A Shape Mapping Framework for Graphics and Visual Computing

A Dissertation Presented
by
Xin Li

to
The Graduate School
in Partial Fulfillment of the
Requirements
for the Degree of
Doctor of Philosophy
in
Computer Science
Stony Brook University

August 2008

Stony Brook University
The Graduate School

Xin Li

We, the dissertation committee for the above candidate for
the degree of Doctor of Philosophy, hereby recommend
acceptance of this dissertation.

Hong Qin, Dissertation Advisor
Professor, Computer Science Department

Xianfeng Gu, Dissertation Co-Advisor
Assistant Professor, Computer Science Department

Joseph Mitchell, Chairperson of Defense
Professor, Computer Science Department

Arie Kaufman
Professor, Computer Science Department

David Breen
Assistant Professor, Computer Science Department, Drexel University

This dissertation is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Dissertation

A Shape Mapping Framework for Graphics and Visual Computing

by

Xin Li

Doctor of Philosophy

in

Computer Science

Stony Brook University

2008

With the rapid advancement of 3D surface scanning technologies and medical modalities, high-fidelity surface models and volumetric data sets of huge size have been routinely acquired through hardware devices. This dissertation concentrates on the challenging research issue of how to build the best possible (surface and volume) mapping between different objects of arbitrarily complicated topological types. Inter-shape mapping, or more specifically, finding meaningful, low distortion correspondence between two given objects is an enabling tool for various applications in digital entertainment, modeling and simulation, shape analysis and retrieval, material synthesis, visualization, etc.

This dissertation research seeks accurate and efficient solutions to this fundamental and important problem. Based on modern geometry, especially differential geometry, algebraic topology and Riemannian Geometry, we have articulated and developed a general and powerful shape mapping paradigm for objects in different dimensions with arbitrary topologies: in the 1D curve case, we devise the conformal invariants as curve signatures; in the 2D surface case, we exploit techniques of topological decomposition and conformal mapping; and in the 3D volumetric case, we focus on harmonic map based on Green function theory. Consequently, compared with all the existing surface and volumetric mapping techniques, our work offers a better and integrated solution to the inter-object mapping problem. The great potential of our shape mapping framework will be highlighted through many valuable applications such as shape analysis, deformation editing, animation morphing, information transfer, re-meshing, texture synthesis, and physics-based modeling. Furthermore, we envision broader application scopes including computer vision, shape database and content-driven information retrieval, digital medicine, virtual environments, etc.

For my beloved family, advisors, friends and colleagues.

Contents

List of Tables

List of Figures

Acknowledgements

First, I would like to thank my advisor Hong Qin for his help during my Ph.D. studies. I cannot imagine myself completing this dissertation without his inspiration, discussion, and encouragement. He led me to the field of geometric modeling and processing and showed such kindness, patience, and deep understanding about geometry and physics that no graduate students expect more from their advisors.

I would like to thank my co-advisor Xianfeng Gu for leading me into the world of mathematics and mesh parameterization. His deep insights and view of math and geometry inspired me to break through my research in parameterization and mapping fields.

I would like to thank Arie Kaufman, Joseph Mitchell, Klaus Mueller, and Steven Skiena for their advices, collaborations, as well as for serving on various committees.

I would like to thank David Breen for taking the time to serve as the external member of my dissertation committee.

I would also like to thank my colleagues in our center of visual computing and computer science department, Xiaohu Guo, Ying He, Jing Hua, Yunfan Bao, Yue Wang, Hongyu Wang, Wei Zeng, Andrew Mehler, Miao Jin, Haodong Hu, Christopher Carner, Yang Wang, Feng Qiu, Zhe Fan, Kexiang Wang, Sen Wang, Xiaotian Yin, Xiang Zeng, Bo Li, and Juan Cao for delightful collaborations and discussions we had together, and thank my friends in Stony Brook, Ce Shi, Marvin Hazan, Jane Hazan, and so many others for their help during the past five years.

My research was supported in part by the following grants awarded to Professor Hong Qin and Professor Xianfeng Gu: CCF-0448399, DMS-0528363, DMS-0626223, IIS-0713145, and IIS-0710819.

Last but not least, I want to thank my family for their endless love and support.

Without their support, this thesis would not have been possible.

This dissertation is dedicated to them.

