2015

Constructing Desirable Scalar Fields for Morse Analysis on Meshes

Mustafa Hajij
Louisiana State University and Agricultural and Mechanical College

CONSTRUCTING DESIRABLE SCALAR FIELDS FOR MORSE ANALYSIS ON MESHES

A Thesis

Submitted to the Graduate Faculty of the
Louisiana State University and
Agricultural and Mechanical College
in partial fulfillment of the
requirements for the degree of
Master of Science

in

The School of Electrical Engineering and Computer Sciences

by
Mustafa Hajij
B.S., Damascus University, 2004
M.S., Jordan University for Science and Technology, 2008
August 2015

Acknowledgments

I would like to express my gratitude to Xin Li for his advice and patience. I am grateful to my teachers in the Computer Science department Konstantin Busch and Bijaya Karki for their support, many discussions, and insights. I also would like to thank Wuyi Yi for many helpful and valuable discussions. Finally, I would like to thank my family and friends for their encouragement and support.

Table of Contents

List of Figures

Abstract

Morse theory is a powerful mathematical tool that uses the local differential properties of a manifold to make conclusions about global topological aspects of the manifold. Morse theory has been proven to be a very useful tool in computer graphics, geometric data processing and understanding. This work is divided into two parts. The first part is concerned with constructing geometry and symmetry aware scalar functions on a triangulated 2-manifold. To effectively apply Morse theory to discrete manifolds, one needs to design scalar functions on them with certain properties such as respecting the symmetry and the geometry of the surface and having the critical points of the scalar function coincide with feature or symmetry points on the surface. In this work, we study multiple methods that were suggested in the literature to construct such functions such as isometry invariant scalar functions, Poisson fields and discrete conformal factors. We suggest multiple refinements to each family of these functions and we propose new methods to construct geometry and symmetry-aware scalar functions using mainly the theory of the Laplace-Beltrami operator. Our proposed methods are general and can be applied in many areas such as parametrization, shape analysis, symmetry detection and segmentation. In the second part of the thesis we utilize Morse theory to give topologically consistent segmentation algorithms.

Chapter 1
Introduction

Morse Theory is a tool in differential topology that is concerned with the relations between the geometric and topological aspects of manifolds and the real-valued functions defined on them. One of the primary interest in this theory is the relationship between the topology of a smooth manifold M and the critical points of a real-valued smooth function f defined on M. Intuitively, Morse theory studies the topological changes of the level sets of a real-valued smooth function as the height of f varies. In computer graphics and geometric processing Morse theory has found applications in global surface parameterization [33], finding a fundamental domain of a surface [63], surface quadrangulation [22], topological matching [36], implicit surfaces [85], surface segmentation [96], spline construction [92] and many other applications.

This thesis is divided into two parts. In the first part we study multiple methods that were suggested in the literature to construct geometry and symmetry-aware scalar functions on triangulated surfaces such as isometry invariant scalar functions, Poisson fields and discrete conformal factors. We suggest multiple refinements to each family of these functions and we propose new methods to construct geometry and symmetry-aware scalar functions using mainly the theory of Laplace-Beltrami operator.

In the second part we utilize Morse theory to propose two algorithms to decompose a surface into topologically consistent pieces. More precisely, we give two

algorithms that segment any surface with negative Euler characteristic into a collection of topological pair of pants. Such a decomposition is usually called pants decomposition.

In chapter 2 we give the necessary background that we need in this thesis. In particular, we introduce Morse theory on smooth surfaces and show how this theory can be extended to triangulated 2-manifolds.

The theory of Laplace-Beltrami operator has found many applications in computer graphics. The applications include mesh segmentation [42, 71], differential coordinates [83], biharmonic distance [53], intrinsic symmetry detection [93, 44, 93], correspondence [38], shape processing and analysis [48, 50], and parameterization [32]. In chapter 3 we review the theory of Laplace-Beltrami operator on Riemannian surfaces and show how it can be applied to reduce undesirable critical points from a scalar function. We also show how the heat equation can be utilized as a computationally less expensive approach to smooth a scalar function than using the eigenfunctions of the Laplacian.

Many problems in computer graphics require solving certain scalar function on a mesh. In fact, many applications of Morse theory depend on the distribution of critical points on the mesh such as [96, 23, 22]. Multiple attempts in the literature have been directed to solve the problem of undesirable critical points includeinclude persistence-based simplfication of critical points [26, 27], using harmonic maps as fair Morse functions [63], salient smoothing of a Morse function [54], and combinatorially removing unwanted critical points of a scalar function [89, 88]. However, most applications also require the Morse scalar function to be symmet-

ric and/or geometry-aware which is not addressed simultaneously in the literature along with the critical point distribution of the scalar field on the surface. In Chapter 4, we review the existing methods in the literature to construct geometry and/or symmetry aware scalar functions and we give new constructions for such functions. Furthermore, we give a detailed analysis for each construction to keep the configuration of the set of critical points optimal. These scalar functions can be applied in many areas in geometric processing such as parametrization, shape analysis, symmetry detection and segmentation. In particular a Morse scalar function with desirable geometric and symmetric properties is a fundamental task for any algorithm that uses Morse theory. Our work utilizes heavily the theory of the Laplace-Beltrami operator to design a scalar function on mesh:

1. One can use the eigenfunctions of the Laplacian to obtain scalar functions that are isometry invariant. This means that they have *intrinsic symmetry* [44, 65] with respect to the surface as well as the distribution of their critical points on the surface. We give new families of such symmetric functions and we show how to make existing symmetric fields in the literature more robust with respect to the configuration of their critical points on the manifold. In particular we also introduce the *Modified Heat Kernel Signature* and the *Modified Auto Diffusion Function*, which are parameterized families of isometry invariant scalar functions that generalize Heat Kernel Signature [86] and Auto Diffusion Function [31] respectively.
2. We use the biharmonic [53] distance to study the average geodesic distance, or AGD, function on the manifold [36]. The AGD function has been proven to have intrinsic symmetry [44] but it has only been studied in the context of Dijkstra's algorithm geodesic distance function which has many drawbacks. We show how that by using the nice properties of the biharmonic one can

gain many desirable properties that were not available with the Dijkstra's algorithm geodesic distance function.

3. Poisson fields on meshes [23] have been used to obtain a geometry aware scalar functions. We show that the discrete conformal factor [8, 9] can be used for the same purpose and propose a robust modification for the discrete conformal factor solution suggested in [8]. The proposed method make the discrete conformal factor more suitable for applications such as parametrization, segmentation and Morse theory analysis. Furthermore, we also demonstrate that the heat kernel map suggested in [64] can be used for similar purposes.

Mesh segmentation is a fundamental problem in geometric modeling. One often needs to divide a complicated mesh geometry into multiple pieces that are easier to process. Mesh segmentation has applications including texture mapping [49], collision detection [52], skeletonization [12] and three-dimensional shape retrieval [102].

Many segmentation algorithms have been proposed in the graphics literature includes [55, 78, 51, 14, 82, 2]. In chapter 5 we propose two algorithms to decompose a surface into topologically consistent pieces using Morse theory. More precisely, we give algorithms that segment the the surface into a collection of pants which are topological sphere with three disks removed. This decomposition is called pants decomposition. See Figure for an example of pants decomposition of a genus 2 surface 1.1. Using Morse theory makes our algorithm robust with respect to these two points : we obtain a decomposition that is independent of the embedding and the computation of the curves of the decomposition is a task that can be computed in linear time.

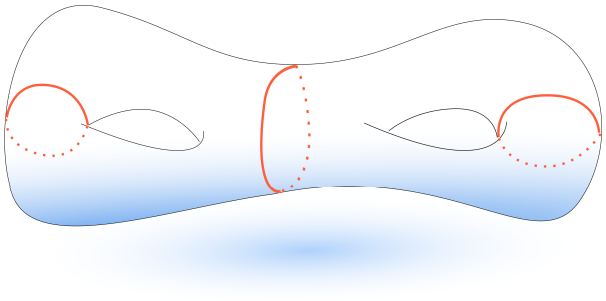
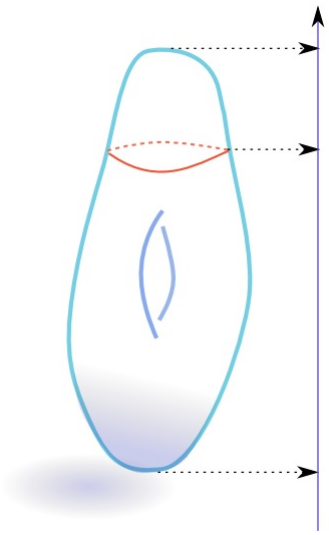


FIGURE 1.1. Pants decomposition of genus 2 surface

Similar algorithms to ours have been proposed by [51] and [100]. However, all the existing methods rely on computing certain loops on the surface called *Handles and Tunnels loops* and computing such curves is computationally expensive and more importantly it depends on the embedding of the surface.

Chapter 2
Preliminaries and Background



2.1 Introduction

Morse Theory is a tool in differential topology that is concerned with the relations between the geometric and topological aspects of manifolds and the real-valued functions defined on them. One of the primary interest in this theory is the relationship between the topology of a smooth manifold M and the critical points of a real-valued smooth function f defined on M. Intuitively, Morse theory studies the topological changes of the level sets of a real-valued smooth function as the height of f varies. Morse theory was first introduced by Morse [60] for infinite dimensional spaces. A comprehensive introduction to Morse theory on finite-dimensional manifolds is given [58]. See also [56, 7]. Morse theory was introduced on triangulated surfaced by Banchoff [6]. In this chapter we review the basics of Morse theory on smooth surfaces and then we review the basics of Morse theory on triangulated

surfaces. For a more thorough treatment of the subject the reader is refereed to the references mentioned above.

2.2 Previous Work

2.2.1 Morse Theory on Meshes

Morse theory has been proven to be a very useful tool in computer graphics, geometric data processing and understanding. Morse theory has been extended to triangulated 2-manifolds by [6]. Recently, Morse theory has found applications in global surface parameterization [33], finding a fundamental domain of a surface [63], surface quadrangulation [22], topological matching [36], implicit surfaces [85], surface segmentation [96], spline construction [92] and many other applications. Any algorithm that utilities Morse theory requires a careful design of the so-called Morse function. Most of the work in this thesis is aimed at this particular point.

2.2.2 Designing Scalar Functions on Meshes

In every application of Morse theory one needs to select a Morse scalar function with an appropriate configuration of critical points. In fact many applications of Morse theory depends on distribution of the critical points on the mesh such as [96, 23, 22]. Multiple attempts in the literature have been directed to solve the problem of undesirable critical points include persistence-based simplfication of critical points [26, 27], using harmonic maps as fair Morse functions [63], salient smoothing of a Morse function [54], and combinatorially removing unwanted critical points of a scalar function [89, 88]. All the existing methods however do not address the symmetry and the geometry features of the scalar function simultaneously with distribution of the critical points of the scalar function on the surface.

This work is inspired by multiple areas in computer graphics and geometric processing. The technical part of the thesis utilizes the theory of the Laplace-Beltrami

operator. This operator has many applications in computer geometric processing including mesh segmentation [42, 71], differential coordinates [83], biharmonic distance [53], intrinsic symmetry detection [93, 44, 93], correspondence [38], shape processing and analysis [48, 50], and parametrization [32]. Our work utilizes mainly the theory of the Laplace-Beltrami operator to design a scalar function on meshes:

1. One can use the eigenfunctions of the Laplacian to obtain scalar functions that are isometry invariant. This means that they have *intrinsic symmetry* [44, 65] with respect to the surface as well as the distribution of their critical points on the surface. We give new families of such symmetric functions and we show how to make existing symmetric fields in the literature more robust with respect to the configuration of their critical points on the manifold.
2. We use the biharmonic distance [53] to study the average geodesic distance function, or AGD, on the manifold [36]. The AGD function has been proven to have intrinsic symmetry [44] but it has only been studied in the context of Dijkstra's algorithm geodesic distance function which has many drawbacks. We show that by using the nice properties of the biharmonic one can gain many desirable properties that were not available with the Dijkstra's algorithm geodesic distance function.
3. Poisson fields on meshes [23] have been used to obtain a geometry aware scalar functions. We show that the discrete conformal factor [8, 9] can be used for the same purpose and propose a robust modification for the discrete conformal factor solution suggested in [8]. The proposed method make the discrete conformal factor more suitable for applications such as parametrization, segmentation and Morse theory analysis. Furthermore, we also demon-

strate that the heat kernel map suggested in [64] can be used for similar purposes.

4. Any scalar function can be written as a linear summation of the eigenfunctions of the Laplace-Beltrami operator. This gives us powerful tool to filter the undesired critical points of a scalar function. We also show how the heat equation can be utilized as a computationally less expensive approach than using the eigenfunctions of the Laplacian. This method is commonly used for mesh smoothing [87, 18].

Even though we are designing scalar functions here for applications on Morse theory, it should be emphasized that the techniques are general and can be applied in other areas in geometric processing. In our construction of scalar functions we have paid attention to the following desirable properties:

1. The isolines of the scalar function are shape-aware in the sense that they follow one of the principal directions of the surface.
2. The critical points of the scalar function coincide with feature or the symmetry points on the surface.
3. If the surface has some sort of symmetry then the scalar field also inherits the symmetry of the surface.
4. Minimal user input. In fact, our methods require no user input or an input one initial condition.

For the rest of this chapter we will give the basic definitions for Morse theory on smooth and triangulated surfaces.

2.3 Basics of Topological Manifolds

The purpose from the following two sections is to state the necessary definitions needed to talk about Morse theory. The material introduced in these sections is

standard and can be found in any introduction on topological and differentiable manifolds textbook. See for instance [47, 46]. We will assume that the reader is familiar with basic topology.

Definition 2.1. *A topological space M is said to be n-Euclidean if for every point in M there exists a neighborhood around that point that is homeomorphic to $\mathbb{R}^n$.*

Definition 2.2. *Let M be a topological space and let $U \subset M$ be an open set. A homeomophism $\phi : U \longrightarrow \mathbb{R}^n$ is called a chart coordinate of M. We denote such a chart by (U, ϕ).*

Definition 2.3. *A topological n-manifold is a topological space M that is locally n-Euclidean, Hausdorff and second countable.*

An atlas on a manifold M is a collection of coordinate charts $\{(U_\alpha, \phi_\alpha)\}$ such that $\cup_\alpha U_\alpha = M$. In this thesis we are interested mainly in 2-manifolds with or without boundary. The definition of a manifold with boundary is the following.

Definition 2.4. *A second countable Hausdorff topological space such that for every $x \in M$ there exists a neighborhood of x locally homeomophic to $\mathbb{H} = \{(x_1, ..., x_n) \in \mathbb{R}^n : x_n \geq 0\}$ is called an n-manifold with boundary.*

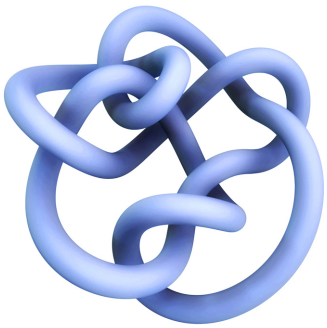


FIGURE 2.1. A surface that is homeomophic to the torus

The topological spaces that we will deal with in this thesis will be 2-manifolds, also called surfaces. Recall that the classification theorem of surfaces states that any

compact, connected, orientable and without boundary surface is homeomorphic to one of the following surfaces :

1. The sphere.
2. The connected sum of g tori, for $g \geq 1$.

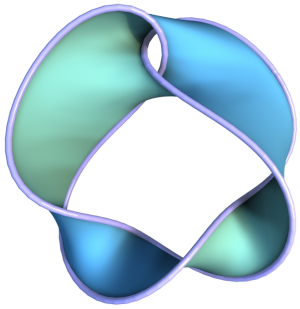


FIGURE 2.2. A surface with boundary

On the other hand, any compact, connected, orientable surface with boundary is simply a surface from the list of the classification theorem above with a finite number of holes. See Figure 2.2 for an example of a surface with one boundary component. Compact, connected and orientable surfaces with and without boundary can be classified up to homeomorphism via the Euler Characteristic. The Euler characteristic of a surface F with genus g and b boundary component is given by $\chi(F) = 2 - 2g - b$.

2.4 Basics of Differential Topology

The definition of a topological manifold does not allow us to to investigate the local properties of the space. This problem can be resolved by introducing a differential structure on the topological manifold.

Definition 2.5. *Let (U, ϕ) and (V, ψ) be two charts such that $U \cap V \neq \emptyset$. The composition map $\psi \circ \phi^{-1} : \phi(U \cap V) \longrightarrow \psi(U \cap V)$ is called a transition map . If all the transition functions are smooth then we say that the manifold is smooth (or differentiable).*

We say that an atlas on a manifold is smooth if all its charts are smooth. *A differntiable structure* on a manifold is a maximal smooth atlas. Now we are ready to define smooth scalar functions on a smooth manifold.

Definition 2.6. *A function $f : M \longrightarrow \mathbb{R}$ is smooth if $f \circ \phi : U \longrightarrow \mathbb{R}$ is smooth for every chart (U, ϕ).*

More generally, we have the following definition of smooth functions between two manifold M and N.

Definition 2.7. *A map $f : M \longrightarrow N$ is smooth if for every chart (V, ψ) on N and every chart (U, ϕ) in M, the composition $\psi \circ f \circ \phi^{-1}$ is smooth in the Euclidean sense.*

Suppose now that M is embedded in some $\mathbb{R}^n$ and let $\gamma : (-\epsilon, \epsilon) \longrightarrow M$ be a curve on the manifold M.

Definition 2.8. *We say that a vector $v \in \mathbb{R}^n$ is tangent to the manifold $M \subset \mathbb{R}^n$ at a point p if there exists a curve $\gamma : (-\epsilon, \epsilon) \longrightarrow M$ such that $\gamma(0) = p$ and $\gamma'(0) = v$.*

The set of all tangent vectors at a point $p \in M$ is called the *tangent space at p* and it is denoted by T_pM. The tangent space of an n-manifold can be shown to be a n-dimensional vector space over $\mathbb{R}$.

Definition 2.9. *Let p be a point on a n-manifold M and let $(\phi, (x_1, .., x_n))$ be a local chart around p such that $x_i(p) = 0$ for $i = 1, .., n$. Let $c_k : (-\epsilon, +\epsilon) \longrightarrow M$ be the curves on M defined by $\phi^{-1} \circ e_k$, where*

$$e_k(t) = (t\delta_{1k}, ..., t\delta_{nk}), \ \ k = 1, ..., n.$$

The vectors $\frac{\partial}{\partial x_1}|_p$,...,$\frac{\partial}{\partial x_n}|_p$ are by definition $c_1'(p)$,...,$c_2'(p)$ respectively.

One can show that the vectors $\frac{\partial}{\partial x_1}|_p,...,\frac{\partial}{\partial x_n}|_p$ form a basis for tangent space T_pM. We will also need the concept of a smooth vector field on a manifold.

Definition 2.10. *A smooth vector field on M is a function that assigns to every point p a vector $v_p \in T_pM$ such that v_p is smooth in p.*

For every point $p \in M$ any smooth function $f : M \longrightarrow N$ induces a linear map $df_p : T_pM \longrightarrow T_{f(p)}N$. This is given in the following definition.

Definition 2.11. *Suppose that $f : M \longrightarrow N$ is a smooth map and $\gamma : (-\epsilon, \epsilon) \longrightarrow M$ is a curve such that $\gamma(0) = p$ and $\gamma'(0) = v$. For every point $p \in M$ the function f induces a linear map $df_p : T_pM \longrightarrow T_{f(p)}N$, called the differential of f at p, defined by :*

$$df_p(v) = (f \circ \gamma)'(0) \in T_{f(p)}M$$

The vector $df_p(v)$ is also called the push-forward of the vector v induced by f and sometimes the notation $f_*(v)$ is used in place of $df_p(v)$.

2.5 Morse Theory on Smooth Surfaces

Let M be a compact smooth surface and let $I = [a, b] \subseteq \mathbb{R}$, where $a < b$, be a closed interval. Let $f : M \longrightarrow I$ be a smooth function defined on M. A point $x \in M$ is called a *critical point* of f if the differential df_x is zero. A value c in $\mathbb{R}$ is called a *critical value* of f is $f^{-1}(c)$ contains a critical point of f. A point in M is called a *regular point* if it is not a critical point. Similarly, if a value $c \in \mathbb{R}$ is not a critical value then we call it a regular value. The inverse function theorem implies that for every regular value c in $\mathbb{R}$ the level set $f^{-1}(c)$ is a 1-manifold, i.e., $f^{-1}(c)$ is a disjoint union of circles. A critical point is called *non-degenerate* if the matrix of the second partial derivatives of f, called the *Hessian matrix*, is non-singular. If all the critical points of f are non-degenerate and all critical points have distinct values then f is a *Morse function*. If the surface M has boundary then we will

also require two other conditions : (1) $f^{-1}(\partial I) = \partial M$ and (2) there are no critical points on ∂M. In other words, the boundary points in the interval I, the values in ∂I, are regular values for the function f. The *index* of a critical point x of f, denoted by $index_f(x)$, is defined to be the number of negative eigenvalues of its Hessian matrix. Since the Hessian of a scalar function on smooth surface is a 2×2 symmetric matrix, then the index takes the values $0, 1$ or 2.

