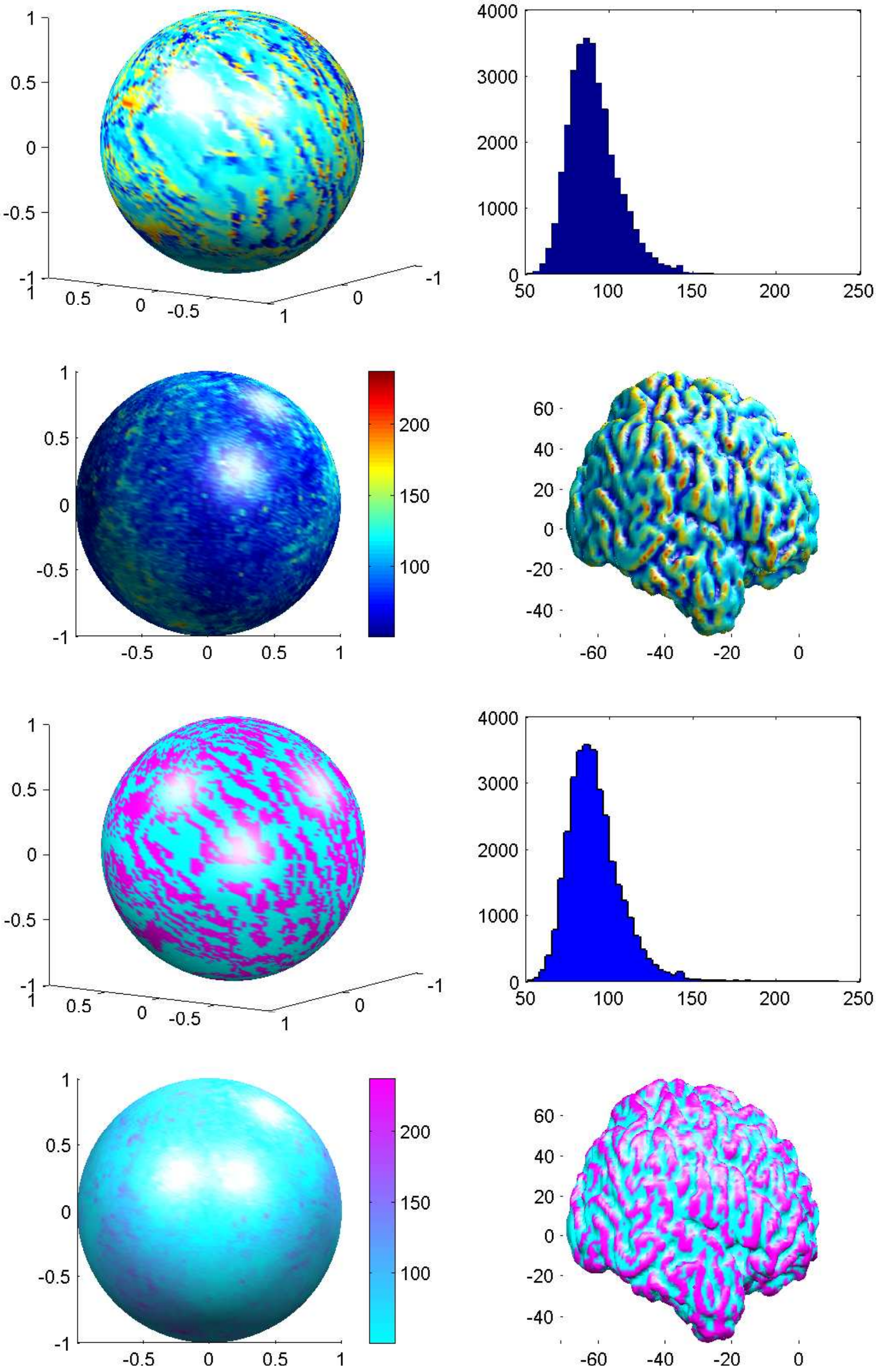


Figure 8.2: An example of mapping between the cortex and a sphere. The order is the same as in the previous figure, but now the domain and target surface are colored with a curvature-based color code. Note once again the concentration of the Lipschitz constant for the computed map. On the left, the texture map corresponding to $I_H(x)$ as described in the text is used. On the right, the texture is $\max(I_H(x), \delta)$ for a user selected value of the threshold δ, which highlights the gyral crests.

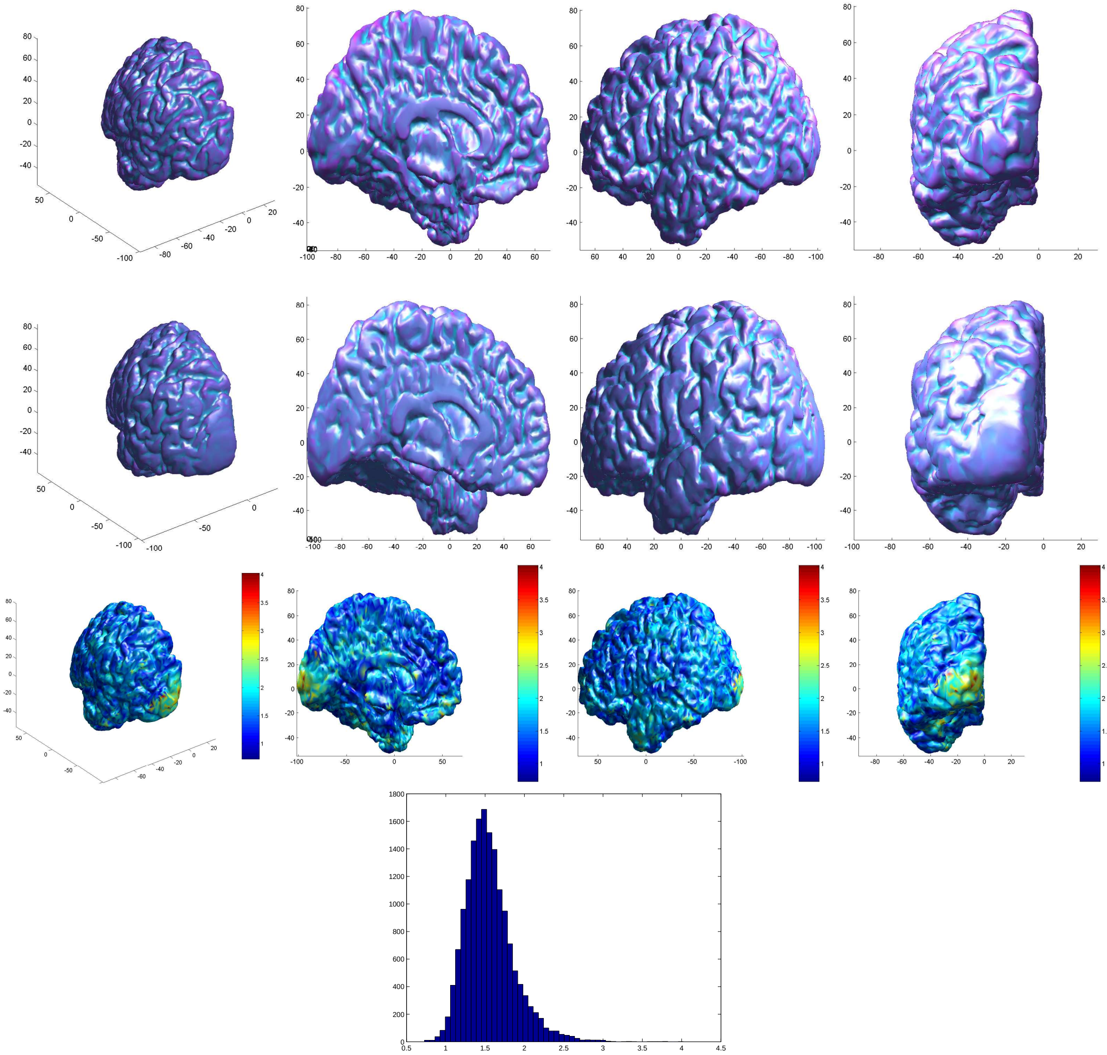


Figure 8.3: Warping between the cortical surfaces of two brains. In the first row we show 4 views of $\mathbb{B}_1$: posterior, medial, lateral and directly viewing the occipital cortex. The corresponding 4 views of $\mathbb{B}_2$ are shown in the second row. In the third row, we show $\mathbb{B}_1$ with texture $I(x_i) = L_i(\Phi)$ which can interpreted as a measure of local deformation needed to match $x_i \in \mathbb{B}_1$ to $\Phi(x_i) \in \mathbb{B}_2$. Relatively little deformation (blue colors) is required to match features across subjects on the flat interhemispheric surface (second image in the second row). This is consistent with the lower variability of the gyral pattern in the cingulate and medial frontal cortices. By contrast, there is significant expansion required to match the posterior occipital cortices of these two subjects, especially in the occipital poles which are the target of many functional imaging studies of vision. The final panel in the figure shows the corresponding histogram for $L_i(\phi)$, the local Lipschitz constants of the map.