Publications

Journal Papers

1. **X. Li**, Y. Bao, X. Guo, X. Gu, and H. Qin, "Globally Optimal Surface Mapping between Surfaces with Arbitrary Topology", *IEEE Tran. Visualization and Computer Graphics, Vol. 14, No. 4, pp. 805-819, 2008.*
2. **X. Li**, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin, "Meshless Harmonic Volumetric Mapping using Fundamental Solution Methods", *to appear in IEEE Tran. Automation, Science and Engineering, 2008.*
3. H. Wang, Y. He, **X. Li**, X. Gu, and H. Qin, "Polycube Splines", *to appear in Computer Aided Design, 2008.*
4. **X. Li**, X. Gu, and H. Qin, "Curve Space: Classifying Curves on Surfaces", *in Communications in Information and Systems, Vol. 7, No. 3, pp. 207-226, 2007.*
5. W. Zeng, **X. Li**, S-T Yau, X. Gu, "Conformal Spherical Parametrization for High Genus Surfaces", *in Communications in Information and Systems, Vol. 7, No. 3, pp. 273-286, 2007.*
6. X. Guo, **X. Li**, Y. Bao, X. Gu, and H. Qin, "Meshless Thin-shell Simulation Based on Global Conformal Parameterization", *in IEEE Tran. Visualization and Computer Graphics, Vol. 12, No. 3, pp. 375-385, 2006.*
7. A. Mehler, Y. Bao, **X. Li**, Y. Wang, and S. Skiena, "Spatial Analysis of News Sources", *in IEEE Tran. Visualization and Computer Graphics, Vol. 12, No. 5, pp. 765-772, 2006.*
8. **X. Li**, X. Gu, H. Qin, "Surface matching using consistent pants decomposition", invited to IEEE Transaction of Visualization and Computer Graphics.

Conference Papers

9. **X. Li**, X. Gu, and H. Qin, “Surface Matching using Consistent Pants Decomposition”, *in Proc. ACM Solid Physical Modeling Symposium, pp. 125-136, 2008.*
10. **X. Li**, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin, “Harmonic Volumetric Mapping for Solid Modeling Applications”, *in Proc. ACM Solid Physical Modeling Symposium, pp. 109-120, 2007.*
11. H. Wang, Y. He, **X. Li**, X. Gu, and H. Qin, “Polycube splines”, *in Proc. ACM Solid Physical Modeling Symposium 2007, pp. 241-251.*
12. **X. Li**, Y. He, X. Gu, and H. Qin, “Curves-on-Surfaces: a general shape comparison framework”, *in Proc. IEEE International Conference on Shape Modeling and Applications 2006, pp 352-357.*
13. Y. He, **X. Li**, X. Gu, and H. Qin, “Brain image analysis using spherical splines”, *in Proc. EMMCVPR ’05, Lecture Notes in Computer Science, Vol. 3757, pp.* $633-644$*, Nov. 2005.*

Technical Reports

14. **X. Li**, “Shape Mapping and its Applications”, *Technical Report, Stony Brook University (SUNY), Jan. 2008.*
15. **X. Li**, “Shape Mapping and its Applications”, *2007 IBM Graphics and Visualization Student Symposium, Apr. 2007.*
16. **X. Li**, “Parameterization based Shape Comparison”, *Technical Report, Stony Brook University (SUNY), Jul. 2005.*

Chapter 1

Introduction

During the past decades, proliferation of 3D digital photographic/scanning devices and shape modeling techniques boosts the number of available high quality 3*D* geometric digital data. More and more digital objects have been captured or created, and stored in databases. As a result, the need to the ability of making good use of existing models has gained the prominence. A key concern of computer graphics and shape modeling research is shifting to effectively understanding, retrieving, analyzing and reusing models in databases. Inter-shape mapping is a powerful tool for all these applications because it provides one-to-one correspondence between two given objects. Such correspondence, also called shape registration, is a necessary and important step for these applications.

Inter-shape matching research could arise from the necessity of shape registration. In order to precisely measure and analyze differences between two existing digital models point-by-point, people need to compute registration from one given object to another. Rigorously speaking, this is to find globally continuous one-to-one correspondence between two shapes. However, with the exploration of inter-shape matching techniques, much more other potential has been found besides shape registration and alignment. Shape mapping becomes a very useful tool in broader research fields and serves for many valuable applications in graphics, vision, visualization, animation, and modeling fields. My dissertation research focuses on this fundamental and important problem as well as its various applications.