The definition of a Morse function is motivated mainly by the following Lemma.

Lemma 2.12. *(Morse Lemma) Let M be a smooth surface, $f : M \longrightarrow \mathbb{R}$ be a smooth function and p be a non-degenerate critical point of f. We can choose a chart (ϕ, U) around p such that $f \circ \phi^{-1}$ takes exactly one of the following three forms:*

1. $f \circ \phi^{-1}(X, Y) = X^2 + Y^2 + c.$
2. $f \circ \phi^{-1}(X, Y) = -X^2 - Y^2 + c.$
3. $f \circ \phi^{-1}(X, Y) = X^2 - Y^2 + c.$

The form of a Morse function f around a critical point can be proven to be independent of the choice of the chart. Hence, we will call this form the *standard form* of the Morse function f around p. The previous Lemma says that the surface looks very nice around a non-degenerate critical point. In fact, one can use the previous theorem to define the index of a critical point p : the number of minus signs in the standard form of the function f around p. On can see that on a non-degenerate critical point of index 0 the function f takes a minimum value, on a non-degenerate critical point of index 1, the graph of the function looks like a

saddle and on a non-degenerate critical point of index 2 the function f takes a maximum value. See Figure 2.3

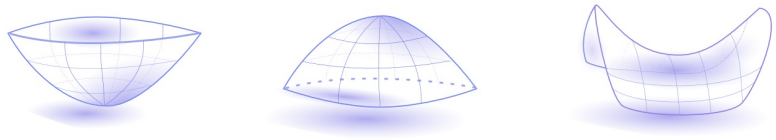


FIGURE 2.3. Minimum, Maximum and Saddle.

2.5.1 Handle Decomposition of a Surface

Morse theory study the shape of a manifold by studying the critical points of a scalar function defined on M. The following lemma is an example of what we mean.

Theorem 2.13. *Let M be a closed surface and let $f : M \longrightarrow [a, b]$ be a Morse function. If f has exactly two critical points then M is diffeomorphic to the 2-sphere.*

The previous theorem shows how our knowledge of critical points of a Morse scalar function defined on M can provide insight about the topology of the surface. In fact, Morse theory studies this observation in a systematic manner. Before giving the theorems that show this point on surfaces we need a few definitions.

Let M be a smooth surface and let $f : M \longrightarrow \mathbb{R}$ be a Morse function defined on M. Define the set :

$$M_{f,t} = \{x \in M : f(x) \leq t\}$$

Let $a < b$ be real numbers. Define

$$M_{f,[a,b]} = \{x \in M : a \leq f(x) \leq b\}$$

When it is clear from the context, we will drop f from the notation and use simply M_t and $M_{[a,b]}$ to refer to the previous two sets. Morse theory studies the topological changes on the set M_t as the value t varies.

Theorem 2.14. *Let $f : M \longrightarrow \mathbb{R}$ be a smooth function on a smooth surface M. Let $a < b$ be real numbers such that f has no critical values in the interval $[a, b]$, then the surfaces M_a and M_b are diffeomorphic.*

The previous theorem says that the topology of the surface M_t does not change as t passes through regular values. In the following we use D^1 to denote the interval $[0, 1]$. The end points of D^1 will be denoted by ∂D^1. In other words $\partial D^1 = \{0, 1\}$. Given a Morse function f on M, the following theorem gives s precise description for the change that occur in the topology of the surface M_t as t passes through a critical value.

Theorem 2.15. *Let $f : M \longrightarrow \mathbb{R}$ be Morse function. Let p be a critical point of index i and $f(p) = t$ be its corresponding critical value. Let ϵ be chosen small enough so that f has no critical values on the interval $[t - \epsilon, t + \epsilon]$. Then :*

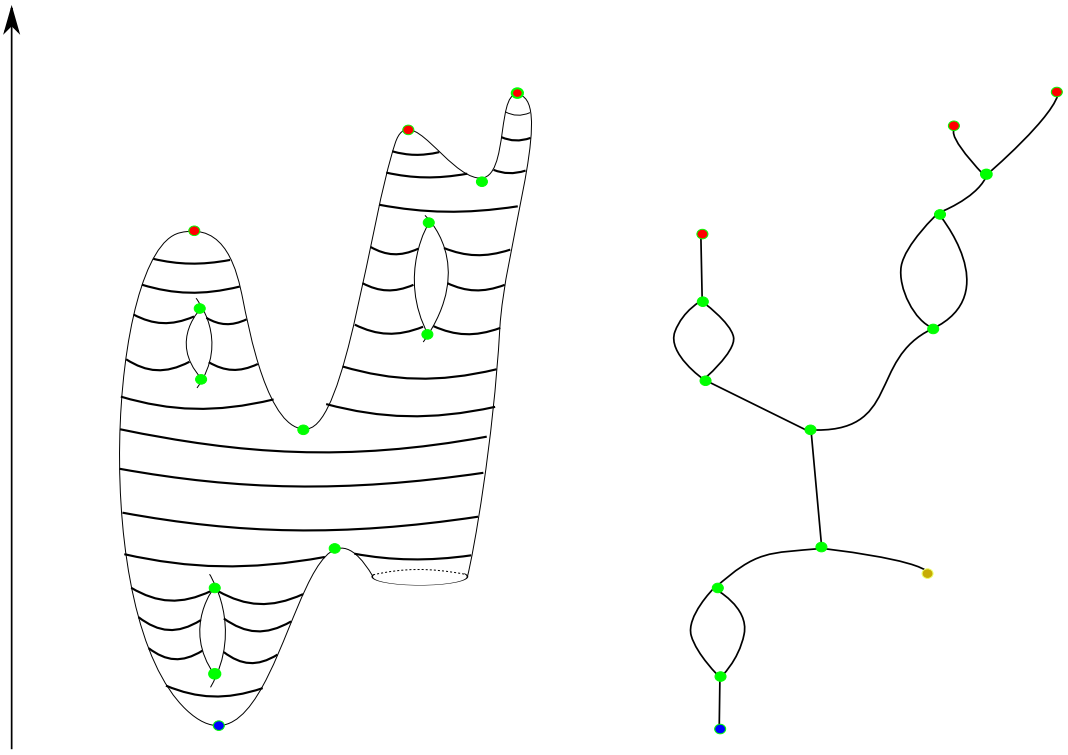
1. *If $index_f(p) = 0$, then $M_{t+\epsilon}$ is diffeomorphic to the disjoint union of $M_{t-\epsilon}$ and a 2-dimensional disk D^2.*
2. *If $index_f(p) = 1$, then $M_{t+\epsilon}$ can be obtained from $M_{t-\epsilon}$ by attaching a 1-handle. That means that $M_{t+\epsilon}$ can be obtained by gluing a rectangular strip $D^1 \times D^1$ to the boundary of $M_{t-\epsilon}$ along $D^1 \times \partial D^1$.*
3. *If $index_f(p) = 2$, then $M_{t+\epsilon}$ can be obtained by capping off the surface $M_{t-\epsilon}$ with a disk D^2. That means that $M_{t+\epsilon}$ is obtained by gluing a disk D^2 along its boundary ∂D^2 to one of the boundary components of $M_{t-\epsilon}$.*

2.5.2 Reeb Graph

Let M be a smooth orientable surface possibly with boundary ∂M and let $f : M \longrightarrow [0, 1]$ be a Morse function. Define the equivalence relation $\sim$ on M by $x \sim y$ if and only if x and y belong to the same connected component of a level set

$f^{-1}(c)$ for the same $c \in [0,1]$. The set $X/\sim$ with the standard quotient topology induced by the function $\hat{f} : R(f) \longrightarrow [0,1]$ is called the *Reeb graph* of f. Every vertex of the $R(f)$ arises from a critical point of f. Every maximum or minimum of f or a boundary component M gives rise to a degree 1-node of $R(f)$ and every saddle of f gives rise to degree 3-node. If M is an embedded surface without boundary then M can be recovered up to a homeomorphism from $R(f)$ as the boundary of an oriented 3-dimensional regular neighborhood of the graph $R(f)$.

Remark 2.16. *One reason for our choice of Morse function with boundary to satisfy $f^{-1}(\partial I) = \partial M$ is to make it consistent with the definition of Reeb graph we give here. Note that each boundary component circle maps exactly to one point on the Reeb graph.*



The definition of Reeb graph goes back to G. Reeb [70]. It was first introduced to computer graphics in [76]. Reeb graph has found applications in shape under-

standing [1], quadrangulation [35], surface understanding [11], segmentation [95], parametrization [67, 99], animation [40] and many other applications. Reeb graphs algorithms can be found in many papers such as [81, 16, 66, 24].

2.6 Triangulated 2-Manifolds

In practice, we usually approximate smooth surfaces by triangulated meshes. The combinatorial structure of a triangulated mesh is represented by a simplicial complex. Concepts from smooth differential geometry and analysis can be generalized to simplical complexes. We start this section by giving the basics of triangulated 2-manifolds as well as the basics of Morse theory on a triangulated 2-manifold.

2.6.1 Simplicial Complex

Let k be a positive integer and let $v_0, ..., v_k$ be vectors in $\mathbb{R}^n$. We say that $v_0, ..., v_k$ are in general position if $v_0 - v_1, ..., v_0 - v_k$ are linearly independent. By convention, if $k = 0$ then v_0 is in general position. Define a *$k-$dimensional simplex with vectors* $\{v_0, .., v_k\}$ to be the set:

$$\sigma^k = [v_0, ..., v_k] = \{\sum_{i=0}^{k} \lambda_i v_i \in \mathbb{R}^n : \sum_{i=0}^{k} \lambda_i = 1, \lambda \geq 0\}$$

A k-simplex can be proven to be the smallest convex set, or the convex hull, that contains the vertices $v_1, .., v_k$. Figure 2.4 shows examples of simplices of various dimensions. We usually call the vectors $v_0, ..., v_k$ of a simplex σ^k the *vertices* of the simplex σ^k. Note that a simplex is completely determined by its set of vertices. If

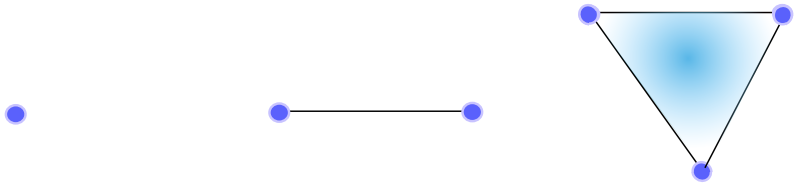


FIGURE 2.4. From left to right, 0-simplex, 1-simplex and 2-simplex

σ_1 and σ_2 are simplices and the vertices of σ_1 is a subset of the vertices of σ_2 then we say that σ_1 is a *facet* of σ_2.

Definition 2.17. *A simplical complex is a finite collection of simplices* Σ *that satisfies the following conditions:*

1. *If* σ *is in* Σ *then all the facets of* σ *are also in* Σ.
2. *If* $\sigma_1, \sigma_2 \in \Sigma$, $\sigma_1 \cap \sigma_2 \neq \phi$, *then* $\sigma_1 \cap \sigma_2$ *is in* Σ.
3. *Every point in* Σ *has a neighborhood that intersects at most finitely many simplices of* Σ.

A simplicial complex is said to be n-dimensional if it contains at least one n-dimensional simplex and no simplexes of dimension greater than n. The geometric concept of orientation of a simplex can also be defined combinatorially using vertices. This can be done by imposing an ordering on the vertices of the simplex $\sigma^k = [v_0, ..., v_k]$. Let σ^k be a simplex and let $(v_{i_0}, ..., v_{i_k})$ and $(v_{j_0}, ..., v_{j_k})$ be two ordering of the vertices of σ^k. We say that the ordering $(v_{i_0}, ..., v_{i_k})$ is equivalent to $(v_{j_0}, ..., v_{j_k})$ if there a permutation s on the set $\{0, ..., k\}$ such that, i.e., $s(i_n) = j_n$ for $n = 0, ..., k$, and s is a product of even number of transpositions (a permutation that swaps only two elements and leave the other elements unchanged). One can use this definition to define an equivalence relation on the set of all ordering of a simplex. This equivalence relation partitions this set into two equivalence classes. An orientation of a simplex σ^k is a choice of one of these equivalence classes. An *oriented* k*-simplex* is a simplex σ^k with an orientation. Note that a 0-simplex has a unique orientation. This combinatorial definition matches our geometric intuitive notion of the two possible orientation of simplex.

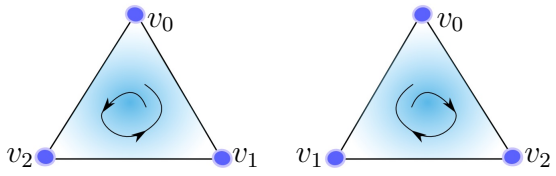


FIGURE 2.5. Possible orientations : (v_0, v_2, v_1) (left) and (v_0, v_1, v_2) (right).

An oriented k-simplex induces orientation on its $(k-1)$ in a natural way. See Figure 2.6 for an example of the induced orientation of the face $[v_0, v_1, v_2]$ on its edges. Care must be taken when one defines orientation on a simplicial complex

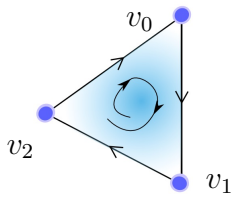


FIGURE 2.6. An oriented 2-simplex with its induced orientation on the edges.

Σ. Suppose that σ_1 and σ_2 are two adjacent oriented simplexes in a simplicial complex Σ. We say that the two orientations of σ_1 and σ_2 are *compatible* if they induce opposite orientations on their common facet. See Figure2.7 for an example of compatible orientation. A choice of orientation on each simplex of Σ such that the orientations of any two adjacent simplexes in Σ are compatible is called compatible orientation of the simplicial complex Σ. A simplicial complex with a compatible orientation is called an oriented simplicial complex. Such an orientation may or may not exist. If it exists on a simplicial complex then we call such a complex *orientable* otherwise it is called *nonorientable*. In our work here we will mainly work with oriented simplicial complexes.

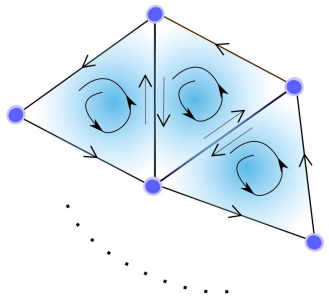


FIGURE 2.7. Compatible orientation

Suppose that Σ is a 2-dimensional simplicial complex. We will denote by $|\Sigma|$ to the set obtained by taking the union of all simplexes of Σ and equipped with the

relative subspace topology of the usual topology of $\mathbb{R}^3$. We are mainly interested in the case when $|\Sigma|$ is a 2-manifold. For this reason we have the following definition.

Definition 2.18. *A triangulation of a surface S is 2-dimensional simplicial complex Σ with a homeomorphism $f : S \longrightarrow |\Sigma|$.*

By a *triangulated 2-manifold*, or triangulated surface, we mean a topological 2-manifold equipped with a triangulation. It is known that every surface can be triangulated [46].

We will often call 0-simplex a vertex, 1-simplex an edge and 2-simplex a face. If Σ is a 2-dimensional simplicial complex then we will denote by $\mathcal{V}(\Sigma)$, $\mathcal{E}(\Sigma)$ and $\mathcal{F}(\Sigma)$ to the set of vertices, edges and faces of Σ respectively. When it is clear from the context we will also drop Σ from the notation.

2.6.2 Simplicial Maps

Suppose that Σ_1 and Σ_2 are simplicial complexes. We will often be interested in specifying a *piece-wise linear continuous* map between $|\Sigma_1|$ and $|\Sigma_2|$. Such maps can be induced by simplicial maps as in the following definition.

Definition 2.19. *Let Σ_1 and Σ_2 be two simplical complexes. A simplicial map between Σ_1 and Σ_2 is a map $f : \mathcal{V}(\Sigma_1) \longrightarrow \mathcal{V}(\Sigma_2)$ such that if $v_1, ..., v_k$ are vertices of a simplex in Σ_1 then $f(v_1), ..., f(v_k)$ are vertices of a simplex in Σ_2.*

If $f : \mathcal{V}(\Sigma_1) \longrightarrow \mathcal{V}(\Sigma_2)$ is a simplicial map and $\sigma^k = [v_0, ..., v_k]$ is a simplex in Σ_1 then the map f can be extended uniquely to a continuous map $\hat{f}$ between $|\sigma^k|$ and $|conv(f(v_0), ..., f(v_k))|$ (the smallest convex set containing $f(v_0), ..., f(v_k)$) as follows :

$$\hat{f}(\sum_{i=0}^{k} \lambda_i v_i) = \sum_{i=0}^{k} \lambda_i f(v_i)$$

where $\sum_{i=0}^{k} \lambda_i = 1, \lambda_i \geq 0$. Extending f in this manner on every $\sigma \in \Sigma_1$ we obtain a piece-wise linear continuous map between $|\Sigma_1|$ and $|\Sigma_2|$. In other words, we define f on the vertices of M and in order to obtain $\hat{f}$ we extend f linearly on the edges and the triangles of M.

Remark 2.20. *We will often abuse the notation between the function f and its induced function $\hat{f}$.*

Remark 2.21. *The previous definition of simplicial map works in general between any two simplicial complex Σ_1 and Σ_2. However, it is useful sometimes to think about a simplicial map $f : \Sigma \longrightarrow \mathbb{R}$ on a surface Σ in the following way. Let $B_i : |\Sigma| \longrightarrow \mathbb{R}$ be the hat function defined by $B_i(v_j) = \delta_{ij}$ and extended linearly to the entire simplicial complex $|\Sigma|$, then for every $x \in \Sigma$ one can write:*

$$\hat{f}(x) = \sum_{i=0}^{k} f(v_i) B_i(x),$$

where $\sum_{i=0}^{k} B_i(x) = 1$ and $B_i(x) \geq 0$.

This function is called the hat function because if one defines B_i on a face $[v_i, v_j, v_k]$ then the graph of the function of B_i looks like Figure 2.8.

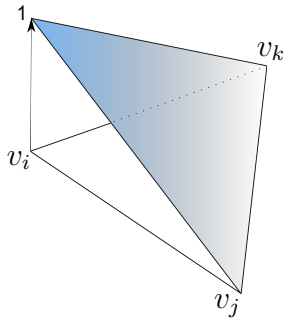


FIGURE 2.8. The graph of the i^{th} hat function on a face $[v_i, v_j, v_k]$.

2.6.3 Morse Theory on Triangulated 2-Manifolds

Morse theory was extended to triangulated manifolds by Banchoff [6]. Let M be a triangulated 2-manifold and let $f : M \longrightarrow \mathbb{R}$ be piece-wise linear continuous

function on M. The link $Lk(v)$ of a vertex v is defined as the set of all vertices w such that v shares an edge $[v, w]$ with w. The upper link of v is defined as

$$Lk^+ = \{u \in Lk(v) : f(u) > f(v)\},$$

and the lower link is defined by

$$Lk^- = \{u \in Lk(v) : f(u) < f(v)\},$$

and mixed link

$$Lk^{\pm}(v) = \{(u_1, u_2) : f(u_1) < f(v) < f(u_2)\}.$$

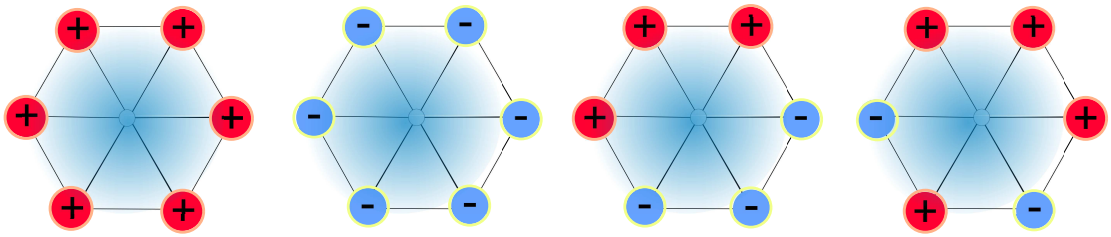


FIGURE 2.9. (a) minimum, (b) maximum, (c) regular vertex, (d) saddle .

Using the link definitions we can classify vertices of M as follows. A vertex v in M is *PL regular* if the cardinality $|Lk^{\pm}(v)|$ of $Lk^{\pm}(v)$ is equal to 2. If $|Lk^+(v)| = 0$ then v is a *PL maximum* vertex with index 1 and if $|Lk^-(v)| = 0$ then v is a *PL minimum* with index 0. If $|Lk^{\pm(v)}| = 2 + 2m$ then v is a PL saddle with index 1 and multiplicity $m \geq 1$. See Figure 2.9. A PL function is *PL Morse function* if each vertex is either PL regular or simple PL critical and the function values of the vertices are distinct.