Chapter 9

Conclusions and Future Lines of Research

In this chapter we briefly summarize the contents of the core chapters of the thesis and provide concluding remarks along with future lines of research for each of them, and then some general remarks.

Chapters 2 and 4

In this pair of chapters we have presented a novel computationally optimal algorithm for the computation of intrinsic distance functions (and geodesics) on submanifolds of euclidean space (possibly with convex boundary) given either implicitly or as random point clouds. 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. We discussed on the plausibility of tighter bounds for the error between $d^{\tilde{g}}_{\Omega_h}$ and $d^{g}_{\mathcal{S}}$. When considering a random point cloud representation of the manifold, we obtained probabilistic bounds for the error. We based this bounds on relatively simple but useful estimatives on the probability of covering the manifold by the union of balls centered at the points in the cloud, Chapter 3. As another interesting extension, it remains to characterize what is the 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, with 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.

Chapter 5

In this chapter we dealt with the comparison of shapes under certain invariance properties. More precisely, we chose to think of shapes as of metric spaces endowed with a certain metrics. We then considered thee Gromov-Hausdorff distance between metric spaces as a measure of similarity between these (shapes) metric spaces. We explained how this pre-exisiting theory can suit the problem well and proposed a computational approach. We discussed possible extension and improvements. In particular we noted that instead of only considering shapes to be metric spaces, it may be more useful to consider them as *measure metric spaces* and use analogous definitions of the Gromov-Hausdorff metric which may make the computational framework more efficient. We also discussed how the choice of the metric with which we endow each of the shapes can allow for more flexibility in the comparison process. In particular, we suggested considering *scale dependent* comparisons. Finally, the algorithmic part of the framework has been tested experimentally but some theoretical work is still needed.

Chapters 6 and 7

In Chapter 6 we have shown how to implement variational problems and partial differential equations on implicit surfaces for data taking values onto general target surfaces. We have also addressed the case of open target surfaces and sub-manifolds. The key concept is to represent the target (sub-)manifolds in implicit form, and then implement the equations in the corresponding embedding space. This framework extends the work with general domain manifolds reported in [134], thereby providing a possible approach to the computation of maps between generic manifolds. As a byproduct of the work, a dictionary has been obtained which permits the translation of intrinsic differential quantities into properly modified extrinsic ones. In Chapter 7 we showed a couple applications of the machinery developed for solving PDEs on implicit surfaces. These applications were done using Brain Imaging data. An interesting application was obtained for obtaining smooth interpolation/approximation of points on a surface.

Chapter 8

In this chapter we introduced the notions of minimizing Lipschitz extensions into the area of surface and brain warping. These maps provide a more global constraint than ordinary p-harmonic ones, and allow for more general boundary conditions. The proposed computational framework leads to an efficient surface-to-surface warping algorithm that avoids distorting intermediate steps that are common in the brain warping literature. The discrete local Lipschitz constants of our computed mappings also provide a useful index of deformation that can be analyzed statistically across subjects. This framework introduced can also be applied in 3D for volumetric warping and with weighted geodesic distances instead of natural ones to include additional geometric characteristics in the matching. The work presented in this thesis is preliminary and deeper investigations are needed in both, the theoretical and practical aspects of the work.

General Remarks

In this thesis a set of problems in Computer Vision were considered. These problems dealt with geometric data in two formats, implicit representations and point clouds. The solutions proposed attempted to provide a more or less rigorous model for the problem, a well justified solution with some theoretical guarantees and a computational framework for the application of the ideas.

Appendix A

Some Technical Ingredients

A.1 Signed Distance Functions to Hypersurfaces in Euclidean Space

We now present a few important results on distance maps. These have been mainly adapted (and adopted) from [5, 6, 83, 183].

Let $\mathcal{S}$ be a smooth closed hypersurface in $\mathbb{R}^d$ and let $\psi(x)$ denote the *signed* distance function to $\mathcal{S}$, taken positive in the exterior and negative in the interior of $\mathcal{S}$, respectively. Then, ψ is smooth we know that it satisfies the *Eikonal* equation

$$\|\nabla\psi\| = 1 \tag{A.1}$$

The signed distance function satisfies this PDE everywhere in the *viscosity sense* [104, 54]. 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 [76]:

Lemma 9 *Let $\mathcal{S}$ be a C^k $(k \geq 2)$ codimension 1 closed hyper-surface 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

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

Therefore,

$$\mathbf{H}_\psi\,\nabla\psi = 0 \tag{A.2}$$

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

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

The next Lemma, whose detailed proof can be found in [5], is mainly based in the relations (A.2) and (A.3), and it is used to verify that the function $\mu : (-\varepsilon, \varepsilon) \to \mathbb{R}^{d\times d}$ 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 10 *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 $|\epsilon| \left(\max_{1 \le i \le d-1} |\lambda_i(0)|\right) < 1$.

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

Corollary 10 *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 $\mathcal{S}$ [183]:

Theorem 14 *If $\mathcal{S} \subset I\!R^d$ is a compact C^k $(k \ge 2)$ codimension 1 hyper-surface, then there is a $h(\mathcal{S}) > 0$ such that the map $\Pi_\mathcal{S}$ is well defined and belongs to $C^{k-1}(\{x : d(x, \mathcal{S}) < h\}, I\!R^d)$.*

A.2 Properties of Squared Euclidean Distance Functions

The references for this section are [6], pp.12-16, and [69].

Theorem 15 *([6]) Let $\Gamma \subset I\!R^d$ be a compact, smooth manifold without boundary. Then $\eta(x) \overset{\triangle}{=} \frac{1}{2}d^2(\Gamma, x)$ is smooth in a tubular neighborhood U of Γ. Also, in U it satisfies $\|D\eta\|^2 = 2\eta$.*

Corollary 11 *The projection operator $\Pi : U \to \Gamma$, for a given $x \in U$, can be written as $\Pi(x) = x - D\eta(x)$. Moreover, this operator is smooth.*

Remark 24 *Differentiation of the relation $< D\eta, D\eta >= 2\eta$ gives us $D^2\eta D\eta = D\eta$. Differentiating once more we also find $D^3\eta D\eta = D^2\eta$.*

Theorem 16 *([6]) Let Γ and U be as in the previous Theorem and let $y \in U$ and $x = y - D\eta(x) \in \Gamma$, $k = dim(\Gamma)$. Then, denoting by $\lambda_1, \dots, \lambda_n$ the eigenvalues of $D^2\eta(y)$,*

$$\lambda_i(y) = \begin{cases} \frac{d(\Gamma,y)\kappa_i(x)}{1+d(\Gamma,y)\kappa_i(x)} & \textit{if } 1 \le i \le k \\ 1 & \textit{if } k < i \le n, \end{cases}$$

where $\kappa_i(x)$ are the principal curvatures of Γ at x along $Dd(\Gamma, y) \in N_x\Gamma$, where $N_x\Gamma$ is the normal space to Γ at x.

A.3 Technical Lemma

Lemma 11 *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:*

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

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

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

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 $\widehat{\varphi}\Big|_{[a,b]}$ 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 ($\rightrightarrows$ means *uniform* convergence)

(a)

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

(b)

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

(c)

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

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

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

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 $f\,f'\Big|_a^b$. 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. □

Appendix B

Appendices to Chapter 6

B.1 Physical Intuition

We now present a simple physical interpretation of harmonic maps. The argument is just intuitive, not at all rigorous. Moreover, we will restrict ourselves to simple manifolds. Let's assume that the domain manifold $\mathcal{M}$ is an interval $[a, b] \subset I\!R$, and that the target manifold $\mathcal{N}$ is a surface in $I\!R^3$. Assume that $\mathcal{M}$ is uniformly discretized and that p_i, $0 \leq i \leq L-1$, are the points in the discretization, where L defines the resolution of this division. Springs of zero natural length and constant k_L are attached to the points $\{a\}$ and $\{b\}$ and sharing the other extremes with other springs on the points inside $\mathcal{M}$.