Figure 1 illustrates the conceptual hierarchy of my dissertation research (we focus on mapping surface and volumetric data because they are most commonly

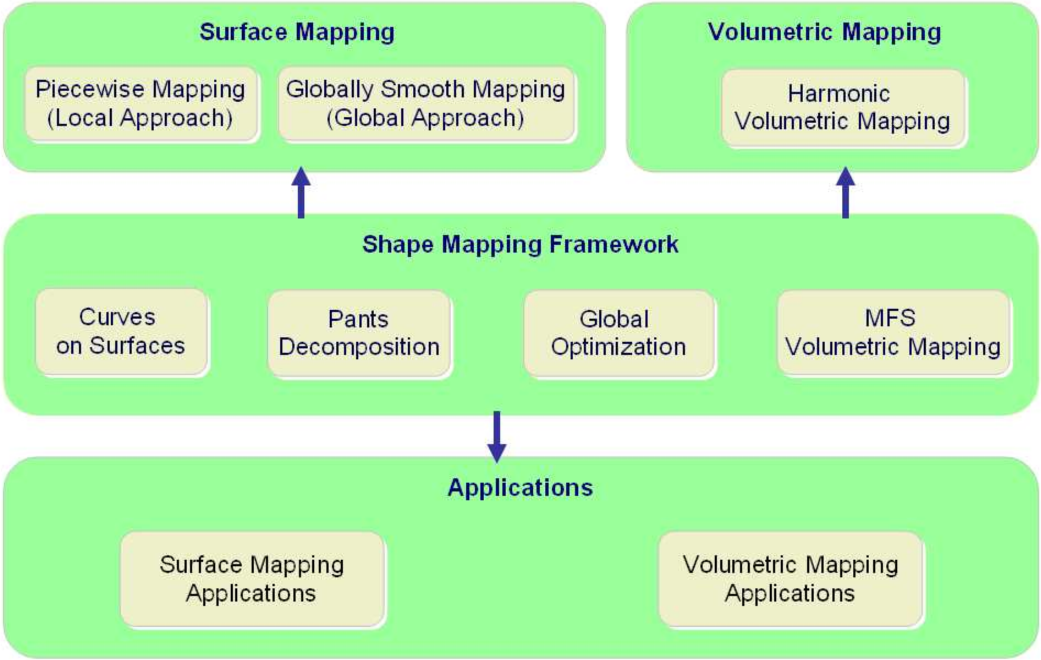

Surface Mapping
Piecewise Mapping (Local Approach)
Globally Smooth Mapping (Global Approach)
Volumetric Mapping
Harmonic Volumetric Mapping
Shape Mapping Framework
Curves on Surfaces
Pants Decomposition
Global Optimization
MFS Volumetric Mapping
Applications
Surface Mapping Applications
Volumetric Mapping Applications

Figure 1: Hierarchy of this Dissertation Research: Key difficulties in current surface and volumetric computation (upper row), our integrated work (middle row), and utilized mapping applications (bottom row).

used in our surrounding physical world). This framework integrated a few projects (the middle row), targeting on key challenging problems in current surface and volumetric mapping computation (the upper row). By solving these key difficulties, we have improved the effectiveness and efficiency of shape mapping computation, and are able to utilize this framework for various applications of surface and volumetric data (the bottom row).

1.1 Problem Statement

1.1.1 Surface Mapping

Rigorously speaking, a surface mapping is a continuous one-to-one function f from one surface M_1 to the second surface M_2. Many existing surface mapping techniques primarily target on surfaces with **trivial topology** (also called genus-zero surfaces), in which case satisfactory inter-surface maps can be composed with the help of many well-established surface parameterization techniques, through some canonical uniform shape domains (such as spheres or disks).

When the input surfaces are general, i.e., **with nontrivial topology**, there is no canonical domain such as the sphere or plane. Therefore, directly computing the global surface mapping becomes much more difficult (please check Chapter 2 and Chapter 5 for details from technical aspects). We can classify current approaches for general surfaces mapping computation into two categories: one is **the local approach**, also called the piecewise mapping method (segmentation + local optimization); and the other is **the global approach** (global optimization without segmentation).

Piecewise Mapping

Most state-of-the-art techniques for general surface mapping follow the direction of local approach. They firstly partition two surfaces into two consistent sets of sub-regions with simple topology, and then compute locally optimized mapping between corresponding sub-regions. These local results are finally composed to a global continuous map.