The Euler characteristic of a closed triangulated surface M can be computed via a PL Morse function f as follows:

$$\chi(M) = \sum_{x \text{ critical}} (-1)^{index_f(x)}.$$

Chapter 3
Manifold Harmonics

3.1 Introduction

Given a Riemannian surface M, one can define the so called Laplace-Beltrami operator, or the Laplacian, on the Hilbert space $\mathcal{L}(M)$ of all smooth real valued functions on M, or smooth 0-forms. The eigenfunctions of the Laplace-Beltrami operator form a complete set of orthonormal basis of the space $\mathcal{L}(M)$. In other words, any scalar function on M can be decomposed as a linear summation of the eigenfunction, or the harmonics, of the Laplace-Beltrami operator. Hence, the Laplacian on a Riemanian manifold provides us with the classical tools of Fourier analysis for our study of scalar functions on Riemannian manifold.

The eigenfunctions of the Laplacian have found many applications in geometric processing. This includes mesh segmentation [42, 71], differential coordinates [83], biharmonic distance [53], intrinsic symmetry detection [93, 44, 93], correspondence [38], shape processing and analysis [48, 50], and parametrization [32]. In

this chapter we will introduce the Laplace-Beltrami operator on a smooth surface and then study multiple discretizations that were proposed in the literature for this operator on triangulated 2-manifolds. In fact, it was shown that none of discretiztions can capture simultaneously all desirable properties of the smooth Laplace-Beltrami operator [94]. For our purpose here, we are interested in writing any scalar function on the mesh in terms of the eigenfunction of the proposed discrete Laplacian, hence we choose discretiztions that serve this specific purpose.

Solving the eigenfunctions and the eigenvalues of a general Laplace-Beltrami operator on a general smooth manifold is a difficult task. In the discrete setting on a triangulated 2-manifolds however, we will see that the problem can be reduced to a generalized eigenvalue problem.

3.2 Quick Review of Riemannian Surfaces

In this section we give the necessary background to define the Laplace-Beltrami operator on a smooth surface.

Remark 3.1. *Our review here is not comprehensive nor general. We merely introduce the definitions and concepts on smooth surfaces that are needed for our purpose. For more details the reader is refereed to [74, 10]*

Remark 3.2. *It is usually desirable to give intrinsic description for objects on a manifold. Even though it is possible to give intrinsic definitions for all concepts that we will introduce here such as smooth forms, the divergence, the gradient, and the Laplace-Beltrami operators without the need of using charts, but for the sake of introducing less machinery we choose to give chart-based definitions.*

3.2.1 Differential Forms

Definition 3.3. *Let M be a differentiable 2-manifold and let p be a point on M. A differential 1-form on p is an element in the dual space T_p^*M of the tangent space of the manifold M at the point p. In other words $\omega_p : T_pM \longrightarrow \mathbb{R}$ is a linear function.*

A differential 1-form on M is a function that assigns to every p in M a differential 1-form on p such that $\{\omega_p | p \in M\}$ is smooth in p. In local coordinates, suppose that (U, x_1, x_2) is a local chart on M then a differential 1-form on U can represented locally by :

$$w = a_1 dx_1 + a_2 dx_1$$

where a_1 and a_2 are smooth functions on M and dx_i for $i = 1, 2$ are the differential 1-forms on U defined by

$$dx_i(p)(\frac{\partial}{\partial x_j}\Big|_p) = \delta_{ij}, \qquad i, j = 1, 2.$$

Differential 2-forms are defined similarly.

Definition 3.4. *Let M be a differentiable 2-manifold and p be a point on M. A differential 2-form on p is bilinear map $w_p : T_pM \times T_pM \longrightarrow \mathbb{R}$ such that $\omega_p(v_1, v_2) = -w_p(v_2, v_1)$ for every $v_1, v_2 \in T_pM$*

Similar to differential 1-forms, a differential 2-form on M is a function that assigns to every p in M a differential 2-form on p such that $\{\omega_p | p \in M\}$ is smooth in p. In local coordinates, if (U, x_1, x_2) is a local chart on M then a differential 2-form on U is represented locally on U by:

$$w = f dx_1 \wedge dx_2$$

where f is a smooth function on U and $dx_1 \wedge dx_2$ is a differential 2-form on defined by:

$$(dx_1 \wedge dx_2)(p)(v_1, v_2) := dx_1|_p(v_1)dx_2|_p(v_2) - dx_1|_p(v_2)dx_2|_p(v_1).$$

for every $v_1, v_2 \in T_p^*M$ and every p in U. Finally, a differential 0-form on a surface M is simply a smooth function $f : M \longrightarrow \mathbb{R}$. The set of differential i-forms, $i = 0, 1, 2$ on a surface M form a vector space where the vector space operations are induced from the vector space operations defined on point-wise differential forms. We will denote by $\Omega^i(M)$ to the vector space of all differential i-forms on M for $i = 0, 1, 2$.

3.2.2 Integration on Smooth Surfaces

Differential forms are the mathematical objects that can be integrated over a manifold. In order to integrate over a surface, the surface must be oriented. For this reason, we will always assume that this is the case. Moreover, one can only integrate 2-forms on a surface. The definition of the integration of 2-forms on a surface relies on the familiar Riemannian integration on $\mathbb{R}^2$. In fact, differential 2-forms on $\mathbb{R}^2$ can be identified with scalar functions on $\mathbb{R}^2$. One can choose a coordinate system x_1 and x_2 for $\mathbb{R}^2$ and the identification simply goes by sending a function f on $\mathbb{R}^2$ to a 2-form $f dx_1 \wedge dx_2$ on $\mathbb{R}^2$. Using this identification, we can use Riemannian integration to integrate 2-forms on $\mathbb{R}^2$.

Definition 3.5. *Let U be an open set in $\mathbb{R}^2$ with the standard coordinate x_1 and x_2 and let $\omega = f dx_1 \wedge dx_2$ be a smooth 2-from on U then*

$$\int_U \omega = \int_U f dx_1 \wedge dx_2 = \int_U f dx_1 dx_2$$

where the integration on the right hand side of the previous equation is simply the Riemannain integration of the function f on the set U.

To integrate 2-forms on smooth surface we need the following definition first.

Definition 3.6. *Let M be a smooth m-manifold and let N be smooth n-manifold. Let $F : N \longrightarrow M$ be a smooth map. The pullback $F^*\omega$ of a k-form ω on M is a k-form on N defined as follows:*

$$(F^*\omega)_p(v_1, ..., v_k) = \omega_{F(p)}(F_*(v_1), ..., F_*(v_k)),\ p \in N,$$

where $v_1, .., v_k \in T_pN$ and $F_*(v_i) \in T_{F(p)}M$ is the pushforward of $v_i \in T_pN$.

Now we define the integration of 2-forms on a surface M. Let $\phi : U \longrightarrow \phi(U) \subset \mathbb{R}^2$ be a chart on M. Let w be a smooth 2-form on U, the the integration of w on U is defined as :

$$\int_U \omega := \int_{\phi(U)} (\phi^{-1})^*\omega$$

One can prove that this definition is independent of the choice of chart (ϕ, U). Finally, to integrate a 2-form ω defined on a compact surface M. We cover the entire surface by a collection of charts $\{U_i, \phi_i\}_{i\in I}$ and since the surface is compact we can choose this set to be finite. Moreover, we choose a set of non-negative functions $\{\rho_i : M \longrightarrow \mathbb{R}\}_{i\in I}$ such that (1) $\sum_i \rho_i(p) = 1$ for all p in M and (2) $supp(\rho_i) \subset U_i$. Then, the smooth $2-$from w_i defined by $w_i := \rho_i\omega$ can be integrated on U_i as before and we can set the integration of ω as

$$\int_M \omega := \sum_i \int_{U_i} \omega_i.$$

The set of functions $\{\rho_i\}_{i\in I}$ satisfying the conditions (1) and (2) is usually called a *partition of unity*. One needs to show that this definition is independent of the choice of charts and the chosen partition of unity.

3.2.3 The Hilbert Space of 0-Forms on a Riemannian Surface

A Riemannian manifold is a differentiable manifold such that the tangent space at every point on the manifold is equipped with an inner product. This allows us to talk meaningfully about concepts like length, volume and curvature on a manifold. More precisely, we have the following definition.

Definition 3.7. *A Riemannnian manifold is a differentiable manifold such that for each point p in M the tangent space T_pM is equipped with a symmetric positive-definite bilinear form*

$$g_p : T_pM \times T_pM \longrightarrow \mathbb{R}$$

such that g_p is smooth in p. We say that $g = \{g_p : p \in M\}$ is Riemannian metric on the manifold M.

Remark 3.8. *Let V be a vector space. Recall that a symmetric bilinear map $F : V \times V \longrightarrow \mathbb{R}$ is called positive definite if it satisfies : (1) $F(v, v) \geq 0$ for all $v \in V$ and (2) $F(v, v) = 0$ implies $v = 0$.*

We are now ready to define an inner product on differential 0-forms on a smooth surface. Let (M, g) be a compact oriented Riemannian surface. The area form dM on M is a 2-form defined by

$$dM_g = \sqrt{|det(g)|}dx_1 \wedge dx_2.$$

with respect to a positively oriented chart. Here $det(g) = det(g_{ij})$. One can prove that this definition is independent of the chosen chart. For any open set $U \subset M$ the integration $\int_U dM$ is defined to be the area of U. In particular, $\int_M dM_g$ is defined to be the area of M. Let f and g be differential 0-forms on a surface M. Define:

$$\langle f, g \rangle = \int_M fg \; dM_g. \tag{3.1}$$

One can show that $\langle , \rangle$ defines a symmetric positive-definite bilinear form on $\Omega^0(M)$ and hence an inner product.

Definition 3.9. *The Hilbert space $\mathcal{L}(M, g)$ is by definition the completion of the vector space $\Omega^0(M)$ with respect to the inner product defined in 3.1.*

When the choice of the metric g is clear we will denote to $\mathcal{L}(M, g)$ by $\mathcal{L}(M)$ and denote the volume form dM_g by dM.

3.2.4 The Gradient and The Divergence Operators on a Riemannian Surface

Given a Riemannian manifold (M, g) one can generalize the familiar definitions of the gradient and divergence operators in the Euclidean space on the Riemannian manifold M.

Definition 3.10. *Let (M, g) be a Riemannian manifold and let $f : M \longrightarrow \mathbb{R}$ be a smooth function on M. The gradient of f is a vector field on M defined locally on a chart (U, x_1, x_2) by:*

$$\nabla_g f = \text{grad} f = \sum_{i,j=1}^{2} g_{ij} \frac{\partial f}{\partial x_i} \frac{\partial}{\partial x_j} \tag{3.2}$$

This definition can be proven to be independent of the chosen chart U.

Definition 3.11. *Let (M, g) be a Riemannian manifold and let X be a smooth vector field on M. Let (U, x_1, x_2) be a chart on M and write $X = \sum_{i=1}^{2} b_i \frac{\partial}{\partial x_i}$ for some smooth functions b_i on U. The divergence of X is a scalar function on M defined locally on U by :*

$$\text{div}_g X = \frac{1}{\sqrt{|det(g)|}} \sum_{i,j=1}^{2} \frac{\partial}{\partial x_i} (b_i \sqrt{|det(g)|}) \tag{3.3}$$

Again the definition of the divergence operator is independent of the choice of the chart U.

3.2.5 The Laplace-Beltrami Operator on a Riemannian Surface

Let (M, g) be a compact Riemannian surface. Let (U, x_1, x_2) be a chart for M. The Laplace-Beltrami operator $\Delta : \mathcal{L}(M) \longrightarrow \mathcal{L}(M)$ is defined on the chart U by :

$$\Delta = \frac{1}{\sqrt{|det(g)|}} \sum_{i,j=1}^{n} \frac{\partial}{\partial x_i} (g_{ij} \sqrt{|det(g)|} \frac{\partial}{\partial x_i}).$$

On a Riemannian manifold (M, g) the Laplace-Beltrami operator can be characterized by the property that a diffeomorphism is an isometry if and only it preserves

the Laplace-Beltrami operator Δ. This implies that the Laplace-Beltrami operator determines the Riemannian structure of the manifold.

The Laplace-Beltrami operator satisfies $\langle \Delta f, g \rangle = \langle f, \Delta g \rangle$ which means that it is a self-adjoint operator. This implies that all its eigenvalues are real and $\mathcal{L}(M)$ admits a complete orthonormal basis of eigenfunctions for the operator Δ . Moreover, the Laplace-Beltrami operator is semi-positive definite and hence its eigenvalues are all non-negative. In fact, the Spectral theorem implies that $\lambda_0 \leq \lambda_1 \leq \lambda_2 ... \leq \lambda_n \longrightarrow \infty$. The eigenvalues $\{\lambda_i\}_{i=0}^{\infty}$ are called the *Laplace-Beltrami-spectra* and when the surface is without boundary λ_0 is equal to zero corresponding to a constant eigenfunction.

Remark 3.12. *It can be shown that on M one has $\Delta = -\text{div} \circ \text{grad}$.*

3.3 The Laplace-Beltrami Operator on a Triangulated 2-Manifold

The Laplace-Beltrami operator theory has many advantages. One can see from the previous section that the definition is invariant to the Riemannaian metric and independent of the embedding the surface. Furthermore, on triangulated 2-manifold, experiments shows that it is stable under small perturbations of the mesh and topological noises. To approximate the smooth Laplace operator we need to introduce an appropriate discretiztion. Several discretiztion have been proposed and the most popular one is the so called *cotanent scheme* [25, 68]. However, it was shown that non of the discretiztions can simultaneously capture all desirable properties of the smooth Laplace-Beltrami operator [94]. For instance, it is impossible to construct a discrete Laplace-Beltrami operator that is symmetric and converges to the smooth version simultaneously. Therefore, one usually chooses a discretization

that suits the application. For our purpose here, we need an appropriate discretiztion of the Laplacian that allows us to decompose an arbitrary scalar function on the mesh in terms of the eigenvectors of the discrete Laplacian. We review the methods introduced in the literature for this purpose. Finally, we will show how the theory of Laplace-Beltrami operator can be applied to reduce the number of critical points of a scalar function defined on a mesh.

3.3.1 Review of The Eigenvalue Problem

Let V be an inner product real-vector space. Recall that a real M matrix is symmetric if $M = M^T$ where M^T is the transpose matrix of the matrix M. It is known from linear algebra that a symmetric real matrix has only real eigenvalues. In other words the only numbers λ that satisfy the equation $Mv = \lambda v$ are real numbers. Moreover, since M is a symmetric matrix then there is an orthonormal basis of V consisting of eigenvector of M. In other words, if M is symmetric then we can write M as follows:

$$M = Q\Lambda Q^T = \sum_{i=1}^{n} \lambda_i e_i e_i^T,$$

where Q is an orthogonal matrix and each column of Q is an eigenvector of M. Moreover, Λ is a diagonal matrix whose entries are the eigenvalues of M.

We will also need to recall the *generalized eigenvalue problem*. A generalized eigenvalue problem is the problem of finding a vector v that satisfies the equation $Mv = \lambda Bv$ where M and B are matrices. If such a vector exists then it is called a *generalized eigenvector of M and B* and its corresponding λ is called a *generalized eigenvalue*. When the matrices M and B are symmetric and B is positive semi-definite (i.e. all its eigenvalues are non-negative) then the generalized eigenvalues are real and if v_1 and v_2 are generalized eigenvectors for M and B with distinct eigenvalues then they are B-orthogonal in the sense $v_1^T B v_2 = 0$. In other words,

suppose that M and B are symmetric and B is positive semi-definite. Then we can always write :

$$E^T M E = \Lambda \text{ and } E^T B E = I.$$

where $E := [e_1, ..., e_n]$ is the matrix of the generalized eigenvectors of M and B and $\Lambda = diag(\lambda_1, ..., \lambda_n)$ is the diagonal matrix of generalized eigenvalues of M and B. Note that when B is equal to the identity matrix, we are back to the standard eigenvalue problem and the inner product induced by I is the standard inner product. The main advantage when M and B satisfy the nice conditions above is that it is guaranteed that we have a basis of generalized eigenvectors as well as $e_i B e_j = \delta_{ij}$.

3.3.2 Discretization of The Laplace-Beltrami Operator

Suppose that M is a triangulated mesh. Denote by $C^{PL}(M)$ to the set of all piece-wise linear functions defined on M. We can define a vector space structure on $C^{PL}(M)$ in the obvious way. Moreover, this space can be given an inner product structure as follows.

Definition 3.13. *Let $[v_i, v_j]$ be an edge in M and let w_{ij} be a constant associated to the edge $[v_i, v_j]$. Let $f, g \in C^{PL}(M)$ then*

$$\langle f, g \rangle = \frac{1}{2} \sum_{[v_i, v_j] \in \mathcal{E}(M)} k_{ij}(f(v_i) - f(v_j))(g(v_i) - g(v_j))$$

defines an inner product on $C^{PL}(M)$.

The choice for the constants w_{ij} usually depends on the application. These weights have been computed to minimize the Dirichlet energy [68] and the mean value theorem [30]. For our purpose in this chapter we will choose the so-called *cotangent weights* [25, 68]. In this case $w_{ij} = \frac{1}{2}(cot(\alpha_{ij}) + cot(\beta_{ij}))$ for any interior edge $[v_i, v_j]$, where α_{ij}, β_{ij} are the angles that are opposite to the edge $[v_i, v_j]$.

When the edge $[v_j, v_j]$ is a boundary edge then $w_{ij} = \frac{1}{2}cot(\alpha_{ij})$ where α_{ij} is the angel that is opposite to the edge $[v_i, v_j]$. The discrete Laplace operator is defined as follows :

Definition 3.14. *The discrete Laplace operator is the linear operator*

$$\begin{aligned} \Delta : C^{PL}(M) &\longrightarrow C^{PL}(M) \\ f &\longrightarrow \Delta f \end{aligned}$$

defined by:

$$\Delta f(v_i) = \sum_{[v_i, v_j] \in \mathcal{E}(M)} w_{ij}(f(v_i) - f(v_j)).$$

In matrix notation the *discrete Laplace matrix* L is given by [68] :

$$L_{ij} = \begin{cases} w_{ij} = \frac{1}{2}(cot(\alpha_{ij}) + cot(\beta_{ij})) & i \neq j \\ -\sum_{k \in N(i)} w_{ik} & i = j \end{cases}$$

where $N(i)$ is the set of all indices of the vertices incident to v_i. The matrix L is one of the earliest geometric methods to discretize the Laplace-Beltrami operator [68]. However, the problem with the matrix L is that weights w_{ij} depends on the mesh sampling. A refinement for this discretization was proposed by [18] where they defined *the Laplace-Beltrami operator on a triangulated mesh* to be the linear operator given via the matrix $\tilde{L} := D^{-1}L$ where $D := diag(d_1, ..., d_{\mathcal{V}})$ is the diagonal matrix whose entries are defined by $d_i := \frac{1}{3}\sum_{T \in N(i)} Area(T)$. Here $Area(T)$ refers to the area of the triangle T. The matrix D is called the *diagonal mass matrix*. However, since d_i may vary from a vertex to another then the matrix $\tilde{L}$ may not be a symmetric matrix and the eigenvalues are not guaranteed to be real. Although $\tilde{L}$ is not symmetric, it is similar to a symmetric matrix [50] :

$$\tilde{L} = D^{-1}L = D^{-1/2}D^{-1/2}LD^{-1/2}D^{1/2} = D^{-1/2}OD^{1/2}$$

Since the symmetric matrix $O = D^{-1/2}LD^{1/2}$ is similar to the matrix $\tilde{L}$ then they both have the same real eigenvalues. Furthermore, if w_i is an eigenvector for O then $u_i := D^{-1/2}w_i$ satisfies:

$$\tilde{L}u_i = D^{-1/2}OD^{1/2}D^{-1/2}w_i = D^{-1/2}Ow_i = \lambda_i D^{-1/2}w_i = \lambda_i D^{-1/2}D^{1/2}u_i = \lambda_i u_i$$

and hence u_i is an eigenvector for $\tilde{L}$. So one may consider solving the eigenvectors of the matrix O [50]. However, this method is not stable and it is not preferred in practice. Alternatively, one considers the generalized eigenvalue problem of the matrices L and D. In other words, we look at the the problem $L\phi = \lambda D\phi$. Note here that L is a symmetric matrix and D is a symmetric positive-definite matrix. As we mentioned earlier this system admits a complete set of generalized eigenvector $\{\phi\}_{i=0}^{|V|-1}$ that are orthogonal with respect to the inner product $\langle f, g\rangle_D := f^T Dg$.

Another method suggested in the literature is the *discrete FEM Laplacian*. In this method we consider the the generalized eigensystem of L and B where B is the *FEM mass matrix* [72, 91] defined by:

$$B_{ij} = \begin{cases} \frac{1}{12}(Area(T_a)) + Area(T_b)) & j \in N(i), \\ \frac{1}{6}\sum_{k\in T(i)} Area(T_k) & i = j, \\ 0 & \text{else}, \end{cases}$$

here T_a and T_b are the triangles opposite to the edge $[v_i, v_j]$ and $T(i)$ is the set of indices of the triangles that are incident to the vertex i. Note that if FEM mass matrix is lumped (add all the entries in each row to the diagonal element) then it becomes equal to diagonal mass matrix D [91]. Furthermore, if $B = I$ then we obtain the discrete Laplace operator with cotangent weight.