Let $\vec{U} : \mathcal{M} \to \mathcal{N}$ be a map such that $\vec{U}(a) = p \in \mathcal{N}$, $\vec{U}(b) = q \in \mathcal{N}$. Let's consider the elastic energy of E_L of all the springs when we lay them on the surface $\mathcal{N}$:

$$E_L(\vec{U}) = \sum_{i=0}^{L-1} \frac{k_L}{2}\, d^2_{\mathcal{N}}(\vec{U}(p_i), \vec{U}(p_{i+1})$$

When we force the ends of the springs to be on $\mathcal{N}$, the elastic energy must be minimized in the equilibrium, and it characterizes the map we are seeking.

When L is large and the target manifold is sufficiently smooth, we can approximate the distances $d^2_{\mathcal{N}}(\vec{U}(p), \vec{U}(q))$ on $\mathcal{N}$ by $\|\vec{U}(p) - \vec{U}(q)\|^2$. When the vector field $\vec{U}$ is differentiable, and using the mean value theorem, we write $\vec{U}(p) - \vec{U}(q)) = \mathbf{J}_{\vec{U}}(s)\,(p-q)$, where s is an intermediate point of the interval between p and q. We then obtain

$$d^2_{\mathcal{N}}(\vec{U}(p), \vec{U}(q)) \simeq (p-q)^T \left(\mathbf{J}^T_{\vec{U}}(s)\mathbf{J}_{\vec{U}}(s)\right)(p-q).$$

Since, $\mathbf{J}_{\vec{U}} = \vec{U}_x$, the elastic energy can be approximated by

$$E_L(\vec{U}) \simeq \sum_{i=0}^{L-1} \frac{k_L}{2}\, \|\vec{U}_x(s_i)\|^2 (p_i - p_{i+1})^2 = \sum_{i=0}^{L-1} \frac{k_L}{2}\, \|\vec{U}_x(s_i)\|^2 (\frac{b-a}{L})^2$$

We must choose k_L so that for every L the springs are being pulled by the same force, obtaining $k_L = k \cdot L$, for some constant k. This is easily seen, observing that as L increases the resting length of the springs reduces as $\frac{b-a}{L}$. With this in mind, from the above expression for the energy, we obtain

$$E_L(\vec{U}) \simeq \frac{k \cdot (b-a)}{2} \sum_{i=0}^{L-1} \|\vec{U}_x(s_i)\|^2 \frac{b-a}{L} \overset{L\uparrow\infty}{\longrightarrow} \frac{k \cdot (b-a)}{2} \int_a^b \|\vec{U}_x(x)\|^2 \, dx$$

This is just the *harmonic energy* already defined.

B.2 Boundary Conditions for the Gradient Descent Flow

We now justify the use of Neumann boundary conditions for the gradient descent flow in §6.2.1. In the scalar case, one has the evolution problem

$$\begin{cases} I_t(x,t) = \Delta I(x,t) \;\; x \in \mathcal{M},\, t \geq 0 \\ I(x,0) = I_0(x), \;\; x \in \mathcal{M} \\ \nabla I \cdot \mathbf{n}|_{\partial\mathcal{M}} = \mathbf{0}. \end{cases} \tag{B.1}$$

We observe that the quantity $\sigma(t) \overset{\triangle}{=} \int_{\mathcal{M}} I(x,t)\, d_{\mathcal{M}}v$ remains constant,

$$\dot{\sigma}(t) = \int_{\mathcal{M}} I_t(x,t)\, d_{\mathcal{M}}v = \int_{\mathcal{M}} \Delta I(x,t)\, d_{\mathcal{M}}v = \int_{\mathcal{M}} \nabla \cdot (\nabla I)\, d_{\mathcal{M}}v = \int_{\partial\mathcal{M}} \nabla I \cdot \mathbf{n}\; d_{\mathcal{M}}s = 0$$

thereby imposing the boundary conditions.

One wonders which quantity is preserved thru time by the flow in the general case, when imposing the boundary condition (6.15). We illustrate this for the particular case of $\mathcal{N} = S^1$. In this case, the evolution equations are given by (see also §6.2.3)

$$\begin{cases} X_t = \Delta X + \big(\|\nabla X\|^2 + \|\nabla Y\|^2\big)X \\ Y_t = \Delta Y + \big(\|\nabla X\|^2 + \|\nabla Y\|^2\big)Y \end{cases} \tag{B.2}$$

The Neumann boundary conditions for this case are written as

$$\nabla X \cdot \mathbf{n} = \nabla Y \cdot \mathbf{n} = 0 \;\text{ in } \partial\mathcal{M}$$

Transforming to polar coordinates (ρ, θ) one finds that the evolution equations (for smooth initial data, and at least for some time) are (see also [166])

$$\begin{cases} \theta_t = \Delta\theta \\ \rho_t = 0 \end{cases} \tag{B.3}$$

with boundary conditions

$$\nabla\theta \cdot \mathbf{n} = 0 \;\text{ in } \partial\mathcal{M}$$

Again one finds that $\int_{\mathcal{M}} \theta(x,t)\, d_{\mathcal{M}}v$ is *constant.*

In the most general case, when the target manifold is arbitrary, one might guess that the *intrinsic* *barycenter*[1] of the map is preserved through time, since that's exactly what the particular cases given above show us. However, to the best of our knowledge, there is not such a result in the literature of harmonic maps, and the conservation of the barycenter is only obtained when constraints are added. The examples discussed above still motivate the use of Neumann boundary conditions.

B.3 Implicit Calculus

We now present basic facts about differential calculus on implicitly represented surfaces. For more information see for example [18, 40, 144].

We have a smooth scalar function $f : \mathbb{R}^d \to \mathbb{R}$, and a smooth vector field $\vec{\lambda} : \mathbb{R}^d \to \mathbb{R}^D$ (d and D are not necessarily equal). The manifold onto which the calculus is to be done is represented as $\mathcal{S} = \{\psi = 0\}$, for $\psi(\cdot)$ the signed distance function to $\mathcal{S}$.

All the ideas of differentiation can be obtained from simple considerations related to the restriction of the function to a geodesic curve living in the manifold. We consider an arc-length parameterized geodesic curve $\gamma : [-\epsilon, \epsilon] \to \mathcal{S}$ such that $\gamma(0) = p$ is a given point of $\mathcal{S}$. We denote $F(t) = f(\gamma(t))$ and $\vec{\Lambda}(t) = \vec{\lambda}(\gamma(t))$.

Implicit gradient

We differentiate once $F(t)$ to obtain $\dot{F}(0) = \nabla f(p) \cdot \dot{\gamma}(0)$. Since $\dot{\gamma}(0) \in T_p\mathcal{S}$ (the tangent plane), we find the implicit gradient of f at p to be $\nabla_{\mathcal{S}} f(p) = \nabla f(p) - \nabla f(p) \cdot \vec{n}(p)\ \vec{n}(p)$, where $\vec{n}(p)$ stands for the normal to the manifold at p. Since we can also write $\vec{n}(p) = \nabla\psi(p)$, we obtain

$$\nabla_{\mathcal{S}} f(p) \overset{\triangle}{=} \nabla f(p) - \big(\nabla f(p) \cdot \nabla\psi(p)\big)\ \nabla\psi(p)$$

We often use the alternative notation $\nabla\psi f$ since the definition can be applied to any level set of ψ. Note that we can write $\nabla_\psi f = \mathbf{\Pi}_{\nabla\psi} \nabla f$ where

$$\mathbf{\Pi}_{\nabla\psi} \overset{\triangle}{=} \mathbf{I} - \nabla\psi\nabla\psi^T$$

Implicit Hessian

If we compute the second derivative of F we find that $\ddot{F}(0) = \nabla f(p) \cdot \ddot{\gamma}(0) + \mathbf{H}_f[\dot{\gamma}(0), \dot{\gamma}(0)]$. Now, we know that an arc-length parameterized geodesic curve of $\mathcal{S}$ must satisfy the *harmonic maps* differential equation

$$\ddot{\gamma} + \mathbf{H}_\psi(\gamma)[\dot{\gamma}, \dot{\gamma}]\ \nabla\psi(\gamma) = 0$$

We then find that $\ddot{F}(0) = \big(\mathbf{H}_f(p) - \nabla f(p) \cdot \nabla\psi(p)\ \mathbf{H}_\psi(p)\big)[\dot{\gamma}, \dot{\gamma}]$. Again we have that $\dot{\gamma} \in T_p\mathcal{S}$, and we find the implicit Hessian of f at p to be

$$\mathbf{H}_f^{\mathcal{S}}(p) \overset{\triangle}{=} \mathbf{\Pi}_\psi \mathbf{h}_f \mathbf{\Pi}_\psi$$

where

[1] The intrinsic barycenter G of the map $\vec{u} : \Omega \to \mathcal{N}$ is defined by $G = argmin_{p\in\mathcal{N}} \frac{1}{2} \int_\Omega d^2_{\mathcal{N}}(p, \vec{u}(x))\, dx$. See [49] for more details on the barycenter.

$$\mathbf{h}_f \overset{\triangle}{=} \mathbf{H}_f(p) - \nabla f(p) \cdot \nabla \psi(p)\ \mathbf{H}_\psi(p)$$

We will frequently use the alternative notation $\mathbf{H}^\psi_f(p)$.