The key technical challenge in piecewise mapping computation is **how to effectively generate consistent partitioning** on two or more input surfaces. Specifically, the partitioning should have the same number of sub-regions, and these sub-regions should have the same adjacency relationship. To generate such consistent partitioning, current surface mapping methods usually use the following two ways:

- The straightforward approach for consistent partitioning generation is by *manual design* [18, 35, 132]. The user needs to sketch the partitioning graph (also called the base mesh) on both surfaces consistently and indicates the correspondence between pairs of subregions. The limitations of this approach include first, it is quite labor intensive, second, the user has to possess deep domain knowledge for the design, and finally, when the input surfaces are with high genus or complicated geometry, consistent base mesh could be too complicated to design manually.

- Recent work has been trying to seek more automatic methods in consistent partitioning generation. This approach is based on *user-provided feature points* [73, 78, 99, 106]. The user firstly labels feature constraint points on two surfaces with corresponding indices. Then some tracing algorithm is conducted by the system to connect these markers in consistent ways, and generate consistent partitioning, with all markers becoming endpoints of connecting paths. These methods can greatly relieve the effort of manually designing the base mesh, but they still have a few limitations. First, when input surfaces are with high genus or are not geometrically similar, heuristic tracing algorithms are often error prone; second, large amount of markers may be necessary in these schemes for tracing unambiguous partitioning graphs on high-genus surfaces (for example, in [73], at least four markers are necessary for each topological handle), which still cost a lot of labor.

The limitation of above existing methods pushes us to consider the following issues for surface mapping:

- **How to analyze segmentation on different surfaces?**
- **How to design a more automatic and robust scheme to generate consistent partitioning on different surfaces, with arbitrary topological types?**

Globally Smooth Mapping

Piecewise methods simplify the computation of the one-to-one continuous mapping between two given surfaces. However, this “partitioning + local mapping” paradigm has difficulties in controlling the global mapping smoothness and distortions. First, the mapping results heavily depend on the initial partitioning; second, the mapping along the sub-regions boundaries is only C^0 continuous; third, many current techniques judge distortion of their mapping functions only by visual effects, without rigorous mathematical foundation and analysis.

On the other hand, mapping quality directly dictates its end effect of its enabling applications. Many applications such as shape comparison, morphing, scientific computation, spline construction, etc. require a globally low-distorted surface mapping. It is worthwhile to study the surface mapping from a global point of view. Two critical issues are:

- **How can we rigorously measure the entire distortion of mapping between two surfaces?**
- **What type of distortion criteria should we use?**

In this dissertation, we choose *harmonicity* as the criterion to measure quality of both surface and volumetric mapping, because harmonicity is a physically natural way to measure *stretching (angular distortion)* of a mapping between two shapes. The formulated energy that characterizes the harmonicity of a map is called the *harmonic energy*. It has already been widely used as a criterion to measure the quality of surface parameterizations. In many well-established surface parameterization techniques [24,96], harmonic map has already been efficiently computed and used. Unfortunately, unlike the surface parameterization, in the surface mapping case, the complicated geometric shape of the target surface will almost inevitably cause the direct computation of the harmonic map between two surfaces (especially with complicated topology) to fail and get stuck at local minima.

- **Whether there is a way that the numerical estimation of the mapping distortion can guide the improvement of surface mapping quality, so that we can reach the globally stretching-minimized mapping?**

1.1.2 Volumetric Mapping

Surface mapping does not consider whether objects are with solid interior regions. And naturally we shall consider the generalization of surface mapping to $3D$ volumetric case because these interior regions of objects oftentimes carry important information. The physically natural criterion of surface mapping, harmonicity is also defined for $3D$ volumetric mapping. A harmonic volumetric map is a smooth one-to-one mapping between a solid object M_1 and the other solid M_2, given their surface mapping $\vec{f'} : \partial M_1 \to \partial M_2$ (∂M_1 and ∂M_1 represent boundaries of M_1 and M_2 correspondingly).