Alternatively, the generalized eigensystem L and $\hat{D}_{mixed}$ can also be considered, where $\hat{D}_{mixed} = diag(A_{mixed}(v_1), ...A_{mixed}(v_\mathcal{V}))$ and $A_{mixed}(v_i)$ is a surface area around the vertex v_i given in the following algorithm [57]:

Algorithm 1 Computing $A_{mixed}(v_i)$

Let $A_{mixed}(v_i) = 0$
For each face F incident to the vertex v_i:
    If F is not an obtuse triangle then
        $A_{mixed}(v) = A_{mixed}(v) + \frac{1}{8}(\cot\theta_{ij}||(v_i - v_j)||^2 + \cot\beta_{ik}||(v_i - v_k)||^2)$
    Else if the obstuse angle is at vertex v then $A_{mixed}(v_i) = A_{mixed}(v_i) + \frac{1}{2}A(F)$
    Else $A_{mixed}(v_i) = A_{mixed}(v_i) + \frac{1}{4}A(F)$

Instead of solving the spectrum of the generalized eigensystem $Lv = \lambda Dv$, Levy [48] considers solving the the eigensystem $Sv = \lambda v$ where S is the symmetric matrix $\frac{1}{2}(\tilde{L} + \tilde{L}^T)$.

3.3.3 Eigenfunctions of the Laplacian

We are interested in the eigenvectors of the Laplacian for mainly two reasons. The first reason is that they provide a natural set of scalar functions on the mesh that have many desirable geometric properties such as :

1. The eigenvectors of the Laplacian are invariant to various transformations and small topological changes.
2. In many cases the isolines of the eigenvectors follow the curvature of the surface.
3. The critical points of these scalar functions coincide with the feature and/or symmetric points on the surface.
4. They can be used to study the symmetry of surfaces.

Figure 3.1 shows the first 8 eigenvectors of the system L and $\hat{D}_{mixed}$ on a genus zero surface. Notice that the first non-constant eigenvector follows the geometry of the surface. This seems to be generally the case on a genus zero surface.

FIGURE 3.1. Eigenfunction of the Laplacian of genus zero surface.

Figure 3.2 shows the first 16 eigenvectors of the system L and $\hat{D}_{mixed}$. Notice how the eigenfunctions are symmetric reflecting the symmetry of the surface itself.

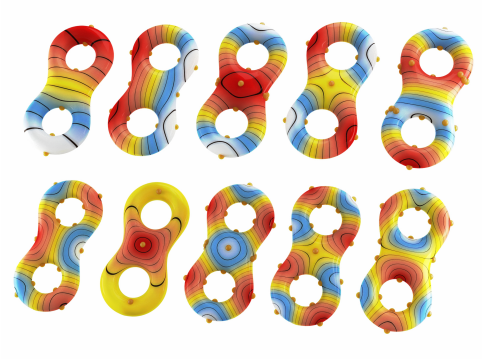


FIGURE 3.2. The first generalized eigenvectors using the system L and $\hat{D}_{mixed}$.

It is not always true that the first non-constant eigenvector provides a good geometrically desirable scalar field. This is particularly noticeable for higher genus surfaces. Figure 3.3 shows the first eigenvectors of the system L and $\hat{D}_{mixed}$ on a higher genus surface. Note that the eigenfunctions 12 and 13 represent the symmetry and the geometry of the mesh better than any other eigenvector shown in Figure3.3. The second reason for our interest in the eigenvectors of the Laplacian

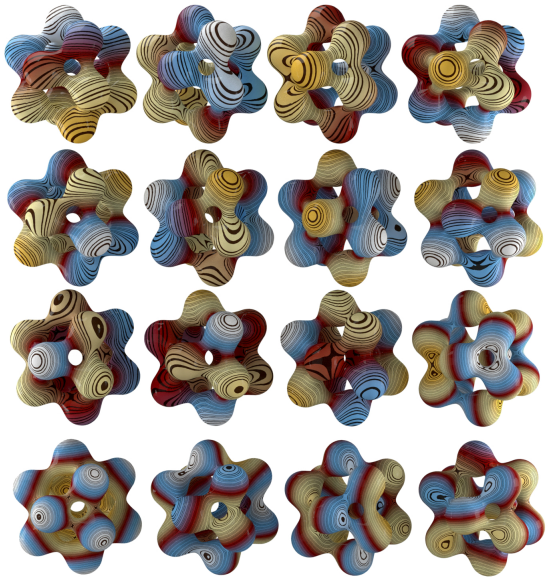


FIGURE 3.3. Eigenfunction of the Laplacian of a higher genus surface.

in this chapter is that they can be used to reduce the number of undesirable critical points of a scalar function. This is the subject of the next section.

3.3.4 Smoothing Scalar Function Using the Laplace-Beltrami Operator

Let M be a triangulated 2-manifold. Consider the problem of having a scalar function f on M with many undesirable critical points. The theory of the Laplace-Beltrami operator that we introduced earlier can be used to obtain a smooth version of f as follows. Let $\mathcal{B} := \{\phi_i\}_{i=0}^{|\mathcal{V}|-1}$ be the set of of eigenvectors of the discrete Laplace-Beltrami operator of the mesh M. Recall here that the eigenvectors are ordered according to the increasing values of their corresponding eigenvalues. In other words, eigenvectors with higher eigenvalues have higher index i. Since $\mathcal{B}$ is an orthonormal basis then :

$$f = \sum_{i=0}^{|\mathcal{V}|-1} \alpha_i \phi_i \tag{3.4}$$

where $\alpha_i =< \phi_i, f >$. Now the idea is that the frequency of eigenfunction ϕ_i increases with the index i. Assuming that the noisy part of a scalar function lies in the high frequency terms of the summation 3.4, we can reconstruct a smoothed

version $\hat{f}$ of f by approximating the summation with the first m eigenvectors where $m < |\mathcal{V}| - 1$. This has the effect of smoothing the function f and getting ride of unnecessary critical points in of the scalar function f. This idea has been used by [91] to smooth the geometry of the mesh. In other words, the abstract scalar function f is replaced by the coordinates x y and z of the vertices of the mesh. Figure 3.4 shows a scalar function with multiple critical points on the right and a an approximation of this scalar function with the first 8 eigenfunctions of the system L and $\hat{D}_{mixed}$ specified above.

FIGURE 3.4. A noisy function f and its smoothed version $\hat{f}$.

Figure 3.5 shows also the same idea on a different model.

FIGURE 3.5. A noisy function f and its smoothed version $\hat{f}$.

Even though smoothing a function using this method is very effective but computing the eigenvectors of a large mesh is expensive computationally. For details

about the numerical and computational issue see [50]. In the next section we will study a cheap alternative method for the spectral method.

3.4 Heat Diffusion Equation

The heat equation studies the propagation of heat along a certain domain. Let M be a Riemannian manifold. The heat function $u : M \times \mathbb{R}^+ \longrightarrow \mathbb{R}$ is by definition the solution of the *heat diffusion equation* :

$$\begin{cases} \Delta u(x,t) = \frac{\partial}{\partial t} u(x,t) \\ u(x,0) = h(x) \end{cases} \quad x \in M,\ t \in \mathbb{R}.$$

The value $u(x,t)$ represents the heat at point x and time t. The heat equation plays a fundamental role in geometric processing [86]. The solution for the heat equation can be written from the convolution of the heat kernel and the function h :

$$u(x,t) = \int_M K(x,y,t)h(y)dM(y) \tag{3.5}$$

where $K(x,y,t)$ is the *heat diffusion kernel*:

$$K(x,y,t) = \sum_{i=0}^{\infty} e^{-\lambda_i t}\phi_i(x)\phi_i(y) \tag{3.6}$$

and this yields easily :

$$u(x,t) = \sum_{i=0}^{\infty} < h, \phi_i > e^{-\lambda_i t}\phi_i(x) \tag{3.7}$$

Even though we can use the previous formula in practice to smooth a scalar function h in order to remove unwanted critical points but this method is not fundamentally different from the method we introduced in the last section. Let $u_x(t) := u(x,t)$ and consider the approximation of $\frac{\partial u_x(t)}{\partial t}$:

$$\frac{\partial u_x(t)}{\partial t} \simeq \frac{u_x(t+h) - u_x(t)}{h}.$$

This yields :

$$u_x(t+h) = u_x(t) + h\Delta u_x(t) \tag{3.8}$$

Many variation of this formula have been proposed. In particular, one can alternatively solve the more stable *implicit Euler integration* system :

$$(Id - hL)u_x(t+h) = u_x(h) \tag{3.9}$$

Equations 3.8 and 4.15 have been employed in computer graphics for mesh smoothing [87, 18]. For our purpose, we use systems 3.8 and 4.15 in order to remove unwanted critical point of f and obtained a smoothed version of f. For instance, in Figure 3.6 we have a function $f : M \longrightarrow \mathbb{R}$ defined on the female model and appears on the right hand side of the Figure. We applied the smoothing operation 3.8 on the function f twenty times with $h = 0.5$ and we obtain the scalar function in the middle which has critical points at better location than the initial f. Applying the smoothing operation 3.8 the scalar function converges to a function that matches the geometry of the mesh. One can obtain similar results using the the implicit system 4.15.

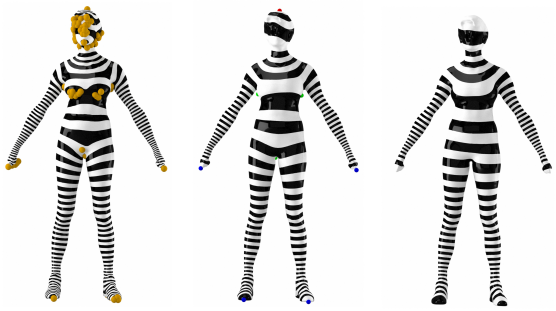


FIGURE 3.6. Smoothing a function using the heat diffusion equation.

Chapter 4

Geometry and Symmetry Aware Scalar Functions on Triangulated 2-Manifolds

4.1 Introduction

Many problems in computer graphics benefit from solving certain scalar function on a mesh. This includes shape approximation [13], skeletonization [5, 79], surface quadrangulation [22, 23, 15], mesh editing [98], shape analysis [**?**], mesh reconstruction [43], shape matching [64], cutting a surface into a disk [**?**], extracting feature lines on a mesh [75], symmetry detection [44, 93] and many other applications. In this chapter we give multiple methods to construct scalar functions on triangulated 2-manifolds such that these functions have *desirable* geometric properties. By the term *desirable* we mean one or more of the following properties :

1. The isolines of the scalar function are shape-aware in the sense that they follow one of the principal directions of the surface.
2. The critical points of the scalar function coincide with feature or the symmetry points on the surface.
3. If the surface has some sort of symmetry then the scalar field also inherits the symmetry of the surface.
4. Minimal user input. In fact, our methods require no user input or an input one initial condition.

In this chapter we review several geometry and symmetry aware scalar functions, suggest multiple improvements to existing methods and we give new constructions for such functions. For each one of the scalar functions we list here we pay particular attention to keep the configuration of the set of critical points of the scalar function is optimal. These scalar functions can be applied in many areas in ge-

ometric processing such as parametrization, shape analysis, symmetry detection and segmentation. In particular a desirable Morse scalar function is a fundamental task for any algorithm that uses Morse theory. Our work utilizes heavily the theory of the Laplace-Beltrami operator to design a scalar function on mesh:

1. One can use the eigenfunctions of the Laplacian to obtain scalar functions that are isometry invariant. This means that they have *intrinsic symmetry* [44, 65] with respect to the surface as well as the distribution of their critical points on the surface. We give new families of such symmetric functions and we show how to make existing symmetric fields in the literature more robust with respect to the configuration of their critical points on the manifold. In particular we also introduce the *Modified Heat Kernel Signature* and the *Modified Auto Diffusion Function*, which are parameterized families of isometry invariant scalar functions that generalize Heat Kernel Signature [86] and Auto Diffusion Function [31] respectively.
2. We use the biharmonic [53] distance to study the average geodesic distance, or AGD, function on the manifold [36]. The AGD function has been proven to have intrinsic symmetry [44] but it has only been studied in the context of Dijkstra's algorithm geodesic distance function which has many drawbacks. We show how that by using the nice properties of the biharmonic one can gain many desirable properties that were not available with the Dijkstra's algorithm geodesic distance function.
3. Poisson fields on meshes [23] have been used to obtain a geometry aware scalar functions. We show that the discrete conformal factor [8, 9] can be used for the same purpose and propose a robust modification for the discrete conformal factor solution suggested in [8]. The proposed method make the discrete conformal factor more suitable for applications such as parametriza-

tion, segmentation and Morse theory analysis. Furthermore, we also demonstrate that the heat kernel map suggested in [64] can be used for similar purposes.

4.2 Poisson Equation on a Smooth Surface

The Poisson equation on Riemannian surface (M, g) is given by:

$$Lf = \nabla.X$$

where L is the Laplace-Beltrami operator and V is a vector field on X. The vector field X is usually called *the guidance vector field*. The reason for this name is the following. The Poisson equation is actually equivalent to the minimization problem:

$$\min_f \int_M |\nabla f - X| dM,$$

which means intuitively that the solution f for the Poisson equation $Lf = \nabla.X$ is the function f whose gradient ∇f is as close as possible to the guidance vector field X.

We are interested here in solving such an equation with Dirichlet boundary condition. In order to be able to solve such equation on a triangulated mesh one needs to make sense of the divergence and gradient operators. This is the purpose of the next section.

4.3 The Poisson Equation on a Triangulated 2-Manifold

In this section we study the Poisson equation on a triangulated 2-manifold. For this purpose, we need to make sense of the following two operations on a mesh M:

1. Given a piece-wise linear function $f : M \longrightarrow \mathbb{R}$ how can we calculate ∇f the gradient of f?

2. Given a vector field $X : \mathcal{F}(M) \longrightarrow \mathbb{R}^3$, what is the appropriated definition of div X ?

The previous two questions are addressed in the literature [68, 69]. Moreover the discrete divergence and the gradient operators are consistent with the discrete Laplace-Beltrami operator on a triangulated in the sense that the equation $\Delta = -\text{div} \circ \text{grad}$ holds on a triangulated 2-manifold.

4.3.1 The Gradient of a PL Function

We first calculate the gradient of a PL function defined on a single face and then we use this to define the gradient of a function defined on any triangulated 2-manifold.

Lemma 4.1. *[68] Let $f : F \longrightarrow \mathbb{R}$ be a piece-wise linear function defined on a face $F = [v_1, v_2, v_3]$ via the values $f(v_i)$, $i = 1, 2, 3$. Let E_i be the counterclockwise oriented edge opposite to the vertex v_i. Then the gradient of f is a constant and tangential vector on F given by :*

$$\text{grad } f = \nabla f = \frac{1}{2A_F} \sum_{i=1}^{3} f(v_i)||E_i||\overrightarrow{u_i} \tag{4.1}$$

where u_i is is a unit vector perpendicular to the edge E_i and oriented so that it

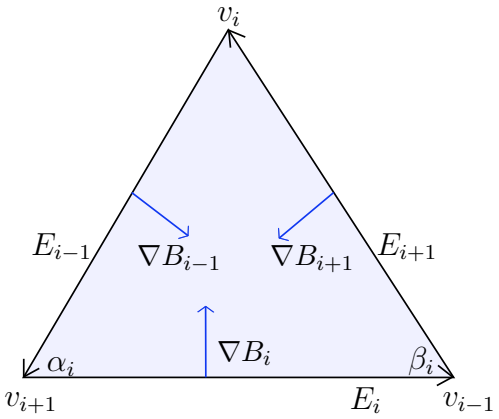


FIGURE 4.1. The gradients of hat functions of triangle.

points into the face F. Moreover, $u_i = \frac{1}{||E_i||}(E_{i+1}\cot\alpha_i - E_{i-1}\cot\beta_i)$ where α_i and β_i are the angles on both sides of the edge E_i, α_i is the angle opposite the edge E_{i+1} and β_i is the angle opposite to the edge E_{i-1}.

Proof. Let r be an arbitrary point inside the triangle F. Then r can be written as

$$r = B_1(r)v_1 + B_2(r)v_2 + B_3(r)v_3.$$

where $B_i : F \longrightarrow \mathbb{R}$ is the hat function on the vertex v_i defined by $B_i(v_j) = \delta_{ij}$ for $i, j = 1, 2, 3$. Hence any the function f has the form:

$$f(r) = \sum_{i=1}^{3} f(v_i)B_i(r) \tag{4.2}$$

Now the barycentric coordinate B_i is given by the ratio $B_i = \frac{A_i}{A_F}$ where $A_i = \frac{1}{2}h_i||E_i||$. See Figure 4.2. Since A_F and $||E_i||$ are invariant for the same face F,

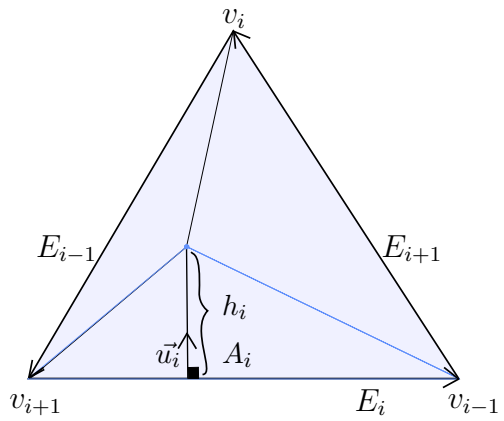


FIGURE 4.2. Barycentric coordinates.

then B_i is a function of h_i. Hence

$$\nabla B_i = \frac{d}{dh_i}\Big(\frac{h_i||E_i||}{2A_F}\overrightarrow{u}\Big) = \frac{||E_i||}{2A_T}\overrightarrow{u_i} \tag{4.3}$$

where $\overrightarrow{u_i}$ is a unit vector perpendicular to the vector E_i and oriented so that it points into the the face F. Now equation (4.2) implies :

$$\nabla f = \sum_{i=1}^{3} \nabla B_i f(v_i),$$

Hence (4.1) follows. For the second part of the Lemma, choosing $\overrightarrow{u_i} = \frac{E_{i-1}\times E_i}{||E_{i-1}\times E_i||} \times \frac{E_i}{||E_i||}$, equation (4.3) becomes:

$$\nabla B_i = \frac{1}{2A_F}\frac{E_{i-1} \times E_i}{||E_{i-1} \times E_i||} \times E_i \tag{4.4}$$

However,

$$\begin{aligned}
\frac{E_{i-1}\times E_i}{||E_{i-1}\times E_i||}\times E_i &= -\frac{\langle E_i,E_i\rangle E_{i-1}+\langle E_i,E_{i-1}\rangle E_i}{||E_{i-1}\times E_i||}\\
&= -\frac{\langle E_i,E_i\rangle E_{i-1}-\langle E_i,E_{i-1}\rangle E_{i-1}+\langle E_i,E_{i-1}\rangle E_{i-1}+\langle E_i,E_{i-1}\rangle E_i}{||E_{i-1}\times E_i||}\\
&= \frac{-\langle E_i,E_{i+1}\rangle E_{i-1}+\langle E_i,E_{i-1}\rangle E_{i+1}}{||E_{i-1}\times E_i||}\\
&= -\frac{\langle E_i,E_{i+1}\rangle E_{i-1}}{||E_i\times E_{i+1}||}+\frac{\langle E_i,E_{i-1}\rangle E_{i+1}}{||E_{i-1}\times E_i||}
\end{aligned}$$

Since the cotangent of an angel between two vectors v and w is equal to $\frac{\langle v,w\rangle}{||v\times w||}$, then

$$\frac{E_{i-1}\times E_i}{||E_{i-1}\times E_i||}\times E_i = (\cot\alpha)E_{i+1}-(\cot\beta)E_{i-1}. \tag{4.5}$$

The result follows. □

Now suppose that M is a triangulated 2-manifold and $f : M \longrightarrow \mathbb{R}$ is a PL function. Then ∇f is a vector field $\nabla f : \mathcal{F}(M) \longrightarrow \mathbb{R}^3$ assigns to every face F in $\mathcal{F}(M)$ a vector defined as in the previous Lemma.

4.3.2 The Divergence Operator on a Triangulated 2-Manifold

As for the the divergence of a vector field X defined on the faces of a triangulated mesh. We need a definition that is consistent with the definition of the discrete Laplacian that we gave in the previous chapter. In other words, we need to define an operator div $X : M \longrightarrow \mathbf{R}$ such that $\Delta f = \text{div } \nabla f$ for any scalar function f on M. Such a definition of the divergence operator was actually given in [69] and it is explicitly given by :

$$\text{div } X(v_i) = \frac{1}{2}\sum_{j\in F(i)} \cot\theta_{j_1}\langle e_{j_1}, X_j\rangle + \cot\theta_{j_2}\langle e_{j_2}, X_j\rangle,$$

where $F(i)$ is the set of indices of all faces that are incident to the vertex v_i, e_{j_1}, e_{j_2} are the two vectors in face j that contain the vertex v_i and $\theta_{j_1}, \theta_{j_2}$ are the angles that are opposite the the edges e_{j_1} and e_{j_2} respectively. See Figure 4.3.