Implicit Laplacian

From the previous computation it's an easy exercise to compute the implicit Laplacian or Laplace-Beltrami of f since by definition $\Delta_{\mathcal{S}} f = trace\{\mathbf{H}^{\mathcal{S}}_f\}$.

For any pair of symmetric matrices $\mathbf{A}$ and $\mathbf{B}$ one has that $trace\{\mathbf{A}BA\} = \sum_i \sum_j \sum_k a_{ij} a_{ik} b_{jk}$ and $trace\{\mathbf{A}B\} = \sum_i \sum_j a_{ij} b_{ij}$. Now we have that $\mathbf{\Pi}_\psi B \mathbf{\Pi}_\psi = \mathbf{B} + \nabla\psi\nabla\psi^T\mathbf{B}\nabla\psi\nabla\psi^T - \nabla\psi\nabla\psi^T\mathbf{B} - \mathbf{B}\nabla\psi\nabla\psi^T$. We then obtain

$$\begin{aligned} trace\{\mathbf{\Pi}_\psi B \mathbf{\Pi}_\psi\} &= trace\{\mathbf{B}\} + \sum_i \sum_j \sum_k \psi_{x_i}\psi_{x_j}\psi_{x_i}\psi_{x_k} b_{jk} \\ &- 2\sum_i \sum_j \psi_{x_i}\psi_{x_j} b_{ij} \end{aligned}$$

Recalling that $\psi(\cdot)$ is a *distance function*, so that it satisfies $\|\nabla\psi\| = 1$, we find

$$\begin{aligned} trace\{\mathbf{\Pi}_\psi B \mathbf{\Pi}_\psi\} &= trace\{\mathbf{B}\} - \sum_i \sum_j \psi_{x_i}\psi_{x_j} b_{ij} \\ &= trace\{\mathbf{B}\} - \mathbf{B}[\nabla\psi, \nabla\psi] \end{aligned}$$

We conclude the reasoning by taking $\mathbf{B} = \mathbf{h}_f$:

$$\begin{aligned} trace\{\mathbf{H}^{\mathcal{S}}_f\} &= trace\{\mathbf{h}_f\} - \mathbf{h}_f[\nabla\psi, \nabla\psi] \\ &= trace\{\mathbf{h}_f\} - \mathbf{H}_f[\nabla\psi, \nabla\psi] \end{aligned}$$

since $\mathbf{H}_\psi[\nabla\psi, \nabla\psi] = 0$. Since $trace\{\mathbf{H}_f\} = \Delta f\ -\ (\nabla f \cdot \nabla\psi)\Delta\psi$, we find that

$$\Delta_{\mathcal{S}} f = \Delta f - (\nabla f \cdot \nabla\psi)\Delta\psi - \mathbf{H}_f[\nabla\psi, \nabla\psi]$$

It's interesting to observe how the expression just found for $\Delta_{\mathcal{S}} f$ coincides with the one obtained by minimizing the *intrinsic Dirichlet integral*,[2]

$$D(f) \overset{\triangle}{=} \frac{1}{2}\int_{I\!R^d} \|\nabla_{\mathcal{S}} f\|^2\, \delta(\psi)\, dv$$

as is done in [134]. The authors showed that a smooth function f extremizing $D(f)$ must satisfy

$$\nabla \cdot \big(\nabla f - (\nabla f \cdot \nabla\psi)\ \nabla\psi\big) = 0$$

[2] As one expects since this is the definition of *harmonic functions.*

We should verify that this definition coincides with ours. This is accomplished as follows:

$$\begin{aligned}\nabla \cdot \big(\nabla f - (\nabla f \cdot \nabla\psi)\ \nabla\psi\big) &= \Delta f - (\nabla f \cdot \nabla\psi)\Delta\psi - \nabla\big(\nabla f \cdot \nabla\psi\big) \cdot \nabla\psi \\ &= \Delta f - (\nabla f \cdot \nabla\psi)\Delta\psi - \mathbf{H}_f[\nabla\psi, \nabla\psi] - \mathbf{H}_\psi[\nabla f, \nabla\psi] \\ &= \Delta f - (\nabla f \cdot \nabla\psi)\Delta\psi - \mathbf{H}_f[\nabla\psi, \nabla\psi] \\ &= \Delta_{\mathcal{S}} f \ \ \text{(according to our definition)},\end{aligned}$$

since $\mathbf{H}_\psi[\nabla\psi, \bullet] = 0$.

Vector Calculus

- **Implicit Jacobian:** With the ideas developed before, we easily find (differentiating $\vec{\Lambda}(t)$) that

$$\mathbf{J}^{\mathcal{S}}_{\vec{\lambda}} \stackrel{\triangle}{=} J_{\vec{\lambda}}\mathbf{\Pi}_\psi$$

- **Implicit Divergence:** Using the expression for the intrinsic Jacobian we write

$$\nabla_{\mathcal{S}} \cdot \vec{\lambda} \stackrel{\triangle}{=} trace\left(\mathbf{J}_{\vec{\lambda}}\mathbf{\Pi}_\psi\right)$$

  and

$$\nabla_{\mathcal{S}} \cdot \vec{\lambda} \stackrel{\triangle}{=} \nabla \cdot \vec{\lambda} - \mathbf{J}_{\vec{\lambda}}[\nabla\psi, \nabla\psi]$$

  It is useful to observe that $\nabla_{\mathcal{S}} \cdot \vec{\lambda} = \nabla \cdot \vec{\lambda}$ when $\vec{\lambda}(x) \in T_x\{\psi = 0\}$

Bibliography

[1] David Adalsteinsson and J. A. Sethian. Transport and diffusion of material quantities on propagating interfaces via level set methods. *J. Comput. Phys.*, 185(1):271–288, 2003.

[2] Ralph Alexander and S. Alexander. Geodesics in Riemannian manifolds-with-boundary. *Indiana Univ. Math. J.*, 30(4):481–488, 1981.

[3] Stephanie B. Alexander, I. David Berg, and Richard L. Bishop. The Riemannian obstacle problem. *Illinois J. Math.*, 31(1):167–184, 1987.

[4] F. Alouges. An energy decreasing algorithm for harmonic maps. In J.M.Coron et al., editor, *Nematics*, Nato ASI Series, pages 1–13. Kluwer Academic Publishers, Netherlands, 1991.

[5] L. Ambrosio and N. Dancer. *Calculus of variations and partial differential equations*. Springer-Verlag, Berlin, 2000.

[6] Luigi Ambrosio and Halil Mete Soner. Level set approach to mean curvature flow in arbitrary codimension. *J. Differential Geom.*, 43(4):693–737, 1996.

[7] Nina Amenta, Sunghee Choi, and Ravi Krishna Kolluri. The power crust, unions of balls, and the medial axis transform. *Comput. Geom.*, 19(2-3):127–153, 2001.

[8] S Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. Conformal geometry and brain flattening. In *Proc. MICCAI*, pages 271–278. MICCAI, 1999.

[9] Tom M. Apostol. *Mathematical analysis*. Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1974.

[10] Gunnar Aronsson, Michael G. Crandall, and Petri Juutinen. A tour of the theory of absolutely minimizing functions. *Bull. Amer. Math. Soc. (N.S.)*, 41(4):439–505 (electronic), 2004.