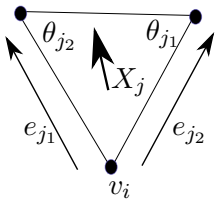


FIGURE 4.3. Computing the divergence at vertex v_i

4.3.3 The Poisson Equation With Dirichet Boundary Condition on a Triangulated 2-Manifold

The Poisson equation with Dirichlet boundary condition is given by :

$$\Delta f = h, \quad f(v_i) = c_i \text{ where } v_i \in V_C \tag{4.6}$$

where $V_C \subset V$ is a set of constrained vertices and $h : M \longrightarrow \mathbb{R}$ is a known function. The cardinality of the set V_C must be at least 1 in order for system (4.6) to have a unique solution. Using the the discrete operators we defined earlier the Poisson equation can be easily solved on a triangulated 2-manifold. Indeed, on a triangulated 2-manifold, solving the piece-wise linear function f in equation (4.6) is equivalent to solving the sparse linear system

$$Af = b$$

where

$$A_{ij} = \begin{cases} \delta_{ij} & v_i \in V_C \\ L_{ij} & else \end{cases} \tag{4.7}$$

and b is the vector defined by :

$$b_i = \begin{cases} c_i & v_i \in V_C \\ h(v_i) & else \end{cases}$$

In the literature Poisson equation has many application. This includes quadratic meshing [23], hole-filling [101], skeletonization [77], and mesh editing [84]. With the

approportae choice of g we can use the Poisson equation for our purpose. Indeed, the solution f of Poisson equation gives us a scalar field whose level sets follow one of the principal curvatures of the manifold. In [23] Dong and *et al.* suggested using the following Poisson equation to obtain such a scalar field :

$$\Delta f(v) = \kappa(v) \text{ where } f(v_{source}) = c \tag{4.8}$$

where $\kappa(v)$ is the mean curvature at the vertex v:

$$\kappa(v_i) = \frac{1}{4A_{mixed}(v_i)} \sum_{j \in N(i)} (\cot \theta_{ij} + \cot \beta_{ij}) ||(v_i - v_j)|| \tag{4.9}$$

here the angles are given in Figure 4.4

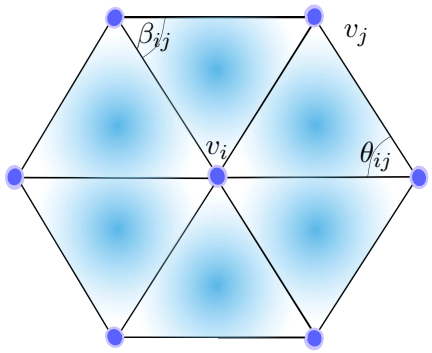


FIGURE 4.4. The angles θ_{ij} and β_{ij}.

and $A_{mixed}(v_i)$ is the surface mixed area around the vertex v_i [57]. See for the algorithm to compute $A_{mixed}(v_i)$. Figure 4.5 shows examples of solutions of Poisson solution obtained from the system 4.8.

4.3.4 Unit Gradient Scalar Fields

Suppose that we are given a scalar function $f : M \longrightarrow \mathbb{R}$ on a triangulated mesh M. We are interested in finding a function g whose gradient has constant magnitude on the entire mesh and whose isolines are parallel to those of the function f. Such a scalar function can be obtained as follows:

1. Compute $X = \frac{\text{grad } f}{||\text{grad } f||}$.

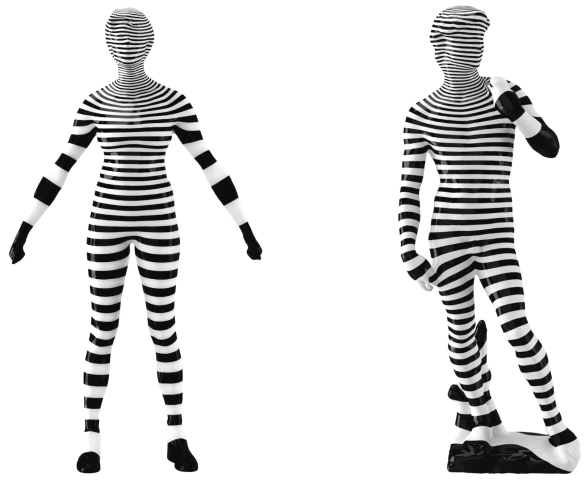


FIGURE 4.5. Examples of Poisson fields.

2. Solve the Poisson equation $\Delta g = \text{div } X$

The previous algorithm works on any generic function f such that the gradient of f makes sense. An example of such a procedure is illustrated in Figure 4.6. The the left model in the figure shows a solution for a scalar function f obtained as a Poisson equation $\Delta f = \kappa$ where we choose the constrained vertex to be the top of the head of the model. The model on the right hand side shows the function g obtained as we explained above in steps 1 and 2. Here the constrained vertex is the same also the top of the head of the model.

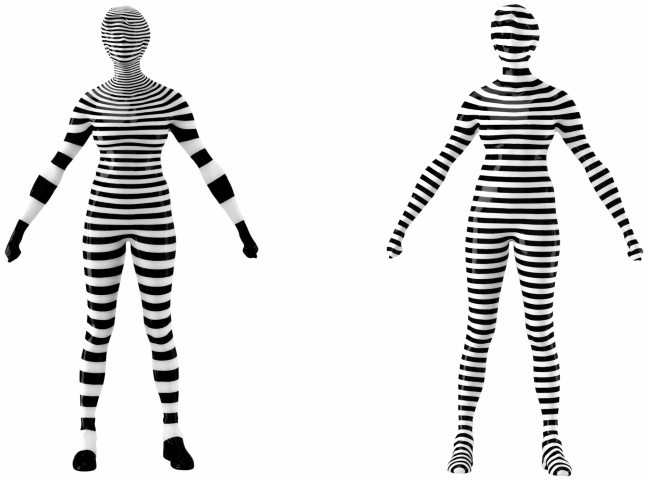


FIGURE 4.6. Obtaining a unit gradient scalar field.

Solving the system $\Delta g = \text{div } X$ with $g(v_{source}) = c$ usually yields new critical points on the the vertex v_{source} that did not exist in the original solution f, see Figure 4.7. These critical points can be removed using the techniques we introduced in the previous chapter or simply we can set $g(v_{source})$ to be very large negative number. This has the effect of "absorbing" the extra critical points around the vertex v_{source}.

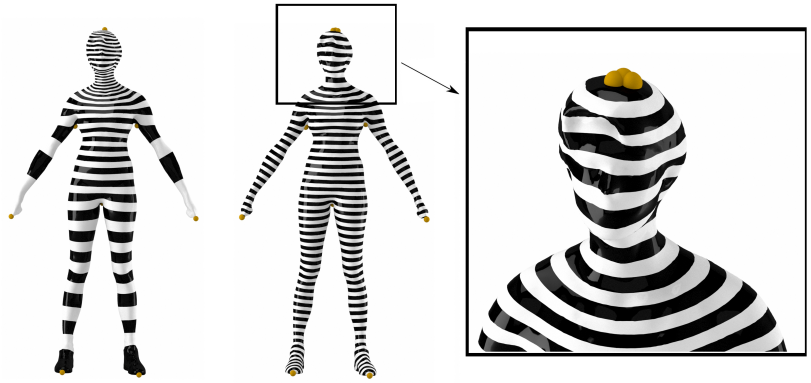


FIGURE 4.7. Fields f (left) and g (right) share same critical points except at v_{source}.

4.4 Discrete Conformal Factors

Recall that the discrete Gaussian curvature at a vertex v on triangulated 2-manifold M is given by :

$$k_v = \begin{cases} 2\pi - \sum_{f \in \mathcal{F}(v)} \alpha^f_{v_i} & v_i \in \mathcal{V} \backslash B \\ \pi - \sum_{f \in \mathcal{F}(v)} \alpha^f_{v_i} & v_i \in B \end{cases}$$

where B is the set of boundary vertices in the mesh M. The following definition is suggested by [8].

Definition 4.2. *Let $K^{orig} = (k_{v_i})$ be the vector of the Gaussian curvature of the vertices $\mathcal{V} = \{v_i\}_{i=1}^n$ of the mesh M and let A be the matrix 4.7. The discrete*

conformal factor, or DKF, is the solution f for the following linear system:

$$Af = K^T - K^{orig} \tag{4.10}$$

where $K^T = (k^T_{v_i})$ is an n vector of the target Gaussian curvatures of the vertices of the mesh [8] :

$$k^T_{v_i} = (\sum_{v \in V(M)} k_v) \frac{\frac{1}{3}\sum_{f \in F_v} Area(f)}{\sum_{f \in \mathcal{F}} Area(f)} \tag{4.11}$$

The discrete conformal factor has applications in parameterization [9] and in shape characterization [8] and many other applications. Notice again that the system 4.11 requires choosing one constrained vertex in order to have a unique solution. We suggest a modification of the target Gaussian curvature as follows:

$$k^{MT}_{v_i} = (\sum_{v \in V(M)} k_v) A_{mixed}(v_i) \tag{4.12}$$

We call this curvature the *modified target Gaussian Curvature*, and we call the solution of the system $Af = K^{MT} - K^{orig}$ the *modified discrete conformal factor* or MDCF. In our experimentation we found that the MDCF factor has better geometric properties that the DKF. For instance, the number of the critical points is for the MDCF is less than the number of critical points on for the DCF and the distribution of the critical points of the MDCF follows in general the geometry of the shape better in the case of DCF. See Figure 4.8.

A more obvious comparison between the critical points of the two functions is given in Figure 4.9.

On surfaces with higher genus $MDCF$ still gives results that are comparable with with the Poisson field. However, the DCF did not give us very good results

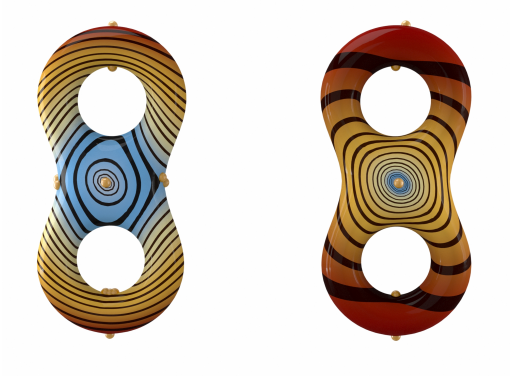


FIGURE 4.8. DCF(left) and $MDCF$(right)

FIGURE 4.9. DCF(left) and $MDCF$(right)

4.5 Harmonic Functions

In many applications in computer graphics one needs to design a scalar function that has local extrema only on a specific set of points on the manifold. One can obtain such a function as a solution for the *Laplace equation subject to Dirichlet boundary condition*. More precisely, we are seeking a scalar function f which satisfies the Laplace equation $\Delta f = 0$ subject to the Dirichlet boundary conditions $f(v_i) = c_i$ for all $v_i \in V_C$. Here $V_C \subset V$ is a set of constrained vertices and c_i are known scalar values providing the boundary conditions. This system has a unique solution provided $|V_C| \geq 2$. Furthermore, the solution for such a system has an important property that it has no local extrema other than the constrained vertices. This property is usually called the maximum principle [74]. Designing such a

FIGURE 4.10. DCF (left) and $MDCF$ (right).

function is possible in practice. Recall that on triangulated mesh M the standard discrtization for the Laplacian operator at a vertex v_i is given by :

$$\Delta f(v_i) = \sum_{[v_i,v_j]\in M} w_{ij}(f(v_j) - f(v_i)),$$

where w_{ij} is a scalar weight assigned to the edge $[v_i, v_j]$ such that $\sum_{[v_i,v_j]\in M} w_{,j} = 1$. Choosing the weights w_{ij} such that $w_{ij} > 0$ for all edges $[v_i, v_j]$ will guarantee the solution has no local extrema other than at constrained vertices V_C [30]. These conditions are satisfied by the *mean value weights*:

$$w_{ij} = \frac{tan(\theta_{ij}/2) + tan(\phi_{ij}/2)}{||v_j - v_i||},$$

where the angles θ_{ij} and ϕ_{ij} are the angles on either sides of the edge $[v_i, v_j]$ at the vertex v_i. Mean value weights are used to approximate harmonic map and they have the advantage that they are always non-negative which prevents any introduction of extrema on non-constrained vertices in the solution of the Laplace equation specified above. On the other hand, the cotangent weights may become negative in presence of oblique triangles and this can produce local extrema on non-constrained vertices.

Harmonic maps are excellent source of scalar functions on surfaces. The main disadvantage of using harmonic functions is that one needs to specify the initial conditions manually. Even though there are multiple methods to semi-automate selecting the constrained vertices [23, 5], these methods do not always produce optimal results and might product critical points at undesired locations. On the other hand, in certain applications one might need a scalar function with certain properties that harmonic functions are the most natural class of functions that satisfy such properties. In Chapter 5 we will see that one of our pants decomposition algorithms rely on having a scalar function that has precisely one global minimum and one global maximum. Such a function can be obtained as a solution for Laplace equation with mean value weights and with only two constrained vertices $V_C = \{v_{min}, v_{max}\}$ such that $f(v_{min}) < f(v_{max})$.

4.6 The Eigenfunctions of The Laplacian

The eigenfunctions of the Laplacian that we studied in the previous chapter provide excellent source of scalar functions that have many desirable geometric features. The first few eigenfunctions are particularly useful. Figure 4.12 shows the first two non-trivial eigenfunctions on a genus 2 surface as well as their critical points.

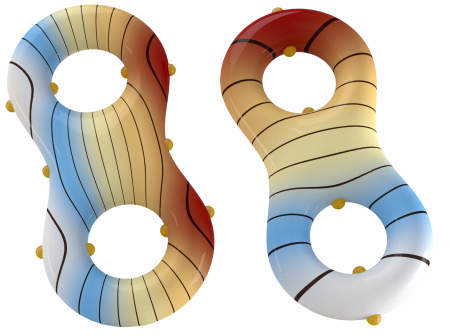


FIGURE 4.11. The first two eigenvectors of the Laplacian.

For instance, the first non-trivial, eigenfunction seems to produce very similar to the Poisson field especially on genus-zero surfaces. This eigenfunction, called Fiedler vector [28, 29] , has many applications in graph theory as well as in computer graphics [37, 61]. This is however not always the case for higher genus

FIGURE 4.12. The Fiedler vector on a genus zero surface.

surfaces the most natural scalar functions usually correspond to eigenfunctions with higher eigenvalues. For instance the eigenfunctions that are shown in Figure 4.13 are ϕ_{13} and ϕ_{14} of this model.

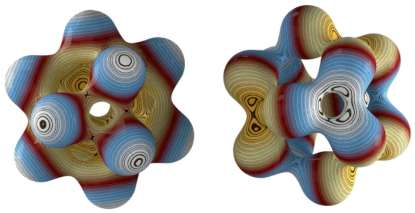


FIGURE 4.13. Higher eigenfunctions on a surface with high genus.

4.7 The Heat Kernal Map

The *heat kernel map* of an orientable, closed Riemannian manifold M, a source point $p \in M$ and a parameter $t \in \mathbb{R}^+$ is the scalar function :

$$\begin{aligned} \Phi_{p,t} : M &\longrightarrow \mathbb{R} \\ x &\longrightarrow \Phi_{p,t}(x) = K(p,x,t) \end{aligned}$$

where $K(p, x, t)$ is the heat diffusion kernel. This map was introduced in [64] for the purpose of matching between two surfaces. For us this function gives us t-parametrized family of scalar functions that has desirable geometric and analytic features. For instance, for a relatively large t this function has properties similar to the Poisson and discrete conformal factors scalar fields. However, due to the nice properties of the eigenfunctions of the Laplacian this function often does not require postprocessing like the Poisson and the discrete conformal factor functions. As mentioned in the definition of this function, one also needs to specify a source point for heat kernel map. This point should be chosen to be a point of symmetry or a feature point if one needs to obtain a desirable scalar field. For instance Figure4.14 shows the function $\Phi_{p,2}$ on a genus zero mesh where p is chosen to be the highest point on the head of the model. On the other hand, this function along

FIGURE 4.14. The scalar function $\Phi_{p,2}$.

with the Poisson field and the scalar conformal factors do not always produce good results on high genus symmetric meshes. To fix this for the heat kernel map one can add multiple heat kernel maps each one with a different source point. More precisely, let $p_1, .., p_k$ be points on the manifold M and let $t \in \mathbb{R}^+$ define the *multi-source heat kernel map* to be the summation:

$$MHKM_{p_1,...,p_k,t} = \sum_{i=1}^{k} \Phi(p_i, t)$$

Choosing the points $p_1, ..., p_k$ to be points of symmetry in the manifold yields a scalar function with symmetric features. See Figure 4.15 for an example of the field MHKM calculated with 4 points of symmetry. The same idea can be applied to the

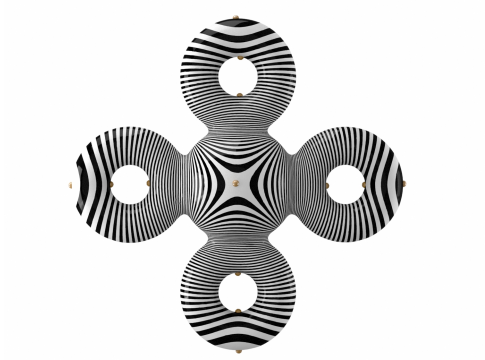


FIGURE 4.15. The field MHKM calculated with 4 points of symmetry.

Poisson field and the discrete conformal factors. This method however, suffers from the fact that one needs to select the symmetry points manually. In the following two sections we give several scalar functions that do not have this drawback.

4.8 Average Geodesic Distance

Another scalar function that exhibits desirable geometric and symmetric properties is the *average distance* function. On a Riemannian manifold (M, g) the average distance function is defined by :

$$AD(p) := \int_M distance(p, q) dM(q) \tag{4.13}$$

where $distance : M \times M \longrightarrow \mathbb{R}^+$ is a metric function defined on M. The most popular distance function is the geodesic distance, or the Riemannian distance, inherited from the Riemannain structure. We will denote to this Riemannian distance by d_g. The metric d_g measures the length of shortest path on a surface between two points. We will refer to the function 4.13 by the *average geodesic distance* when we talk about $distance = d_g$ and we will denote to it by AGD.

On triangulated 2-manifold M the average distance function can be defined by [36]

$$AD(w) = \sum_{v \in \mathcal{V}(M)} distance(w, v) Area(v) \tag{4.14}$$

where $Area(v)$ is defined to be the $1/3$ of the area of 1-ring neighborhood around v, namely : $\frac{1}{3} \sum_{f \in F_v} A(f)$ or we can choose $Area(v)$ to be $A_{mixed}(v)$ which is defined in 4.3.3. As mentioned above, the most interesting distance functions are the ones geodesic distances. For this reason, one a triangulated mesh many algorithms were proposed in the literature to compute the function d_g accurately with a high level of accuracy [80, 39, 45]. The most common choice for to approximate the geodesic distance seems to be the Dijkstra's algorithm based on edge length.

The function 4.13 can be modified so that it is scaling invariant [36]. The main advantage of this function over the Poisson field is that we do not ask the user here to specify any point as an input. However, the AGD based on Dijkstra's algorithm is in general non-smooth and it has many undesirable critical points. See Figure 4.16. For this reason we propose the average geodesic distance based on the biharmonic distance proposed by [53].

FIGURE 4.16. The field AGD and its critical points.

Definition 4.3. *Let M be a triangulated 2-manifold and let $\{\lambda_i\}_{i=0}^{|\mathcal{V}|-1}$ be the set of generalized eigenvalues of discrete Laplace-Beltrami operator and $\{\phi\}_{i=0}^{|\mathcal{V}|-1}$ the corresponding eigenvectors. The discrete biharmonic distance between the vertices x and y on the mesh M is defined by :*

$$d_B(x,y) := \sqrt{\sum_{i=1}^{|\mathcal{V}|-1} \frac{(\phi_i(x) - \phi_i(y))^2}{\lambda_i^2}}. \tag{4.15}$$

Biharmonic distance enjoys many good properties that one wants a distance function to have. For instance, it is smooth everywhere except at the source point. We will denote the average geodesic obtained using the distance d_B by AGD_{BH}.

Other related distance functions are diffusion distance and commute-time distance [97]. However, as demonstrated in [53] biharmonic distance enjoys better properties so we will focus our study on this function.

In practice we approximate the summation 4.15 by considering the first few eigenvectors since the higher eigenvectors do not contribute to the summation significantly since we are dividing by λ_k^2. Recall here that the higher the index k the larger the eigenvalue λ_k. In fact, λ_k grows linearly in k [10]. Figure 4.17 shows the function AGD_{BH} approximated with 6 eigenvectors.

In fact, if we consider approximating the summation 4.15 with the Fiedler vector then AGD_{BH} can be approximated by :

$$AGD_{BH}(w) \simeq \sum_{v \in \mathcal{V}(M)} |\phi_1(v) - \phi_1(w)| Area(v) \tag{4.16}$$

Even with one eigenfunction the AGD_{BH} gives very good result. See Figure 4.18. However, the number of critical points is significantly lower when considering more eigenvectors.