[11] J. Ashburner and K. J. Friston. Nonlinear spatial normalization using basis functions. *Human Brain Mapping*, 7(4):254–266, 1999.

[12] G. Sapiro B. Tang and V. Caselles. Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case. *Int. Journal Computer Vision*, 36(2):149–161, 2000.

[13] G. Sapiro B. Tang and V. Caselles. Chromaticity diffusion. *IEEE Trans. Image Processing*, (1), May 2001.

[14] Chandrajit L. Bajaj and Guoliang Xu. Anisotropic diffusion of surfaces and functions on surfaces. *ACM Trans. Graph.*, 22(1):4–32, 2003.

[15] A. Bartesaghi and G. Sapiro. A system for the generation of curves on 3d brain images. *Human Brain Mapping*, 14(1):1–15, 2001.

[16] Timothy J. Barth and James A. Sethian. Numerical schemes for the Hamilton-Jacobi and level set equations on triangulated domains. *J. Comput. Phys.*, 145(1):1–40, 1998.

[17] M. Bernstein, V. de Silva, J. Langford, and J. Tenenbaum. Graph approximations to geodesics on embedded manifolds. Technical report.

[18] M. Bertalmio. *Processing of Flat and Non-Flat Image Information on arbitrary manifolds using PDEs.* PhD thesis, UMN, 2001.

[19] J. Bloomenthal. *Introduction to Implicit Surfaces.*

[20] J-D. Boissonnat and F. Cazals. Coarse-to-fine surface simplification with geometric guarantees. In *Proc. Eurographics Symp. on Comput. Geom.*, 2001.

[21] Ingwer Borg and Patrick Groenen. *Modern multidimensional scaling.* Springer Series in Statistics. Springer-Verlag, New York, 1997.

[22] J. Borwein and O. Hijab.
`http://www.siam.org/journals/problems/downloadfiles/99-5sii.pdf`.

[23] Mario Botsch, Andreas Wiratanaya, and Leif Kobbelt. Efficient high quality rendering of point sampled geometry. In *Proceedings of the 13th Eurographics workshop on Rendering*, pages 53–64. Eurographics Association, 2002.

[24] Michelle Boué and Paul Dupuis. Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control. *SIAM J. Numer. Anal.*, 36(3):667–695 (electronic), 1999.

[25] M. Boutin and G. Kemper. On reconstructing n-point configurations from the distribution of distances or areas. *Adv. Appl. Math*, 32(4):705–735, 2004.

[26] A. Bronstein, M. Bronstein, A. Spira, and Ron Kimmel. Face recognition from facial surface metric. In *Proceedings of ECCV*, number II, pages 225–237. ECCV, 2004.

[27] A. M. Bruckstein. On shape from shading. *Comp. Vision Graph. Image Processing*, 44, 1988.

[28] D. Burago, Y. Burago, and S. Ivanov. *A Course in Metric Geometry*, volume 33 of *AMS Graduate Studies in Math.* American Mathematical Society, 2001.

[29] D. Burago and B. Kleiner. Separated nets in euclidean space and jacobians of bi-lipschitz maps. *Geom. Funct. Anal.*, 8:273–282, 1998.

[30] D. Burago and B. Kleiner. Rectifying separated nets. *Geom. Funct. Anal.*, 12:80–92, 2002.

[31] G. Carlsson, A. Zomorodian, A. Collins, and L. Guibas. Persistence barcodes for shapes. In *Proc. Symposium on Geometry Processing, Nice, France*, 2004.

[32] V. Caselles, L. Igual, and O. Sander. An axiomatic approach to scalar data interpolation on surfaces. Technical report, Math. Department, U. Pompeu Fabra. `http://www.iua.upf.es/ vcaselles/Papers2004/AMLEmanifoldsNumer.zip`.

[33] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces based object segmentation. *IEEE-PAMI*, 19(4):1394–398, 1997.

[34] Vicent Caselles, Jean-Michel Morel, and Catalina Sbert. An axiomatic approach to image interpolation. *IEEE Trans. Image Process.*, 7(3):376–386, 1998.

[35] T. Chan and J. Shen. Variational restoration of non-flat image features: Models and algorithms. Technical Report 9920, UCLA-CAM, June 1999.

[36] G. Charpiat, O. Faugeras, and R. Keriven. Shape metrics, warping, and statistics. In *Proceedings of the International Conference on Image Processing*, 2003.

[37] Isaac Chavel. *Riemannian geometry—a modern introduction*, volume 108 of *Cambridge Tracts in Mathematics*. Cambridge University Press, Cambridge, 1993.

[38] S. Chen, B. Merriman, and S. Osherand P. Smereka. A simple level set method for solving stefan problem. *Journal of Computational Physics*, (135):8, 1995.

[39] Li-Tien Cheng, Paul Burchard, Barry Merriman, and Stanley Osher. Motion of curves constrained on surfaces using a level-set approach. *J. Comput. Phys.*, 175(2):604–644, 2002.

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

[41] David L. Chopp. Some improvements of the fast marching method. *SIAM J. Sci. Comput.*, 23(1):230–244 (electronic), 2001.

[42] G. E. Christensen, R. D. Rabbitt, and M. I. Miller. A deformable neuroanatomy textbook based on viscous fluid mechanics. In *Proceedings of the 1993 Conference on Information Sciences and Systems*, pages 211–216, 1993.

[43] M.K. Chung, K.J. Worsley, S. Robbins, and A.C. Evans. Tensor-based brain surface modeling and analysis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, number I, pages 467–473, 2003.

[44] M.K. Chung, K.J. Worsley, S. Robbins, T. Paus, J. Taylor, J.N. Giedd, J.L. Rapoport, and A.C. Evans. Deformation-based surface morphometry applied to gray matter deformation. *NeuroImage*, 18(2):198–213, 2003.

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

[46] R. Coifman. Personal Comunication.

[47] A. Collins, A. Zomorodian, G. Carlsson, and L. Guibas. A barcode shape descriptor for curve point cloud data. In *Proc. Symposium on Point-Based Graphics, ETH, Zürich, Switzerland*, 2004.

[48] D. L. Collins, T. M. Peters, and A. C. Evans. An automated 3d non-linear image deformation procedure for determination of gross morphometric variability in the human brain. In *Proc. Visualization in Biomed. Comp. (SPIE)*, pages 180–190, 1994.

[49] José Manuel Corcuera and Wilfrid S. Kendall. Riemannian barycentres and geodesic convexity. *Math. Proc. Cambridge Philos. Soc.*, 127(2):253–269, 1999.

[50] J. M. Coron and R. Gulliver. Minimizing p-harmonic maps into spheres. *J. Reine Angew. Mathem.*, (401):82–100, 1989.

[51] J. A. Costa and A. O. Hero. Geodesic entropic graphs for dimension and entropy estimation in manifold learning. *to appear in IEEE Trans. on Signal Processing*, August, 2004.

[52] M. G. Crandall, L. C. Evans, and R. F. Gariepy. Optimal Lipschitz extensions and the infinity Laplacian. *Calc. Var. Partial Differential Equations*, 13(2):123–139, 2001.

[53] M. G. Crandall and P.-L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Math. Comp.*, 43(167):1–19, 1984.

[54] Michael G. Crandall and Pierre-Louis Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Amer. Math. Soc.*, 277(1):1–42, 1983.

[55] C. Davatzikos. Spatial normalization of 3d brain images using deformable models. *J. Comp. Assisted Tomography*, 20(4):656–665, 1996.

[56] A. Desoulnex, L. Moisan, and J.M. Morel. A theory for digital image analysis. unpublished lecture notes.

[57] T. K. Dey, J. Giesen, and J. Hudson. Decimating samples for mesh simplification. In *Proc. 13th Canadian Conference on Computational Geometry*, pages 85–88, 2001.

[58] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numer. Math.*, 1:269–271, 1959.

[59] Manfredo Perdigão do Carmo. *Riemannian geometry.* Mathematics: Theory & Applications. Birkhäuser Boston Inc., Boston, MA, 1992.

[60] Aryeh Dvoretzky. On covering a circle by randomly placed arcs. *Proc. Nat. Acad. Sci. U.S.A.*, 42:199–203, 1956.

[61] N. Dyn, M. S. Floater, and A. Iske. Adaptive thinning for bivariate scattered data. *J. Comput. Appl. Math.*, 145(2):505–517, 2002.