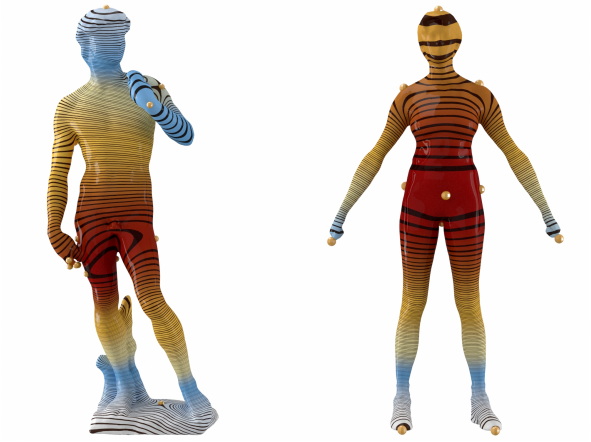


FIGURE 4.17. The field AGD_{BH} approximated by 6 eigenvectors.

FIGURE 4.18. The field AGD_{BH} approximated the Fiedler vector.

The average geodesic distance function is particularly a natural choice on manifolds that have some sort of symmetry into them [44]. We will talk in more details and give precise notion of what we mean by symmetry of a scalar field in section 4.9. See Figure 4.19.

4.9 Isometry Invariant Scalar Functions

The fact that the function AGD captures the symmetry in the geometry in Figure 4.19 is not a coincidence. The scalar function AGD is in fact an isometry invariant scalar function [44]. This explains the symmetry of this scalar functions. There are other scalar functions on Riemannian surfaces that have been proven to be isometry invariant. Examples are : Minimal Geodesic Distance function (MGD) [44], the

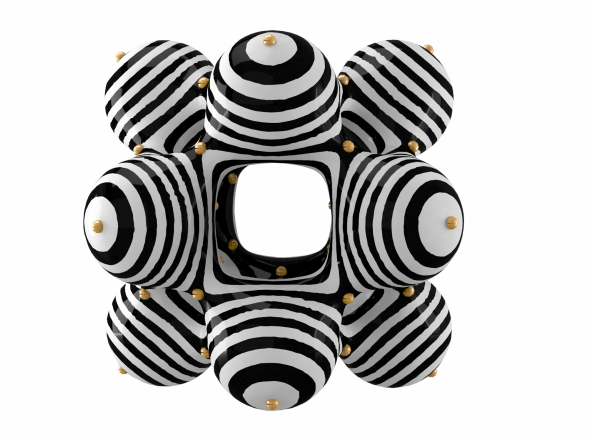


FIGURE 4.19. The field AGD_{BH} respects the geometry of the manifold.

Heat Kernel Signature (HKS) [86], the Wave Kernel Signature (WKS) [4], and the Auto Diffusion Function (ADF) [31]. Using on the spectral theory of the Laplace-Beltrami operator, Wang *et el* [93] proposed a set of isometry invariant functions that can be considered as a generalization for the signatures MGD, WKS and ADF. In this section we review all these functions and we show how to make write them to obtain a more robust results in regard of the distribution of their critical points. We also introduce the *Modified Heat Kernel Signature* and the *Modified Auto Diffusion Function*, which are parameterized families of isometry invariant scalar functions that generalize Heat Kernel Signature and Auto Diffusion Function respectively.

Definition 4.4. *Let M and N be a metric space with metrics d_M and d_N respectively. An onto map $T : M \longrightarrow N$ is called an isometry if it satisfies :*

$$d_M(p, q) = d_N(T(p), T(q)) \tag{4.17}$$

for all $p, q \in M$.

The set of self-isometries on a metric space M forms a group called the group of isometry. We will denote this group by $G(M)$. On the other hand, *an isometry between Two Riemannian manifold* (M, g) and (N, h) is a diffeomorphism T :

$M \longrightarrow N$ such that $T^*g = h$. If such a function T exists and if d_M and d_N are the distance functions induced by the Riemannian structures g and h respectively then $d_M(x, y) = d_N(T(x), T(y))$ and hence (M, d_M) and (N, d_N) are isometric as metric spaces. The following interesting theorem shows that the converse is also true [62].

Theorem 4.5. *(Myers-Steenrod) Let (M, g) and (N, h) be Riemannian manifolds and let d_M and d_N be the metric functions induced by the Riemannian metric g and h. If $T : M \longrightarrow N$ is an onto map that satisfies $d_M(p, q) = d_N(T(p), T(q))$, then T is a Riemanian isometry, it particular it is smooth.*

The previous theorem shows that group of Riemannian isoemtries on a Riemannian manifold (M, g) coincides with the group of isometries on the metric space (M, d) where d is the metric induced by the Riemannian metric g. For this reason we will not distinguish between these two groups. Now we define scalar functions that are invariant under the action of the group $G(M)$.

Definition 4.6. *Let M be a metric space and let $f : M \longrightarrow \mathbb{R}$ be a scalar function defined on M. A scalar function $f : M \longrightarrow \mathbb{R}$ is called isometry invariant over M if for every $T \in G(M)$*

$$f \circ T = f.$$

In the context of computer graphics such functions are studied in [65, 44]. The following simple properties can be proven easily.

1. Constant functions on a manifold are isometry invariant.
2. If f is isometry invariant and $c \in \mathbb{R}$ then $c.f$ is also an isometry invariant .
3. If f, g are isometry invariants then $f.g$ and $f/g (g \neq 0)$ are isometry invariant.

The previous properties shows that the set of all isometry invariant scalar functions on a manifold is an algebra over $\mathbb{R}$.

As we mentioned in the introduction of this section, there are other many scalar function in the literature that have been proven to be isometry invariant such as Minimal Geodesic Distance function (MGD) [44], the Heat Kernal Signature (HKS) [86], the Wave Kernel Signature (WKS) [4], and the Auto Diffusion Function (ADF) [31]. Of particular interest to us is the work of Wang *et el* [93] who recently proposed a family of isometry invariant functions, using the theory of Laplace-Beltrami operator. This isometry invariant family of scalar functions can be considered as a generalization for the functions AGD, HKS and ADF. We talk about this method briefly here.

4.9.1 Spectral Isometry Invariant Functions Via the Laplace-Beltrami Operator

On closed Riemmannain manifold the following theorem provides us with a countable family of isometry invariant functions [93].

Theorem 4.7. *Let (M, g) be closed Riemannian manifold. Let λ_i be an eigenvalue for the Laplace-Beltrami operator on M with a $k-$dimensional eigenfunction space. If $\{\phi_{ij}\}_{j=1}^{k}$ is an orthogonal basis of the corresponding eigenspace of λ_i then the scalar function defined by :*

$$f_i = \sum_{j=1}^{k} \phi_{ij}^2 \tag{4.18}$$

is isometry invariant on M.

See Figure 4.20 for an example of such scalar field on high genus surface.

Recall that HKS [86] is given by :

$$HKS_t = \sum_{i=1}^{\infty} e^{-\lambda_i t} \phi_j^2. \tag{4.19}$$

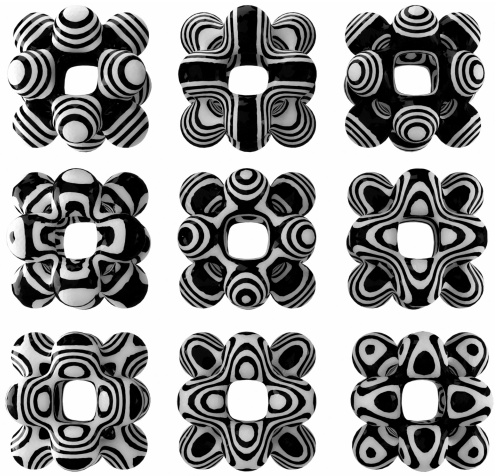


FIGURE 4.20. Isometry invariant fields obtained from theorem 4.7.

Here $t \in [0, \infty)$. Moreover, the ADF [31] is given by :

$$ADF_t = \sum_{i=1}^{\infty} e^{-t\frac{\lambda_i}{\lambda_1}} \phi_j^2 \tag{4.20}$$

Since the set of all isometry invariant scalar functions forms an algebra over $\mathbb{R}$ then by theorem 4.7 one can see that any linear combinations of the functions f_i described in theorem 4.7 is an isometry invariant over M. Hence, in particular, the functions ADF_t and HKS_t are isometry invariant. In our experimentation we have found that AGD_{BH} gives better result with respect to the symmetry of critical points than the ADF_t and HKS_t. Moreover, Poisson fields do not always give desirable scalar fields on symmetric high genus surfaces. See Figure 4.22 for a comparison between AGD_{BH}, $ADF_{0.5}$ approximated by 6 eigenvectors and the Poisson field on a high genus surface.

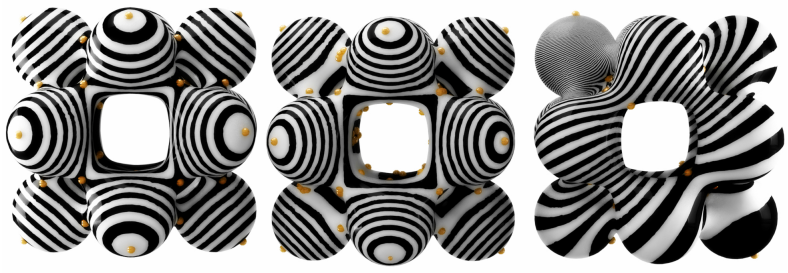


FIGURE 4.21. From left to right, AGD_{BH}, $ADF_{0.5}$ and a Poisson field.

In the next section explain why we have the problem in regard with the critical points of the scalar functions ADF and HKS in Figure4.22. We also give two isometry invariant families of scalar functions that generalize the functions ADF and HKS.

4.9.2 Modified Auto Diffusion and Heat Signature Functions

In practice, the computed eigenvalues for the Laplacian are very rarely exactly equal. For this reason we consider a sequence of eigenvalues $\lambda_i, ..., \lambda_{i+k}$ to be the same eigenvalue λ_i if the difference $|\lambda_i - \lambda_{i+j}| < \epsilon$ for all $1 < j \leq k$ and a sufficiently small ϵ. For each ϵ, this defines an equivalence relation on the set of eigenvalues of the Laplacian. Denote by $\bar{\lambda_i}$ to the equivalence class that contains $\lambda_i, ..., \lambda_{i+k}$. The value of the class $\bar{\lambda}_i$ is by definition λ_i. Denote by Λ to the set of equivalence classes of eigenvalues of the Laplacian. The eigenvalue classes of Λ can be ordered as $\bar{\lambda}_{i_1} < \bar{\lambda}_{i_2} < ... < \bar{\lambda_{i_n}} < ....$ For simplicity of the notation set $\eta_j = \bar{\lambda}_{i_j}$. In simple words, Λ is the set consisting of distinct ordered eigenvalues $\eta_1 < \eta_2 < .... < \eta_n < ....$ of the Laplacian.

Using this convention and the notation 4.18, we rewrite the equations 4.19 and 4.20 as follows :

$$HKS_t = \sum_{i=1}^{\infty} e^{-\eta_i t} f_i, \tag{4.21}$$

and

$$ADF_t = \sum_{i=1}^{\infty} e^{-t\frac{\eta_i}{\eta_1}} f_i. \tag{4.22}$$

While this seems just a different way to write the same function, but the fact that we combined terms in this manner has the the following practical advantage. In practice we usually approximate the summation 4.19 or similarly 4.20, with a k terms where $k < |\mathcal{V} - 1|$. However, this truncation is far from ideal since this truncated summation is not necessarily an isometry invariant scalar function. On

the other hand, approximating the summation 4.21, or similarly 4.22, with k terms preserves this property for any k. This explains the existence of non-symmetric critical points on an isometry invariant function in practice. See Figure 4.22 and notice how the critical points on of the Figure on the left are not symmetric with the rest of the critical points located in symmetric locations whereas this problem disappears in the right hand Figure. This problem is reduces significantly if one considers the versions 4.21 and 4.22 of the HKS and ADF.

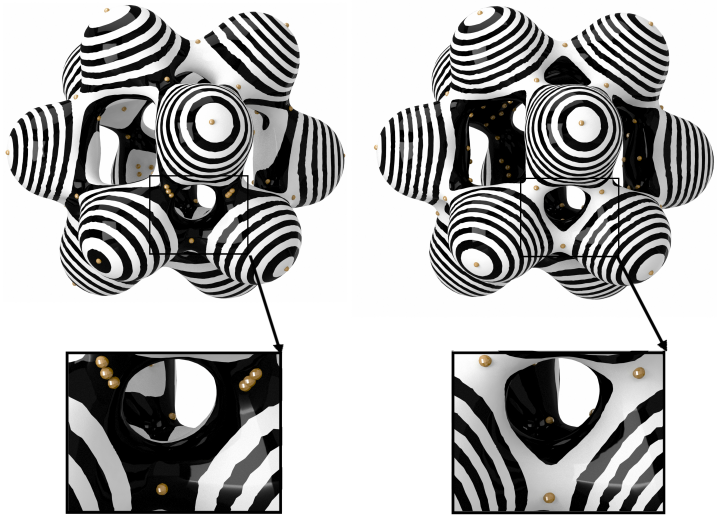


FIGURE 4.22. Naive approximation of HKS (left) and HKS using 4.21 (right)

We extend this idea further and we propose the following family of isometry invariant functions. The *Modified Heat Kernel Signature* is defined to be

$$MHKS_t(., j) = \sum_{i=j}^{\infty} e^{-\eta_i t} f_i, \tag{4.23}$$

and the *Modified Auto Diffusion Function* is defined by :

$$MADF_t(., j) = \sum_{i=j}^{\infty} e^{-t\frac{\eta_i}{\eta_1}} f_i. \tag{4.24}$$

where $j \geq 1$. The motivation of this definition is the following. One can notice from the original definition HKS and ADF that these two functions are largely

determined by the first few eigenfunctions. This however might not always be desirable since it diminishes all symmetry-related information coming from the higher terms. The summations 4.23 and 4.24 recover the information stored in the higher order terms of HKS and ADF. In other words, $\{MHKS_t(., j)\}_{j=1}^{\infty}$ gives a family of "filtered" isometry invariant functions of the original MHK_t. The same applies for $\{MADF_t(., j)\}_{j=1}^{\infty}$. See Figure 4.23 for an example of such functions.

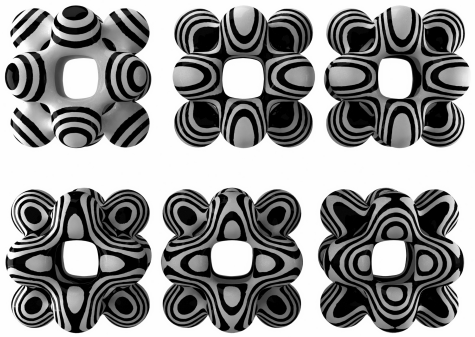


FIGURE 4.23. The isometry invariant functions $MHKS_{0.5}(., 2i)$ for $1 \leq i \leq 6$.

Chapter 5
Pants Decomposition Algorithms

5.1 Introduction

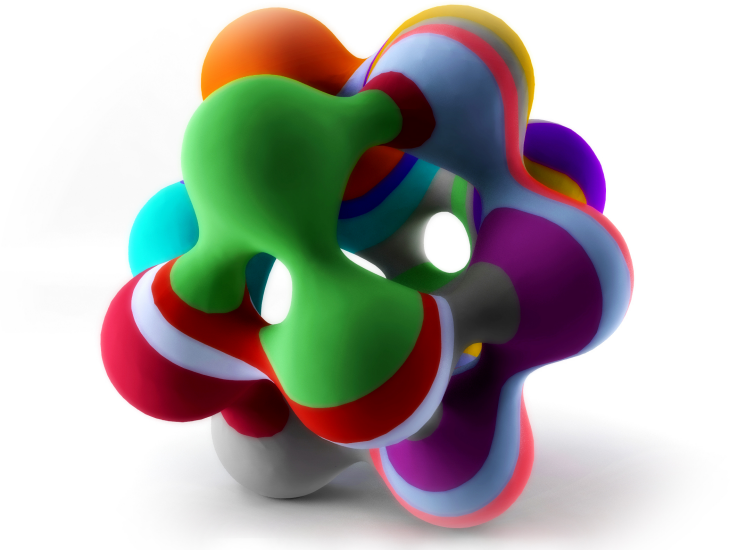


FIGURE 5.1. Pants decomposition obtained using one of our algorithms.

Mesh segmentation is a fundamental problem in geometric modeling. One often needs to divide a complicated mesh geometry into multiple pieces that are easier to process. Mesh segmentation has applications including texture mapping [49], collision detection [52], skeletonization [12] and three-dimensional shape retrieval [102].

Many segmentation algorithms have been proposed in the graphics literature includes [55, 78, 51, 14, 82, 2]. One reason for the variety of segmentation algorithms suggested in the literature is there that is no one universal good algorithm that suits all applications. The techniques used in algorithms are related to other areas in computer graphics such as image segmentation [73, 90] and machine learning [41, 17]. For good surveys on various segmentation algorithms see [3, 78]. In this

chapter we propose two algorithms to decompose a surface into topologically consistent pieces using Morse theory. More precisely, we give algorithms that segment the the surface into a collection of pants which are topological sphere with three disks removed. Similar algorithms to ours have been proposed by [51] and [100]. However, all the existing methods rely on computing certain loops on the surface called *Handles and Tunnels loops* and computing such curves is computationally expensive and more importantly it depends on the embedding of the surface. Using Morse theory makes our algorithm robust with respect to these two points : we obtain a decomposition that is independent of the embedding and the computation of the curves of the decomposition is a task that can be computed in linear time.

5.2 Pants Decomposition

Let M be a compact, orientable and connected surface. We say that M is of type (g, b) if M is of genus g and b zero boundary components. A pair of a pant is a surface of type $(0, 3)$. See Figure 5.2.

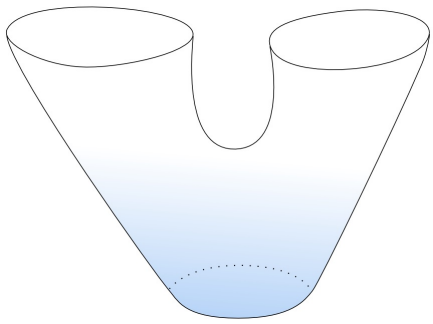


FIGURE 5.2. A pair of pant

A pants decomposition of M is a finite collection of unordered pairwise disjoint simple closed curves $\{c_1, ...c_n\}$ embedded in M with the property that $M - (c_1 \cup ... \cup c_n)$ is a disjoint union of pair of pants. Two pants decompositions of M are equivalent if they are isotopic. See Figure 5.3 for an example of two non-isotopic pants decompositions of a genus 2 surface.

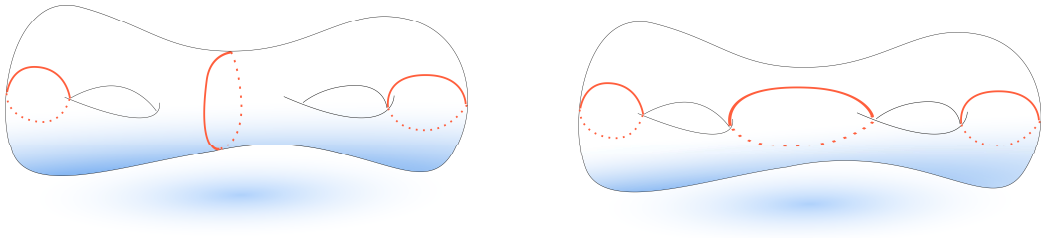


FIGURE 5.3. Two non-isotopic pants decompositions of a genus 2 surface.

Every connected, compact and orientable surface M with $\chi(M) < 0$, genus g and b boundary components admits a pants decomposition with $3g-3+b$ simple closed curves and the number of complementary components is $2g - 2 + n = |\chi(M)|$.

5.3 Morse Function-Based Algorithm to Compute a Pants Decomposition of a Surface

In this section we utilize Morse theory to give an algorithm to decompose a surface with $\chi(M) < 0$ into a collection of surfaces of type $(0, 3)$. Our algorithm works for arbitrary surface M with $\chi(M) < 0$ with and without a boundary. However, in order to guarantee the correctness of our algorithm we must choose a Morse function with certain properties. For the latter reason we divide the algorithm into two sub-cases. It should be noted however that the underlying idea of both cases is the same.

5.3.1 Pants Decomposition of Surface With $\chi(M) < 0$ and Without Boundary

In this section we give an algorithm to compute a pants decomposition of a surface with genus $g \geq 2$ without boundary. Let M be a compact connected orientable surface with genus $g \geq 2$ without boundary and let f be a Morse function on M. Suppose that $t_1, t_2, ..., t_n$ are the critical values for f ordered in an ascending order. Let $p_1, p_2, ..., p_n$ be the corresponding critical points of f. Choose a real number $\epsilon > 0$ small enough so that for each $1 \leq i \leq n$ there are no critical values for

f on the interval $[t_i - \epsilon, t_i + \epsilon]$ except t_i. Finally we assume that function f has exactly one minimum and exactly one maximum. It is clear from the choice of the function f that one of the points p_1 and p_n is the global maximum and one of them is the global minimum. Since multiplying any Morse function on a surface by -1 changes its critical points of index 0 to critical points of index 2 and vise versa, we can always choose our Morse function f so that p_1 is the global minimum and p_n is the global maximum. It should be noted that such a Morse function can be constructed in practice and we will talk about the construction of such functions later. We need the following lemma for the correctness of our algorithm.