[62] Weinan E and X.P. Wang. Numerical methods for the landau-lifshitz equation. Technical report, Math. Department, Princeton. `http://www.math.princeton.edu/~weinan/papers/LL1.pdf`.

[63] M. Eck and H. Hoppe. Automatic reconstruction of b-spline surfaces of arbitrary topological type. In *Computer Graphics 1996*, 1996.

[64] J. Eells and L. Lemarie. A report on harmonic maps. *Bull. London Math. Soc.*, 10(1):1–68, 1978.

[65] J. Eells and L. Lemarie. Another report on harmonic maps. *Bull. London Math. Soc.*, 20:385–524, 1988.

[66] A. Elad (Elbaz) and R. Kimmel. Bending invariant representations for surfaces. *Proc. of CVPR'01 Hawaii*, 2001.

[67] D. C. Van Essen, H. Drury, S. Joshi, and M. I. Miller. Comparisons between human and macaque using shape-based deformation algorithms applied to cortical flat maps. *Neuroimage*, 5(4).

[68] Ali Fardoun and Rachid Regbaoui. Heat flow for p-harmonic maps with small initial data. *Calc. Var. Partial Differential Equations*, 16(1):1–16, 2003.

[69] Herbert Federer. Curvature measures. *Trans. Amer. Math. Soc.*, 93:418–491, 1959.

[70] W. Feller. *An Introduction to Probability Theory and its Applications.* John Wiley & Sons, Inc., New York-London-Sydney, 1971.

[71] B. Fischl, M. I. Sereno, R. B. H. Tootell, and A. M. Dale. High-resolution inter-subject averaging and a coordinate system for the cortical surface. *Human Brain Mapping*, 8(4):272–284, 1999.

[72] Leopold Flatto. A limit theorem for random coverings of a circle. *Israel J. Math.*, 15:167–184, 1973.

[73] Leopold Flatto and Donald J. Newman. Random coverings. *Acta Math.*, 138(3–4):241–264, 1977.

[74] *Flujos Toolbox for Front Evolutions.*
`http://www.iie.edu.uy/investigacion/grupos/gti/flujos/flujos.html`.

[75] S. Fomel. A variational formulation of the fast marching eikonal solver. Technical report, Stanford Exploration Project, May 2000.

[76] Robert L. Foote. Regularity of the distance function. *Proc. Amer. Math. Soc.*, 92(1):153–155, 1984.

[77] N. C. Fox, W. R. Crum, and R. I. Scahill. Imaging of onset and progression of alzheimer's disease with voxel-compression mapping of serial magnetic resonance images. *Lancet*, 358:201–205, 2001.

[78] P. A. Freeborough and N. C. Fox. Modeling brain deformations in alzheimer's disease by fluid registration of serial 3d mr images. *J. Comput. Assist. Tomogr.*, 22:838–843, 1998.

[79] Sarah F. Frisken, Ronald N. Perry, Alyn P. Rockwood, and Thouis R. Jones. Adaptively sampled distance fields: a general representation of shape for computer graphics. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 249–254. ACM Press/Addison-Wesley Publishing Co., 2000.

[80] P. Frosini. A distance for similarity classes of submanifolds of a euclidean space. *Bull. Austral. Math. Soc.*, 42:3:407–416, 1990.

[81] Joachim Giesen and Uli Wagner. Shape dimension and intrinsic metric from samples of manifolds with high co-dimension. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 329–337. ACM Press, 2003.

[82] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order.* Classics in Mathematics. Springer-Verlag, Berlin, 2001.

[83] J. Gomes and O. Faugeras. Representing and evolving smooth manifolds of arbitrary codimension embedded in r^n as the intersection of n hypersurfaces: The vector distance functions. Technical Report RR-4012, INRIA, Oct 2000.

[84] M.T. Goodrich, J.S.B. Mitchell, and M.W. Orletsky. Approximate geometric pattern matching under rigid motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(4):371–376, 1999.

[85] Alfred Gray. *Tubes.* Addison-Wesley Publishing Company Advanced Book Program, Redwood City, CA, 1990.

[86] M. Gromov. Asymptotic invariants of infinite groups. In A. Niblo and Martin A. Roller, editors, *Geometric group theory*, volume 2 of *London Math. Soc. Lecture Notes Ser.*, pages 1–295. Cambridge Univ. Press, Cambridge, UK, 1993.

[87] M. Gromov. *Metric Structures for Riemannian and Non-Riemannian Spaces*, volume 152 of *Progress in Mathematics.* Birkhäuser, Boston, US, 1999.

[88] M. Gross, H. Pfister, M. Zwicker, M. Pauly, M. Stamminger, and M. Alexa. Point based computer graphics. *EUROGRAPHICS 2002 Lecture Notes.*

[89] K. Grove. Metric differential geometry. In *Differential geometry*, volume 1263 of *Lecture Notes in Math.*, pages 171–227. Springer, Berlin, 1987.

[90] X. Gu, Y.L. Wang, T.F. Chan, P.M. Thompson, , and S.T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging*, 23(7), 2004.

[91] Bertil Gustafsson, Heinz-Otto Kreiss, and Joseph Oliger. *Time dependent problems and difference methods.* Pure and Applied Mathematics (New York). John Wiley & Sons Inc., New York, 1995.

[92] J. M. Coron H. Brezis and E. H. Lieb. Harmonic maps with defects. *Communications in Mathematical Physics*, (107):649–705, 1986.

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

[94] S. Haker, L. Zhu, A. Tannenbaum, and S. Angenent. Optimal mass transport for registration and warping. *International Journal on Computer Vision*, 60(3):225–240, 2004.

[95] Peter Hall. *Introduction to the theory of coverage processes.* Wiley Series in Probability and Mathematical Statistics: Probability and Mathematical Statistics. John Wiley & Sons Inc., New York, 1988.

[96] Richard S. Hamilton. *Harmonic maps of manifolds with boundary.* Springer-Verlag, Berlin, 1975.

[97] A. Ben Hamza and Hamid Krim. Geodesic object representation and recognition. In *Lecture Notes in Computer Science*, volume 2886, pages 378–387. 2003.

[98] R. M. Hardt. Singularities of harmonic maps. *Bulletin of the American Mathematical Society*, 34(1):15–34, 1997.

[99] J. Helmsen, E. G. Puckett, P. Collela, and M. Dorr. Two new methods for simulating photolithography development in 3d. In *Proc. SPIE Microlithography IX*, pages 253–261, 1996.

[100] P. Hoch and M. Rascle. Hamilton-jacobi equations on a manifold and applications to grid generation or refinement. Technical report, UNICE Math. Dept., Nice, France. `http://www-math.unice.fr/∼hoch/psfiles/hamja.ps`.

[101] Jørgen Hoffmann-Jørgensen. Coverings of metric spaces with randomly placed balls. *Math. Scand.*, 32:169–186 (1974), 1973.

[102] M. K. Hurdal, P. L. Bowers, K. Stephenson, D. W. L. Sumners, K. Rehm, K. Schaper, and D. A. Rottenberg. Quasi-conformally flat mapping the human cerebellum. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI'99*, 1999.

[103] D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge. Comparing images using the hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:9, 1993.

[104] Hitoshi Ishii. A simple, direct proof of uniqueness for solutions of the Hamilton-Jacobi equations of eikonal type. *Proc. Amer. Math. Soc.*, 100(2):247–251, 1987.

[105] Svante Janson. Random coverings in several dimensions. *Acta Math.*, 156(1-2):83–118, 1986.

[106] Robert Jensen. Uniqueness of Lipschitz extensions: minimizing the sup norm of the gradient. *Arch. Rational Mech. Anal.*, 123(1):51–74, 1993.

[107] Jürgen Jost. *Riemannian geometry and geometric analysis.* Universitext. Springer-Verlag, Berlin, 2002.

[108] L. Ju, J. Stern, K. Rehm, K. Schaper, M. Hurdal, and D. Rottenberg. Cortical surface flattening using least square conformal mapping with minimal metric distortion. In *Proceedings of the Second IEEE International Symposium on Biomedical Imaging*, pages 77–80, 2004.

[109] Petri Juutinen. Absolutely minimizing Lipschitz extensions on a metric space. *Ann. Acad. Sci. Fenn. Math.*, 27(1):57–67, 2002.

[110] D. W. Kahn. *Topology. An Introduction to the Point-Set and Algebraic Areas.* Williams & Wilkins Co., Baltimore, MD, 1975.

[111] Thomas Kailath. *Linear systems.* Prentice-Hall Inc., Englewood Cliffs, N.J., 1980.

[112] N. J. Kalton and M. I. Ostrovskii. Distances between banach spaces. *Forum Math.*, 11:1:17–48, 1999.

[113] T. Kanai, H. Suzuki, and F. Kimura. Three dimensional geometric metamorphosis based on harmonic maps. *The Visual Computer*, 14(4):166–176, 1998.

[114] M. G. Kendall and P. A. P. Moran. *Geometrical probability.* Griffin's Statistical Monographs & Courses, No. 10. Hafner Publishing Co., New York, 1963.

[115] 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):1260–1265, 1998.

[116] Satyanad Kichenassamy, Arun Kumar, Peter Olver, Allen Tannenbaum, and Anthony Yezzi, Jr. Conformal curvature flows: from phase transitions to active vision. *Arch. Rational Mech. Anal.*, 134(3):275–301, 1996.

[117] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. *Proc. Natl. Acad. Sci. USA*, 95(15):8431–8435 (electronic), 1998.

[118] R. Kimmel and N Sochen. Orientation diffusion or how to comb a porcupine. *Journal of Visual Communication and Image Representation*, 13(1-2):238–248, 2002.

[119] Ron Kimmel. *Numerical geometry of images.* Springer-Verlag, New York, 2004.

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

[121] A. N. Kolmogorov and S. V. Fomin. *Elementy teorii funktsii i funktsionalnogo analiza.* Second edition, revised and augmented. Izdat. "Nauka", Moscow, 1968.

[122] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. *Computer Graphics*, pages 313–324, 1996.

[123] R. Kunze, F. Wolter, and T. Rausch. Geodesic voronoi diagrams on parametric surfaces. In *Proceedings of the 1997 Conference on Computer Graphics International*, page 230. IEEE Computer Society, 1997.

[124] F. Lafon and S. Osher. High order two-dimensional nonoscillatory methods for solving Hamilton-Jacobi scalar equations. *J. Comput. Phys.*, 123(2):235–253, 1996.

[125] S. Lafon. *Diffusion maps and geometric harmonics.* PhD thesis, Yale University, May 2004.

[126] U. Lang, B. Pavlović, and V. Schroeder. Extensions of Lipschitz maps into Hadamard spaces. *Geom. Funct. Anal.*, 10(6):1527–1553, 2000.

[127] U. Lang and V. Schroeder. Kirszbraun's theorem and metric spaces of bounded curvature. *Geom. Funct. Anal.*, 7(3):535–560, 1997.

[128] Urs Lang. Extendability of large-scale Lipschitz maps. *Trans. Amer. Math. Soc.*, 351(10):3975–3988, 1999.

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

[130] John M. Lee. *Riemannian manifolds*, volume 176 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 1997.

[131] Greg Leibon and David Letscher. Delaunay triangulations and voronoi diagrams for riemannian manifolds. In *Proceedings of the sixteenth annual symposium on Computational geometry*, pages 341–349. ACM Press, 2000.

[132] A. Leow, C. L. Yu, S. J. Lee, S. C. Huang, R. Nicolson, K. M. Hayashi, H. Protas, A. W. Toga, and P. M. Thompson. Brain structural mapping using a novel hybrid implicit/explicit framework based on the level-set method. *NeuroImage*, 2005.

[133] L. Linsen and H. Prautzsch. Local versus global triangulations. In *Proc. Eurographics '01*, 2001.

[134] S. Osher M. Bertalmí o, L.T. Cheng and G. Sapiro. Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics*, (174):759–780, December 2001.

[135] P. Schröder M. Desbrun, M. Meyer and A. H. Barr. Discrete differential-geometry operators in *nd*. Technical report, Multi-res modeling group TR, Caltech, September 2000.

[136] T. Duchamp H. Hoppe M. Lounsbery M. Eck, T. DeRose and W. Stuetzle. Multi-resolution analysis of arbitrary meshes. In *Computer Graphics (SIGGRAPH '95 Proceedings)*, pages 173–182, 1995.

[137] G. Modica M. Giaquinta and J. Soucek. Variational problems for maps of bounded variation with values in s^1. *Cal. Var.*, (1):87–121, 1993.

[138] Carlo Mantegazza and Andrea Carlo Mennucci. Hamilton-Jacobi equations and distance functions on Riemannian manifolds. *Appl. Math. Optim.*, 47(1):1–25, 2003.

[139] Antonio Marino and D. Scolozzi. Geodesics with obstacles. *Boll. Un. Mat. Ital. B (6)*, 2(1):1–31, 1983.

[140] S. Mauch. Closest point transform. Technical report, Caltech, December 2000.
`www.ama.caltech.edu/∼seanm/software/cpt/cpt.html`.

[141] C. T. McMullen. Lipschitz maps and nets in euclidean space. *Geom. Funct. Anal.*, 8:304–314, 1998.

[142] F. Mémoli and G. Sapiro. Distance functions and geodesics on point clouds. Technical Report 1902, Insitute for Mathematics and its Applications, University of Minnesota, Minneapolis, USA, 2003. `http://www.ima.umn.edu/preprints/dec2002/1902.pdf, to appear in SIAM J. Applied Math.`

[143] F. Mémoli and G. Sapiro. A theoretical and computational framework for isometry invariant recognition of point cloud data. Technical Report 1980, University of Minnesota,IMA, 2004.
`http://www.ima.umn.edu/preprints/jun2004/1980.pdf`.

[144] Facundo. Mémoli. Distance maps on implicitly defined manifolds. Master's thesis, Universidad de la Republica, Uruguay., 2001.

[145] Facundo Mémoli and Guillermo Sapiro. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *J. Comput. Phys.*, 173(2):730–764, 2001.

[146] M. I. Miller, A. Trouve, and L. Younes. On the metrics and euler-lagrange equations of computational anatomy. *Ann. Rev. Biomed Eng.*, 4:375–405, 2002.

[147] J.S.B. Mitchell. An algorithmic approach to some problems in terrain navigation. *Artificial Intelligence*, 37:171–201.

[148] J.S.B. Mitchell, D. Payton, and D. Keirsey. Planning and reasoning for autonomous vehicle control. *International Journal of Intelligent Systems*, 2:129–198.

[149] N. J. Mitra and A. Nguyen. Estimating surface normals in noisy point cloud data. In *Proc. 19th Conf. on Comput. Geom.*, pages 322–328, 2003.

[150] C. Moenning and N. A. Dodgson. Fast marching farthest point sampling for implicit surfaces and point clouds. Technical Report 565, Cambridge University Computer Laboratory, Cambridge, UK, 2003. `http://www.cl.cam.ac.uk/users/cm230/docs/pdfs/FastFPS.pdf`.

[151] C. Moenning, F. Mémoli, G. Sapiro, N. Dyn, and N. A. Dodgson. Meshless geometric subdivision. Technical Report 1977, Institute for Mathematics and its Applications, University of Minnesota, Minneapolis, USA, 2004. `http://www.ima.umn.edu/preprints/apr2004/1977.pdf`.

[152] R. Kimmel N. Sochen and Malladi R. A general framework for low level vision. *IEEE Trans. Image Processing*, (7):310–318, 1998.

[153] V.V. Nekrashevych. On equivalence of nets in hyperbolic spaces. *Dopov. Nats. Akad. Nauk Ukr. Mat. Prirodozn. Tekh. Nauki*, pages 18–21, 1997.

[154] Seiki Nishikawa. On the Neumann problem for the nonlinear parabolic equation of Eells-Sampson and harmonic mappings. *Math. Ann.*, 249(2):177–190, 1980.

[155] Partha Niyogi, Shmuel Weinberger, and Stephen Smale. Finding the homology of submanifolds with high confidence from random samples. Technical Report TR-2004-08, Department of Computer Science, University of Chicago., November, 2004.

[156] R. Deriche R. Keriven T. Papadopoulo J. Gomes G. Hermosillo P. Kornprobst D. Lingrad J. Roberts T. Viéville F. Devernay O. Faugeras, F. Clément. The inverse eeg and meg problems: The adjoint state approach i: The continuous case. Technical Report RR-3673, INRIA, June 1999.