Lemma 5.1. *Let M be a compact connected orientable surface with a Morse function f chosen as specified above then $M_{t_3+\epsilon}$ is diffeomorphic to a surface of type $(1,1)$ or a surface of type $(0,3)$.*

Proof. The choice of the scalar function f implies immediately that for each $2 \leq i \leq n-1$ we have $index_f(p_i) = 1$. Moreover, by construction we have $index_f(p_1) = 0$ and $index_f(p_n) = 2$. By Theorem 2.15 we conclude that $M_{t_0+\epsilon}$ is diffeomorphic to a disk and $M_{t_1+\epsilon}$ is diffeomorphic to surface of type $(0,2)$.

Moreover, by Theorem 2.15, when f passes from t_3 the surface $M_{t_3+\epsilon}$ is obtained from $M_{t_2+\epsilon}$ by gluing a rectangular strip $D^1 \times D^1$ to the boundary of $M_{t_2+\epsilon}$ along $D^1 \times \partial D^1$. Up to a homeomophism, there are two possible ways to glue the rectangular strip $D^1 \times D^1$ to the boundary of $M_{t_2+\epsilon}$ along $D^1 \times \partial D^1$. See Figure 5.4. We either glue this rectangular strip to the same boundary component of $M_{t_2+\epsilon}$ to obtain a surface of type $(0,3)$ or we glue each side of the strip on one of the boundary components to obtain a surface of type $(1,1)$. See Figure 5.5.

□

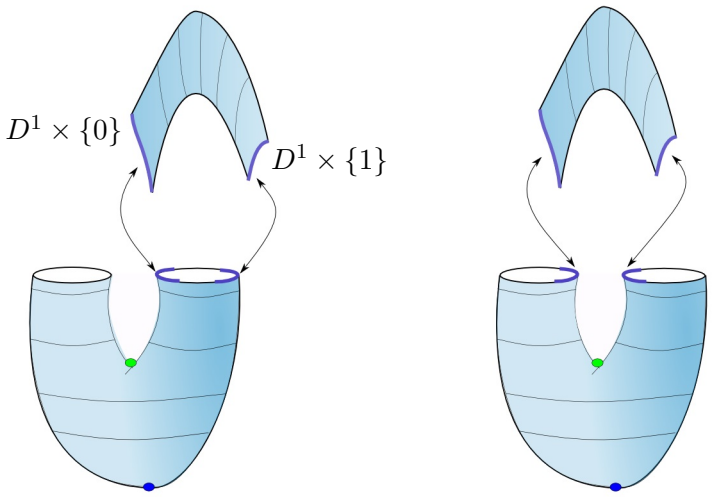


FIGURE 5.4. Two possible ways to glue a disk to the surface of type (0.2).

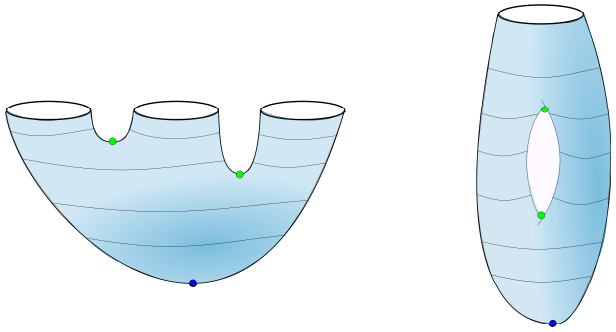


FIGURE 5.5. $M_{f,t_3+\epsilon}$ is diffeomorphic to a type $(0,3)$ or to a type $(1,1)$

.

Remark 5.2. *Using similar argument one can prove that the surface* $M_{t_{n-2}-\epsilon}$ *is either of type* $(0,3)$ *or a surface of type* $(1,1)$.

Lemma 5.1 and remark 5.2 will be used to obtain the first and the last pants in our pants decomposition. The algorithm is as follows:

1. Compute the critical points of f and put them in an ascending order. Let $p_1, p_2, ..., p_n$ be the sequence of ordered critical points of f and let $t_1, t_2, ..., t_n$ be their corresponding critical values. Note that $n = 2g + 2$ by our choice of the scalar function f.

2. For each $3 \leq i < n-3$ let $c_i = \frac{t_i + t_{i+1}}{2}$. After reindexing, we define the set $C = \{c_i | 1 \leq i \leq 2g-3\}$. In other words, the set C is a set of ordered regular values for f such that there is exactly one critical value for the function f in the intervals $[c_i, c_{i+1}]$ for $1 \leq i \leq 2g-3$. See Figure 5.6 for an example.

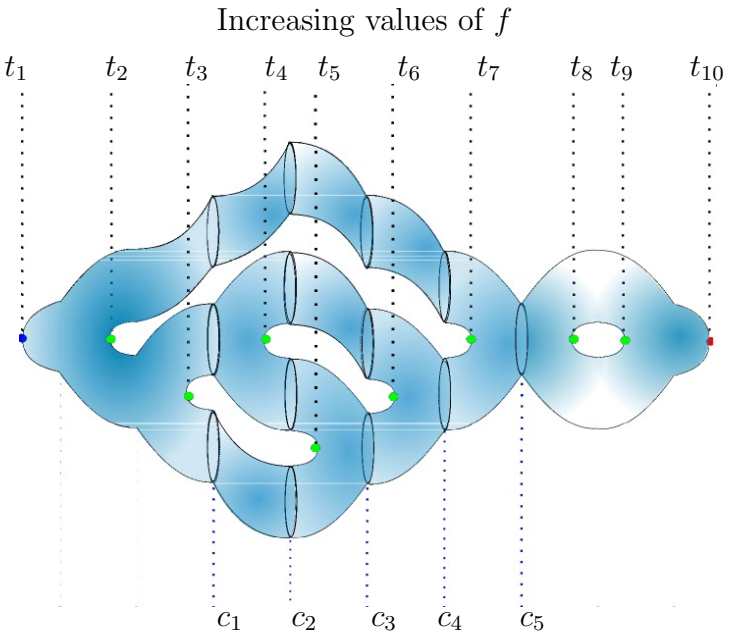


FIGURE 5.6. Cutting a surface of genus 4 along the values c_i.

3. Cut the surface M along the level sets $f^{-1}(c)$ for all $c \in C$. Note again that the values $c \in C$ are all regular values.
4. By Lemma 5.1 the surface M_{c_1} is either if type $(1,1)$ or of type $(0,3)$. If M_{c_1} is of type $(1,1)$, then we use $grad(f)$ to trace a loop from the saddle point p_2 to the minimum point p_1 and then we cut the surface along that loop. See Figure 5.7. Otherwise, if the surface is of type $(0,3)$ we do not need to do anything. In either cases, we denote the resulting surface of type $(0,3)$ by $M_{initial}$.
5. Consider the manifold with boundary $M_{[c_1,c_2]}$. This surface is a finite disjoint union of one surface of type $(0,3)$ and multiple surfaces of type $(0,2)$. Attach

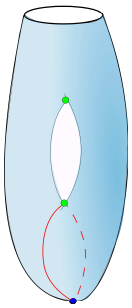


FIGURE 5.7. Tracing a loop from the first saddle point p_2 to the minimum p_1.

every surface of type $(0,2)$ to $M_{initial}$. Note that this gluing does not change the homeomorphism type of $M_{initial}$. See Figure 5.8 (b).

6. For each $1 < i < 2g-3$ consider the manifold with boundary $M_{[c_i,c_{i+1}]}$. This surface is again a finite disjoint union of one surface of type $(0,3)$ and multiple surfaces of type $(0,2)$. Attach every surface of type $(0,2)$ to $M_{[c_{i-1},c_i]}$. See Figure 5.8 (c).

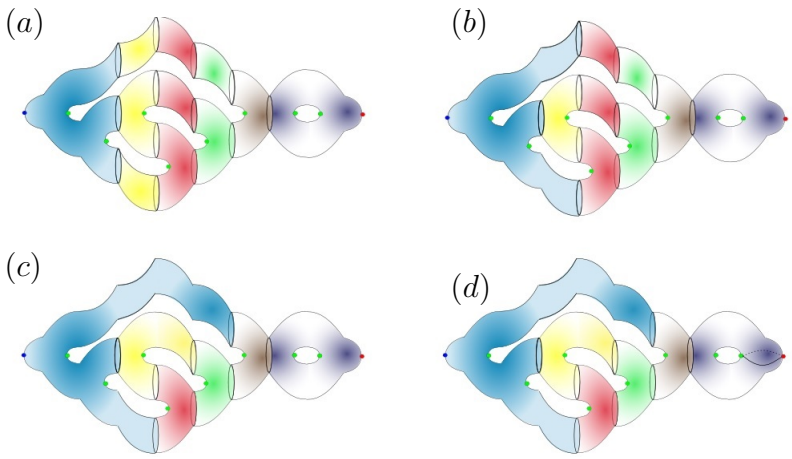


FIGURE 5.8. Illustration of the algorithm

7. The remaining part $M_{-f,c_{2g-3}}$ is either of type $(1,1)$ or of type $(0,3)$. If the surface $M_{-f,c_{2g-3}}$ is of type $(1,1)$ then use $grad(-f)$ to trace a loop from the saddle point p_{n-1} to the point p_n and this cut the surface $M_{-f,c_{2g-3}}$ along this loop to obtain a pant. Otherwise, if the surface is of type $(0,3)$ we do not need to do anything.

Note that algorithm constructs a collection of pants inductively. We start by having the the first pair of pants in 4 and then we go to the next level which is a finite disjoint union of single pair of pants and some topological cylinders. We attach the cylinders to the previous pair of pants and then we go to the next level and repeat the same process.

5.3.2 Pants Decomposition of Surface With $\chi(M) < 0$ and With Boundary

In this section we present an algorithm to decompose a surface M with $\chi(M) < 0$ with boundary components. The algorithm follows almost as before. The main difference is that we need to take care of the choice of the Morse function. We have two cases, (1) The surface M has exactly one boundary component (2) The surface M has more than one boundary components.

Let M be a compact connected orientable surface with $\chi(M) < 0$ and one boundary component Σ. We pick a point p_{max} on the surface and construct a Morse function $f : M \longrightarrow [0, 1]$ that satisfies the following conditions :

1. $f^{-1}(\Sigma) = 0$ and $f(x) > 0$ for all x in $M \backslash \Sigma$.
2. The point $f^{-1}(1) = x_{max}$ is a global maximum.
3. The function f does not have any critical point of index 0 or 2 except for p_{max}.

The pants decomposition algorithm for a surface M with $\chi(M) < 0$ and one boundary component Σ algorithm goes now as follows :

1. Compute the critical points of f and put them in an ascending order. Let $p_1, p_2, ..., p_n$ be the ordered critical points of f and let $t_1, t_2, ..., t_n$ be the corresponding critical values of f. By our choice of Morse function we $index(p_i) = 1$ for all $1 \leq i \leq n - 1$ and $index(p_{max}) = 2$.

2. For each $1 \leq i < n-3$ let $c_i = \frac{t_i + t_{i+1}}{2}$. Define the set $C = \{c_i | 1 \leq i \leq 2g-3\}$. In other words, the set C is a set of ordered regular values for f such that there is exactly one critical value in the interval $[c_i, c_{i+1}]$. See Figure 5.9 for an example.
3. Cut the surface M along the level sets $f^{-1}(c)$ for all $c \in C$.
4. Consider the manifold $M_{initial} := M_{[-\epsilon, c_1]}$. By our choice of the Morse function $M_{initial}$ is a of type $(0, 3)$.

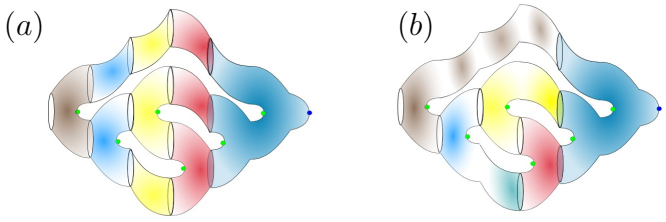


FIGURE 5.9. Pants decomposition for a surface with a single boundary component

The rest of the pant decomposition algorithm for a surface is similar to steps (5), (6) and (7) of the algorithm in section 5.3.1.

Now we discuss the final case. Suppose that M has boundary components $\Sigma_1, ..., \Sigma_k$ where $k \geq 2$. Here we also need to construct a Morse function that serves our purpose. We need a Morse function f that satisfies the following :

1. We choose one of the boundary component, say Σ_1, and we construct f such that $f^{-1}(0) = \Sigma_1$ and $f(x) > 0$ for all $x \in M \backslash \Sigma_1$.
2. $f^{-1}(1) = \cup_{i=2}^{k} \Sigma_i$.
3. The function f does not have any critical point of index 0 or 2.

As we did earlier, let $p_1, p_2, ..., p_n$ be the ordered critical points of f and let $t_1, t_2, ..., t_n$ be the corresponding critical values of f. Define the values $c_i = \frac{t_i + t_{i+1}}{2}$ for all $1 \leq t_i \leq n-1$. By our choice of the Morse function, the manifold $M_{[c_i, c_{i+1}]}$ is

homeomorphic to a finite disjoint union of one surface of type $(0, 3)$ and multiple surfaces of type $(0, 2)$. In particular $M_{[-\epsilon,c_1]}$ is a pant. The pants decomposition algorithm now is similar to the previous algorithm except that we do not to trace any loop since all connected components after cutting along the regular values of C are pants or cylinders. See Figure 5.10 for an example.

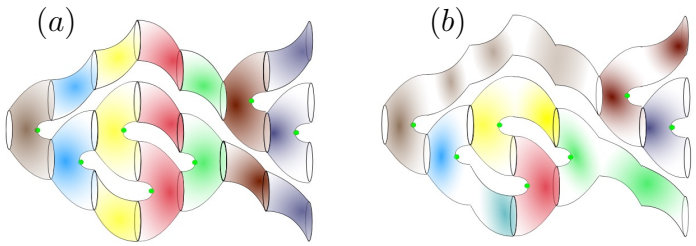


FIGURE 5.10. Pants decomposition for a surface with multiple boundary components

Figure 5.11 shows multiple examples of pants decomposition of surfaces using this algorithm.

FIGURE 5.11. Pants decompositions using our Morse Function-Based algorithm.

5.4 Reeb Graph-Based Pants Decomposition Algorithm for an Arbitrary Surface

In this section we give n pants decomposition algorithm using the Reeb graph of a Morse function. This algorithm has many advantages over the previous algorithm. The main advantage lies in the fact that it does not put any restriction on choice of

the Morse function. This allows us to choose a scalar function with better geometric properties. The second advantage is that choosing a cutting circle on a surface is much more flexible when using the structure of the Reeb graph than choosing the inverse image of a regular value of a Morse function.

Let M be a compact orientable surface, possibly with boundary, such that $\chi(M) < 0$. Let f be an arbitrary Morse function of M. The Reeb graph-based pants decomposition algorithm of the surface M and the Morse function f goes as follows:

1. Compute the Reeb graph $R(f)$ of (M, f).
2. There are two types of 1-valence nodes on the graph $R(f)$, the 1-valence nodes that are a result of collapsing the boundary components of M and the 1-valence nodes that are coming from critical points of f of index 0 and 2. We consider the graph $\overline{R(f)}$ obtained by taking the deformation retract of the graph $R(f)$ that leave the edges coming from the boundary component without retraction. This step can be done iteratively by keep deleting the edges whose one of its defining nodes has valence 1 until there are no more such edges except for the ones which have 1-valence nodes originated from boundary component. See Figure 5.12 step (3) for an example of such retraction.
3. We remove all the nodes on the graph $\overline{R(f)}$ of valency 2 and we combine the two edges that meet at such a node into one edge. We also denote the obtained from this step by $\overline{R(f)}$. Note that this graph is trivalent by construction.
4. We select an interior point on every edge of the graph $\overline{R(f)}$ provided that neither one of the two nodes defining that edge has valency 1. Note that the selection of the these point on the graph $\overline{R(f)}$ corresponds to partitioning

the graph $\overline{R(f)}$ into a collection of small graphs each one of them is a vertex connected by small three arcs. See step (5) in Figure 5.12. Each one of these small graphs corresponds to a pant on the surface M.

5. Each choice of an interior point induces a choice of a simple closed curve on the original surface M. The collection of all curves obtained in this way defines a pants decomposition on the surface M.

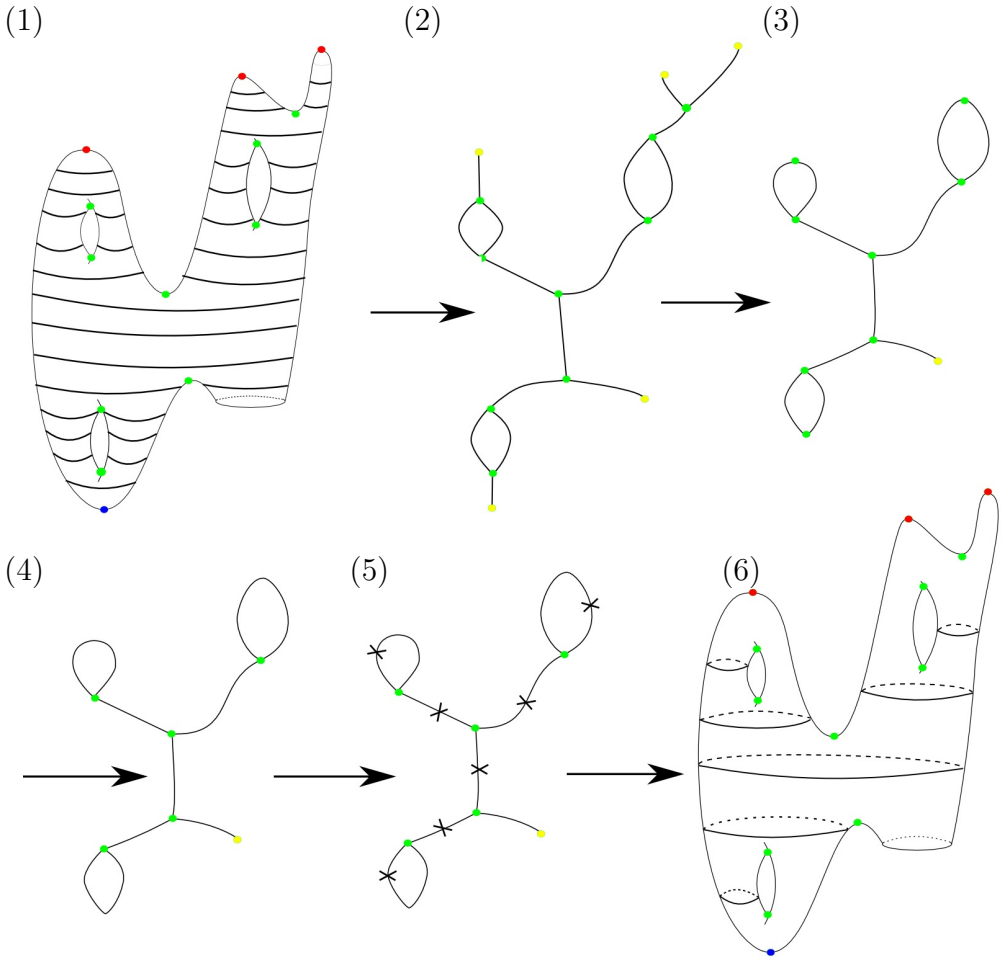


FIGURE 5.12. The steps of the Reeb graph-based pants decomposition algorithm

Figure 5.13 shows examples of application of this algorithm on some surfaces.