[157] Adam Oberman. Convergent difference schemes for the infinity laplacian: Construction of absolutely minimizing lipschitz extensions. *Math. Comp.* to appear.

[158] S. J. Osher and R. P. Fedkiw. Level set methods: An overview and some recent results. *J. Comput. Phys.*, (169):463–502, 2001.

[159] S. J. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, (79):12–49, 1988.

[160] Stanley Osher. A level set formulation for the solution of the Dirichlet problem for Hamilton-Jacobi equations. *SIAM J. Math. Anal.*, 24(5):1145–1152, 1993.

[161] A. Pardo and G. Sapiro. Vector probability diffusion. *IEEE Signal Processing Letters*, 8:106–109, 2001.

[162] M. Pauly and M. Gross. Spectral processing of points sampled geometry. In *ACM SIGGRAPH*, pages 379–386, 2001.

[163] M. Pauly, M. Gross, and L. P. Kobbelt. Efficient simplification of point-sampled surfaces. In *Proc. 13th IEEE Visualization*, pages 163–170, 2002.

[164] M. Pauly, N. J. Mitra, and L. Guibas. Uncertainty and variability in point cloud surface data. In *Symposium on Point-Based Graphics, Zurich 2004*, 2004.

[165] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A pde-based fast local level set method. *J. Comput. Phys.*, (155):410–438, 1999.

[166] P. Perona. Orientation diffusion. *IEEE Trans. Image Processing*, (7):457–467, 1998.

[167] P. Petersen. Gromov-hausdorff convergence of metric spaces. In *Differential geometry: Riemannian geometry, Proc. Sympos. Pure Math.*, volume 54 Part 3, pages 489–504. Amer. Math. Soc., 1993.

[168] P. Petersen. *Riemannian Geometry.* Springer-Verlag, New York, US, 1998.

[169] Franco P. Preparata and Michael Ian Shamos. *Computational geometry.* Texts and Monographs in Computer Science. Springer-Verlag, New York, 1985.

[170] The Digital Michelangelo Project.

[171] D. Kinderlehrer S. Y. Lin R. Cohen, R. M. Hardt and M. Luskin. Minimum energy configurations for liquid crystals: Computational results. In *Theory and Applications of Liquid Crystals*, pages 99–121. IMA Volumes in Mathematics and its Applications, Springer-Verlag, New York, 1987. J. L. Ericksen and D. Kinderlehrer.

[172] Elisabeth Rouy and Agnès Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 29(3):867–884, 1992.

[173] S. Rusinkiewicz and M. Levoy. Qsplat: a multiresolution point rendering system for large meshes. In *Proc. SIGGRAPH '00*, pages 343–352, 2000.

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

[175] Takashi Sakai. *Riemannian geometry*, volume 149 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, 1996.

[176] H. Von Schelling. Coupon collecting for unequal probabilities. *Amer. Math. Monthly*, 61:306–311, 1954.

[177] E. Schwartz, A. Shaw, and E. Wolfson. A numerical solution to the generalized mapmaker's problem: Flattening nonconvex polyhedral surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(9):1005–1008, 1989.

[178] J. Sethian. Fast marching level set methods for three-dimensional photolithography development. In *Proc. SPIE International Symposium on Microlithography*, 1996.

[179] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci. U.S.A.*, 93(4):1591–1595, 1996.

[180] J. A. Sethian. *Level set methods*, volume 3 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, 1996.

[181] Claude E. Shannon. Coding theorems for a discrete source with a fidelity criterion. In *Information and decision processes*, pages 93–126. McGraw-Hill, New York, 1960.

[182] L. A. Shepp. Covering the circle with random arcs. *Israel J. Math.*, 11:328–345, 1972.

[183] L. Simon. *Lectures on Geometric Measure Theory.*

[184] Herbert Solomon. *Geometric probability.* Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1978.

[185] M. Struwe. *Variational Methods.*

[186] M. Struwe. On the evolution of harmonic mappings of riemannian surfaces. *Comment. Math. Helvetici*, (60):558–581, 1985.

[187] K.T. Sturm. Generalized ricci curvature bounds and convergence of metric measure spaces. *C. R. Acad. Sci. Paris, Ser. I, to appear.*, 2004. `http://www-wt.iam.uni-bonn.de/ sturm/publikationen/paper48.pdf`.

[188] 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):115–1138, 1991.

[189] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, pages 2319–2323, 2000.

[190] P. Teo, G. Sapiro, and B. Wandell. Creating connected representations of cortical gray matter for functional mri visualization. *IEEE Trans. Medical Imaging*, 16(6):852–863, 1997.

[191] J. W. Thomas. *Numerical partial differential equations: finite difference methods*, volume 22 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1995.

[192] P. M. Thompson, J. N. Giedd, R. P. Woods, D. MacDonald, A. C. Evans, , and A. W. Toga. Growth patterns in the developing brain detected by using continuum-mechanical tensor maps. *Nature*, 404:190–193, 2000.

[193] Paul Thompson. `www.loni.ucla.edu/∼ thompson/thompson.html`.

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

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

[196] D. Toledo. Book review: Geometric group theory vol 2: Asymptotic invariants of infinite groups. *Bull. Amer. Math. Soc.*, 33.

[197] Yen-Hsi Richard Tsai, Li-Tien Cheng, Stanley Osher, and Hong-Kai Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM J. Numer. Anal.*, 41(2):673–694 (electronic), 2003.

[198] D. Tschumperle and R. Deriche. Regularization of orthonormal vector sets using coupled pdes. In *IEEE Workshop on Variational and Level-Set Methods, Vancouver, Canada.*, 2001.

[199] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, (40):1528–1538, 1995.

[200] D.C. Van Essen, H. Drury, Joshi S., and M.I. Miller. Functional and structural mapping of human cerebral cortex: Solutions are in the surfaces. In *Proceedings of the National Academy of Science*, pages 788–795, 1998.

[201] Luminita A. Vese and Stanley J. Osher. Numerical methods for p-harmonic flows and applications to image processing. *SIAM J. Numer. Anal.*, 40(6):2085–2104 (electronic) (2003), 2002.

[202] *Visualization Toolkit*. `http://www.kitware.com/VTK`.

[203] B. Wandell, S. Chial, and B. Backus. Visualization of cortical measurements. *Journal of Cognitive Neuroscience*, 12(5):739–752, 2000.

[204] Y.L Wang, X. Gu, T. Chan, P.M. Thompson, and S.T. Yau. Intrinsic brain surface conformal mapping using a variational method. In *Proc. Medical Imaging Computing and Computer Assisted Intervention (MICCAI)*, Canada, November 2003.

[205] ISOMAP website.

[206] Joachim Weickert. *Anisotropic diffusion in image processing*. European Consortium for Mathematics in Industry. B. G. Teubner, Stuttgart, 1998.

[207] E. W. Weisstein. `http://www.mathworld.wolfram.com/HyperspherePointPicking.html`.

[208] A. Witkin and P. Heckber. Using particles to sample and control implicit surfaces. In *Computer Graphics (SIGGRAPH)*, pages 269–278. SIGGRAPH, 1994.

[209] F. E. Wolter. *Cut loci in bordered and unbordered Riemannian manifolds*. PhD thesis, Technische Universität Berlin, 1985.

[210] Z Wood, M Desburn, P. Schröder, and D. Breen. Semi-regular mesh extraction from volume. In *IEEE Visualization*, 2000.

[211] K. J. Worsley, M. Andermann, T. Koulis, D. MacDonald, and A. C. Evans. Detecting changes in nonisotropic images. *Human Brain Mapping*, 8(2-3):98–101, 1999.

[212] K. J. Worsley, S. Marrett, P. Neelin, and A. C. Evans. Searching scale space for activation in pet images. *Human Brain Mapping*, 4:74–90, 1996.

[213] M.C. Hong Y. Chen and N. Hungerbuhler. Heat flow of p-harmonic maps with values into sphere. *Math. Z.*, (205):25–35, 1994.

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

[215] G. Yngve and G. Turk. Creating smooth implicit surfaces from polygonal meshes. Technical report, Graphics, Visualization, and Usability Center. Georgia Institute of Technology, 1999.

[216] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *Proc. CVPR '99*. CVPR, 1999.

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

[218] M. Zwicker, M. Pauly, M. Knoll, and M. Gross. Pointshop3d: An interactive system for point-based surface editing. *Proc. SIGGRAPH '02*, pages 322–329, 2002.