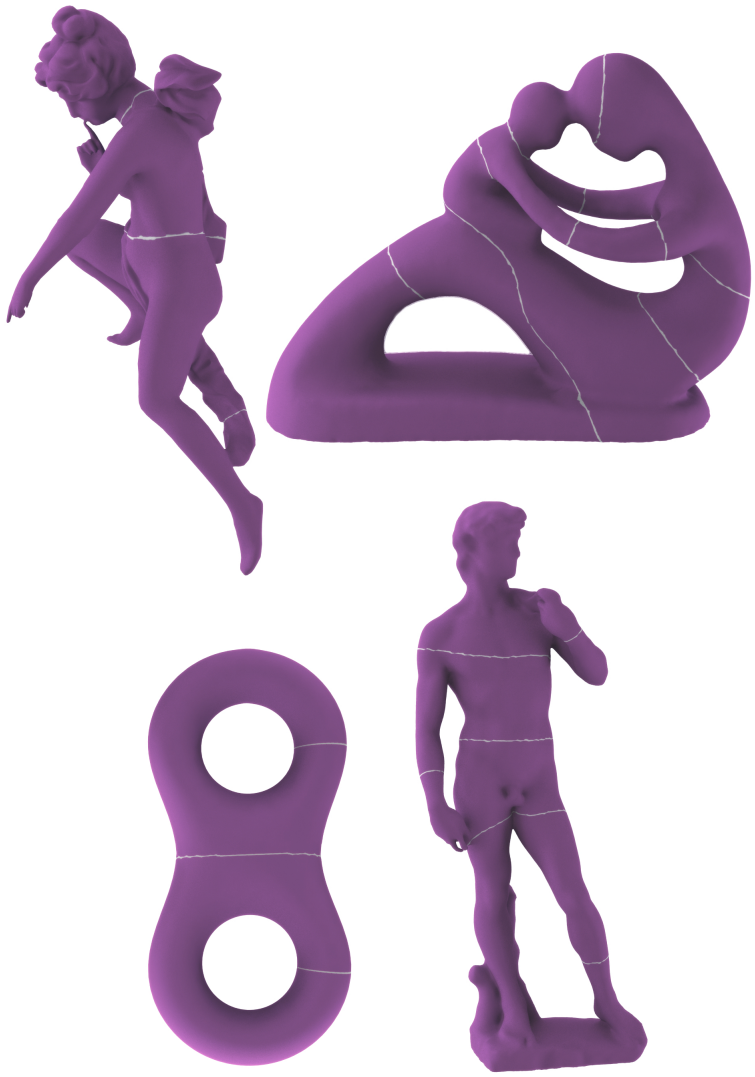


FIGURE 5.13. Examples of our second algorithm

5.4.1 Further Directions

Let $M \subset \mathbb{R}^3$ be a connected, closed, orientable and embedded surface of genus g in $\mathbb{R}^3$ and let f be a Morse function on M, then M can be recovered up to a homeomorphism from $R(f)$ as the boundary of an oriented 3-dimensional regular neighborhood of the graph $R(f)$. In other words, all topological data of M can be recovered from the Reeb graph $R(f)$. This fact is the essence of the Reeb graph-based pants decomposition algorithm 5.4. In fact, $R(f)$ in this case can also be used to recover embedding-related information about the surface M as we will

demonstrate here. The surface M, with the properties described above, determines a partition for $\mathbb{R}^3$ into an "inside" region $\mathbb{I}$ and an "outside" region $\mathbb{O}$. The region $\mathbb{I}$ is the bounded region in $\mathbb{R}^3$ determined by M, and $\mathbb{O}$ is the unbounaded region in $\mathbb{R}^3$ determined by M. We take both $\mathbb{I}$ and $\mathbb{O}$ to be 3-manifold with boundary and their boundary is the surface M. A simple closed curve on M, or a loop, is called *a handle loop* if it is trivial in homology group $H_1(\mathbb{I})$ and non-trivial in the homology group $H_1(\mathbb{O})$. For a connected, closed, orientable and embedded surface of genus g there are g handle loops [59].

Determining these loops and their dual, called tunnel loops, is an important problem in computer graphics and many algorithms have been suggested to compute them [20, 21, 19]. Handle loops can be easily computed using Reeb graph as follows [34]. Let T be a spanning tree of $R(f)$ and consider the edges $e_1, ...e_g$ in $R(f)\backslash T$. Each interior point in e_i, for $0 \leq i \leq g$, determines a handle loop l_i under the surjective map $f : M \longrightarrow R(f)$. Similar method was employed by [19] to determine handle loops. Choosing an interior point for each e_i, where $0 \leq i \leq g$, determines a finite collection of curves $l_1, ..., l_g$ and these curves determine a *cutting system* [34]. A cutting system for a connected, closed, orientable and embedded surface M of genus g is a collection of unordered disjoint simple closed loops whose complement $M\backslash(l_1 \sqcup ... \sqcup l_g)$ is a sphere with $2g$ boundary components [34]. One can see that these facts immediately yield an algorithm to obtain the a cutting system for M.

5.5 Choosing a Morse Function

We need to construct a Morse function that suits the algorithms that we have presented. For the algorithm given in section 5.3.1 we need a scalar function that

has one global minimum and one local maximum. This can be done by solving a Laplace equation on a mesh with Dirichlet boundary condition. For more details see Chapter 3 section 4.5. Using the notation we presented in 4.5, the constrained vertices V_C is chosen to have exactly two vertices $V_C = \{v_{min}, v_{max}\}$ such that $f(v_{min}) < f(v_{max})$. This choice will guarantee that the solution f has a single minimum at v_{min} and a single maximum at v_{max}. Hence, every other critical point for f must be a saddle point which is an essential for our algorithm 5.3.1.

Algorithm 5.3.1 also requires choosing appropriate Morse functions when the surface M has one or more boundary components. Recall in the case when M has one boundary component we require that the Morse function has only one global maximum and the entire boundary component is mapped to the lowest value of the Morse function. Such a function can also be obtained in practice by using harmonic function with Dirichlet conditions chosen precisely as specified in the algorithm. Similarly, a harmonic scalar function can be used to obtain for the pants decomposition algorithm 5.3.1 for a surface with multiple boundary components.

Even though our second algorithm works on a generic Morse function, we choose a Morse function that captures the geometry and the symmetry aspects of the mesh. This was not possible in the previous algorithm due to the restriction of the input function. Our choice for scalar functions were made based on the object itself. For organic objects like the human body Poisson fields, the modified discrete conformal factor and AGD presented in the previous chapter gave us the best results. On the other hand, for objects with some symmetry, we found that isometry invariant scalar functions gave the best results.

5.6 Conclusion

Morse theory is a powerful mathematical tool that uses the local differential properties of a manifold to make conclusions about global topological aspects of the manifold. We have given two algorithms that utilizes Morse theory to decompose a 2-manifold into topological pants.

Although the choice of a Morse function has an impact on the output of the pants decomposition algorithms that we propose here, the topological resections that every patch in the segmentation must be a topological pant prevents our algorithms from being able in every case to segment every mesh into geometrically desired sub-patches.

5.7 Acknowledgment

This research was funded partially by NSF grant IIS-1320959. We would like to thank Aim@Shape for providing the models used for our experimentations.

References

[1] M. Attene, S. Biasotti, and M. Spagnuolo. Shape understanding by contour-driven retiling. *The Visual Computer*, 19(2):127–138, 2003.

[2] M. Attene, B. Falcidieno, and M. Spagnuolo. Hierarchical mesh segmentation based on fitting primitives. *The Visual Computer*, 22(3):181–193, 2006.

[3] M. Attene, S. Katz, M. Mortara, G. Patané, M. Spagnuolo, and A. Tal. Mesh segmentation-a comparative study. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 7–7. IEEE, 2006.

[4] M. Aubry, U. Schlickewei, and D. Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1626–1633. IEEE, 2011.

[5] G. Aujay, F. Hétroy, F. Lazarus, and C. Depraz. Harmonic skeleton for realistic character animation. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 151–160. Eurographics Association, 2007.

[6] T. Banchoff et al. Critical points and curvature for embedded polyhedra. *J. Diff. Geom*, 1(245-256):34, 1967.

[7] A. Banyaga and D. Hurtubise. *Lectures on Morse homology*, volume 29. Springer Science & Business Media, 2013.

[8] M. Ben-Chen and C. Gotsman. Characterizing shape using conformal factors. In *3DOR*, pages 1–8, 2008.

[9] M. Ben-Chen, C. Gotsman, and G. Bunin. Conformal flattening by curvature prescription and metric scaling. In *Computer Graphics Forum*, volume 27, pages 449–458. Citeseer, 2008.

[10] M. Berger. *A panoramic view of Riemannian geometry*. Springer Science & Business Media, 2003.

[11] S. Biasotti, B. Falcidieno, and M. Spagnuolo. Extended reeb graphs for surface understanding and description. In *Discrete geometry for computer imagery*, pages 185–197. Springer, 2000.

[12] S. Biasotti, S. Marini, M. Mortara, et al. An overview on properties and efficacy of topological skeletons in shape modelling. In *null*, page 245. IEEE, 2003.

[13] S. Biasotti, G. Patanè, M. Spagnuolo, B. Falcidieno, and G. Barequet. Shape approximation by differential properties of scalar functions. *Computers & Graphics*, 34(3):252–262, 2010.

[14] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3d mesh segmentation. In *ACM Transactions on Graphics (TOG)*, volume 28, page 73. ACM, 2009.

[15] Y. T. P. A. D. Cohen and S. M. Desbrun. Designing quadrangulations with discrete harmonic forms. In *Eurographics symposium on geometry processing*, pages 1–10, 2006.

[16] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Loops in reeb graphs of 2-manifolds. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 344–350. ACM, 2003.

[17] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

[18] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324. ACM Press/Addison-Wesley Publishing Co., 1999.

[19] T. K. Dey, F. Fan, and Y. Wang. An efficient computation of handle and tunnel loops via reeb graphs. *ACM Transactions on Graphics (TOG)*, 32(4):32, 2013.

[20] T. K. Dey, K. Li, and J. Sun. On computing handle and tunnel loops. In *Cyberworlds, 2007. CW'07. International Conference on*, pages 357–366. IEEE, 2007.

[21] T. K. Dey, K. Li, J. Sun, and D. Cohen-Steiner. Computing geometry-aware handle and tunnel loops in 3d models. In *ACM Transactions on Graphics (TOG)*, volume 27, page 45. ACM, 2008.

[22] S. Dong, P.-T. Bremer, M. Garland, V. Pascucci, and J. C. Hart. Spectral surface quadrangulation. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 1057–1066. ACM, 2006.

[23] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer aided geometric design*, 22(5):392–423, 2005.

[24] H. Doraiswamy and V. Natarajan. Efficient algorithms for computing reeb graphs. *Computational Geometry*, 42(6):606–616, 2009.

[25] T. Duchamp, A. Certain, A. DeRose, and W. Stuetzle. Hierarchical computation of pl harmonic embeddings. *preprint*, 1997.

[26] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28(4):511–533, 2002.

[27] H. Edelsbrunner, D. Morozov, and V. Pascucci. Persistence-sensitive simplification functions on 2-manifolds. In *Proceedings of the twenty-second annual symposium on Computational geometry*, pages 127–134. ACM, 2006.

[28] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[29] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its application to graph theory. *Czechoslovak Mathematical Journal*, 25(4):619–633, 1975.

[30] M. S. Floater. Mean value coordinates. *Computer aided geometric design*, 20(1):19–27, 2003.

[31] K. G?bal, J. A. Bærentzen, H. Aanæs, and R. Larsen. Shape analysis using the auto diffusion function. In *Computer Graphics Forum*, volume 28, pages 1405–1413. Wiley Online Library, 2009.

[32] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. In *ACM Transactions on Graphics (TOG)*, volume 22, pages 358–363. ACM, 2003.

[33] X. Guo, X. Li, Y. Bao, X. Gu, and H. Qin. Meshless thin-shell simulation based on global conformal parameterization. *Visualization and Computer Graphics, IEEE Transactions on*, 12(3):375–385, 2006.

[34] A. Hatcher and W. Thurston. A presentation for the mapping class group of a closed orientable surface. *Topology*, 19(3):221–237, 1980.

[35] F. Hétroy and D. Attali. Topological quadrangulations of closed triangulated surfaces using the reeb graph. *Graphical Models*, 65(1):131–148, 2003.

[36] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 203–212. ACM, 2001.

[37] M. Isenburg and P. Lindstrom. Streaming meshes. In *Visualization, 2005. VIS 05. IEEE*, pages 231–238. IEEE, 2005.

[38] V. Jain and H. Zhang. Robust 3d shape correspondence in the spectral domain. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 19–19. IEEE, 2006.

[39] T. Kanai and H. Suzuki. Approximate shortest path on a polyhedral surface based on selective refinement of the discrete graph and its applications. In *Geometric Modeling and Processing 2000. Theory and Applications. Proceedings*, pages 241–250. IEEE, 2000.

[40] P. Kanongchaiyos and Y. Shinagawa. Articulated reeb graphs for interactive skeleton animation. *Proceeding Modeling Multimedia Information and System*, pages 451–467, 2000.

[41] G. Karypis and V. Kumar. A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices. *University of Minnesota, Department of Computer Science and Engineering, Army HPC Research Center, Minneapolis, MN*, 1998.

[42] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. *The Visual Computer*, 21(8-10):649–658, 2005.

[43] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, 2006.

[44] V. G. Kim, Y. Lipman, X. Chen, and T. Funkhouser. Möbius transformations for global intrinsic symmetry analysis. In *Computer Graphics Forum*, volume 29, pages 1689–1700. Wiley Online Library, 2010.

[45] M. Lanthier, A. Maheshwari, and J.-R. Sack. Approximating weighted shortest paths on polyhedral surfaces. In *Proceedings of the thirteenth annual symposium on Computational geometry*, pages 274–283. ACM, 1997.

[46] J. Lee. *Introduction to topological manifolds*, volume 940. Springer Science & Business Media, 2010.

[47] J. M. Lee. *Smooth manifolds*. Springer, 2003.

[48] B. Levy. Laplace-beltrami eigenfunctions towards an algorithm that" understands" geometry. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 13–13. IEEE, 2006.

[49] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 362–371. ACM, 2002.

[50] B. Lévy and R. H. Zhang. Spectral geometry processing. 2009.

[51] X. Li, X. Gu, and H. Qin. Surface mapping using consistent pants decomposition. *Visualization and Computer Graphics, IEEE Transactions on*, 15(4):558–571, 2009.

[52] X. Li, T. W. Woon, T. S. Tan, and Z. Huang. Decomposing polygon meshes for interactive applications. In *Proceedings of the 2001 symposium on Interactive 3D graphics*, pages 35–42. ACM, 2001.

[53] Y. Lipman, R. M. Rustamov, and T. A. Funkhouser. Biharmonic distance. *ACM Transactions on Graphics (TOG)*, 29(3):27, 2010.

[54] Y.-S. Liu, M. Liu, D. Kihara, and K. Ramani. Salient critical points for meshes. In *Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 277–282. ACM, 2007.

[55] A. P. Mangan and R. T. Whitaker. Partitioning 3d surface meshes using watershed segmentation. *Visualization and Computer Graphics, IEEE Transactions on*, 5(4):308–321, 1999.

[56] Y. Matsumoto. *An introduction to Morse theory*, volume 208. American Mathematical Soc., 2002.

[57] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.

[58] J. W. Milnor. *Morse theory*. Number 51. Princeton university press, 1963.

[59] E. E. Moise. *Geometric topology in dimensions 2 and 3*, volume 47. Springer Science & Business Media, 2013.

[60] M. Morse. *The calculus of variations in the large*, volume 18. American Mathematical Soc., 1934.

[61] P. Mullen, Y. Tong, P. Alliez, and M. Desbrun. Spectral conformal parameterization. In *Computer Graphics Forum*, volume 27, pages 1487–1494. Wiley Online Library, 2008.

[62] S. B. Myers and N. Steenrod. The group of isometries of a riemannian manifold. *Annals of Mathematics*, pages 400–416, 1939.

[63] X. Ni, M. Garland, and J. C. Hart. Fair morse functions for extracting the topological structure of a surface mesh. *ACM Transactions on Graphics (TOG)*, 23(3):613–622, 2004.

[64] M. Ovsjanikov, Q. Mérigot, F. Mémoli, and L. Guibas. One point isometric matching with the heat kernel. In *Computer Graphics Forum*, volume 29, pages 1555–1564. Wiley Online Library, 2010.

[65] M. Ovsjanikov, J. Sun, and L. Guibas. Global intrinsic symmetries of shapes. In *Computer graphics forum*, volume 27, pages 1341–1348. Wiley Online Library, 2008.

[66] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas. Robust on-line computation of reeb graphs: simplicity and speed. In *ACM Transactions on Graphics (TOG)*, volume 26, page 58. ACM, 2007.

[67] G. Patane, M. Spagnuolo, and B. Falcidieno. Para-graph: Graph-based parameterization of triangle meshes with arbitrary genus. In *Computer Graphics Forum*, volume 23, pages 783–797. Wiley Online Library, 2004.

[68] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. *Experimental mathematics*, 2(1):15–36, 1993.

[69] K. Polthier et al. *Polyhedral surfaces of constant mean curvature*. PhD thesis, Habilitationsschrift TU Berlin, 2002.

[70] G. Reeb. Sur les points singuliers dune forme de pfaff completement intégrable ou dune fonction numérique. *CR Acad. Sci. Paris*, 222:847–849, 1946.

[71] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo. Discrete laplace–beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3):381–390, 2009.

[72] M. Reuter, F.-E. Wolter, and N. Peinecke. Laplace–beltrami spectra as shape-dnaof surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.

[73] D. Reynolds, R. C. Rose, et al. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995.

[74] S. Rosenberg. *The Laplacian on a Riemannian manifold: an introduction to analysis on manifolds*. Number 31. Cambridge University Press, 1997.

[75] J. Sahner, B. Weber, S. Prohaska, and H. Lamecker. Extraction of feature lines on surface meshes based on discrete morse theory. In *Computer Graphics Forum*, volume 27, pages 735–742. Wiley Online Library, 2008.

[76] H. Sato, T. Takino, Y. Okada, J. Cao, A. Shinagawa, E. Yamamoto, and M. Seiki. A matrix metalloproteinase expressed on the surface of invasive tumour cells. *Nature*, 370(6484):61–65, 1994.

[77] S. Schaefer and C. Yuksel. Example-based skeleton extraction. In *Symposium on Geometry Processing*, pages 153–162. Citeseer, 2007.

[78] A. Shamir. A survey on mesh segmentation techniques. In *Computer graphics forum*, volume 27, pages 1539–1556. Wiley Online Library, 2008.

[79] L. Shapira, A. Shamir, and D. Cohen-Or. Consistent mesh partitioning and skeletonisation using the shape diameter function. *The Visual Computer*, 24(4):249–259, 2008.

[80] M. Sharir and A. Schorr. On shortest paths in polyhedral spaces. *SIAM Journal on Computing*, 15(1):193–215, 1986.

[81] Y. Shinagawa and T. L. Kunii. Constructing a reeb graph automatically from cross sections. *IEEE Computer Graphics and Applications*, (6):44–51, 1991.

[82] S. Shlafman, A. Tal, and S. Katz. Metamorphosis of polyhedral surfaces using decomposition. In *Computer Graphics Forum*, volume 21, pages 219–228. Wiley Online Library, 2002.

[83] O. Sorkine. *Laplacian mesh processing*. PhD thesis, Citeseer, 2006.

[84] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, pages 175–184. ACM, 2004.

[85] B. T. Stander and J. C. Hart. Guaranteeing the topology of an implicit surface polygonization for interactive modeling. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 279–286. ACM Press/Addison-Wesley Publishing Co., 1997.

[86] J. Sun, M. Ovsjanikov, and L. Guibas. A concise and provably informative multi-scale signature based on heat diffusion. In *Computer graphics forum*, volume 28, pages 1383–1392. Wiley Online Library, 2009.

[87] G. Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358. ACM, 1995.

[88] J. Tierny, D. Günther, and V. Pascucci. Optimal general simplification of scalar fields on surfaces. In *Topological and Statistical Methods for Complex Data*, pages 57–71. Springer, 2015.

[89] J. Tierny and V. Pascucci. Generalized topological simplification of scalar fields on surfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2005–2013, 2012.

[90] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[91] B. Vallet and B. Lévy. Spectral geometry processing with manifold harmonics. In *Computer Graphics Forum*, volume 27, pages 251–260. Wiley Online Library, 2008.

[92] H. Wang, Y. He, X. Li, X. Gu, and H. Qin. Geometry-aware domain decomposition for t-spline-based manifold modeling. *Computers & Graphics*, 33(3):359–368, 2009.

[93] H. Wang, P. Simari, Z. Su, and H. Zhang. Spectral global intrinsic symmetry invariant functions. In *Proceedings of the 2014 Graphics Interface Conference*, pages 209–215. Canadian Information Processing Society, 2014.

[94] M. Wardetzky, S. Mathur, F. Kälberer, and E. Grinspun. Discrete laplace operators: no free lunch. In *Symposium on Geometry processing*, pages 33–37, 2007.

[95] N. Werghi, Y. Xiao, and J. P. Siebert. A functional-based segmentation of human body scans in arbitrary postures. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 36(1):153–165, 2006.

[96] I. Yamazaki, V. Natarajan, Z. Bai, and B. Hamann. Segmenting point sets. In *Shape Modeling and Applications, 2006. SMI 2006. IEEE International Conference on*, pages 6–6. IEEE, 2006.

[97] L. Yen, F. Fouss, C. Decaestecker, P. Francq, and M. Saerens. Graph nodes clustering based on the commute-time kernel. In *Advances in Knowledge Discovery and Data Mining*, pages 1037–1045. Springer, 2007.

[98] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with poisson-based gradient field manipulation. *ACM Transactions on Graphics (TOG)*, 23(3):644–651, 2004.

[99] E. Zhang, K. Mischaikow, and G. Turk. Feature-based surface parameterization and texture mapping. *ACM Transactions on Graphics (TOG)*, 24(1):1–27, 2005.

[100] K. Zhang and X. Li. Optimizing geometry-aware pants decomposition. In *Proc. Pacific Graphics*, pages 11–16, 2012.

[101] W. Zhao, S. Gao, and H. Lin. A robust hole-filling algorithm for triangular mesh. *The Visual Computer*, 23(12):987–997, 2007.

[102] E. Zuckerberger, A. Tal, and S. Shlafman. Polyhedral surface decomposition with applications. *Computers & Graphics*, 26(5):733–743, 2002.

Vita

Mustafa Hajij was born in Damascus City, Syria. He finished his undergraduate studies at Damascus University in 2004. He earned a master of science degree in mathematics from Jordan University for Science and Technology in 2008. In August 2008 he came to Louisiana State University to pursue graduate studies in mathematics. He is currently a candidate for the degree of Doctor of Philosophy in mathematics and the degree of Master of Science in computer sciences, which will be awarded in August 2015.