This can be rigorously formalized as a $3D$ vector function $\vec{f} \in \mathbb{R}^3$, such that

$$\begin{cases} \Delta\vec{f}(\mathbf{p}) = 0 & \mathbf{p} \in M_1, \\ \vec{f}(\mathbf{p}) = \vec{f'}(\mathbf{p}) & \mathbf{p} \in \partial M_1. \end{cases} \tag{1}$$

where the Δ is defined continuously in $3D$ as

$$\frac{\partial^2}{\partial x^2}+\frac{\partial^2}{\partial y^2}+\frac{\partial^2}{\partial z^2}, \tag{2}$$

As shown in the above equation 1, first, $\vec{f}$ bijectively maps points in M_1 to points in M_2; second, it has to be harmonic, meaning, the $3D$ Laplacian operator, as defined in the equation 2, acting on f is zero everywhere; third, it must satisfy the boundary condition, mapping source surface points onto the target surface.

Due to the huge complexity of volumetric grids, direct computation is computationally challenging. The existing harmonic volumetric mapping technique such as [42] discretizes the volume space by a tetrahedral mesh, and then iteratively reduces the discretized tetrahedral harmonic energy. This kind of approach has some limitations. First, the generation of a tetrahedralization of the given solid object is nontrivial and sometimes error-prone; second, when the resolution of the tetrahedral mesh is high, the computation is still very costly. Therefore, it is necessary for us to further explore on this problem and study:

- **How to reduce the computation complexity of volumetric mapping?**
- **How to compute the volumetric mapping more robustly?**
- **How to analyze or restrict the mapping distortion?**

1.2 Contribution

This dissertation consists of the following contributions.

1. In one-dimensional data case, we study spacial simple closed curves embedded in genus-0 surfaces, and find a conformal invariant that characterizes how a simple closed curve partitions its embedded surface. We use such conformal invariant functions as signatures for curves-on-surface; these signatures provide a classification for all curves-on-surfaces, meaning signatures and curves correspond uniquely. Furthermore, this curve signature is used to study segmentation of different given surfaces, more specifically, it can be utilized for segmentation matching and segmentation transfer. Based on this, we design a semi-automatic piecewise surface mapping framework. This work is based on complex analysis and conformal mapping theory. (Chapter 3)

2. In two-dimensional data case, we study the surface mapping computation problem. Our work on curve signature provides a semi-automatic piecewise surface mapping framework; however, it is not fully satisfactory in terms of surface mapping. The reason is first, the consistent segmentation is not solved automatically: even we can transfer segmentation consistently among different surfaces, we need the segmentation to be generated on at least one surface; second, this framework only works for input surfaces with the same topology (specifically, currently only genus-0 surfaces, although we believe this framework can be generalized to high-genus-surface cases), which is not enough for general surface mapping purpose. (Chapter 3)
3. Therefore, for two-dimensional manifolds (surfaces), we design a decomposition scheme called "canonical pants decomposition" that applies on different surfaces automatically. With this decomposition, we are able to get the consistent segmentation on surfaces with complicated topology with least user involvement. This framework also handles mapping computation between surfaces with different topology, which is called *topological evolution*. Under our framework, the user can easily manipulate topological evolution with only a fewest intuitive operations. This piecewise surface mapping framework is automatic and robust, and also easily integrates feature alignment from users for possible semantics purpose. (Chapter 4)
4. We also study the globally optimal mapping between two surfaces with complicated topology. To our best knowledge, this is the first presented algorithm for computing surface mapping with the least global stretching energy, with both experimental and rigorous theoretic proof for its optimism, uniqueness and convergence. (Chapter 5)
5. Towards three-dimensional data, we further extend our surface mapping framework to solid models. Using our surface mapping as a boundary condition, we compute the harmonic volumetric mapping between two given solid objects. We bring into graphics and modeling field the so called method of fundamental solution, which greatly improves the efficiency and robustness of the volumetric mapping computation. To our best knowledge, this is the first investigation for mapping solid objects with nontrivial topology. (Chapter 6)

6. Besides the theoretical contribution on the above shape mapping in one, two and three dimensional data, we also demonstrate the usefulness and power of shape mapping by applying surface and volumetric mappings in various applications in modeling, graphics, and visualization fields. We will address this in the following sections. (Also see Chapter 7 and Chapter 8)

1.3 Motivating Applications

Both surface mapping and volumetric mapping have various valuable applications in a wide research fields including geometric modeling, geometric processing, computer graphics, visualization, vision, and medical imaging.

Surface Mapping Applications

1.3.1 Morphing

Shape mapping can be used to generate the morphing sequence between two meshes for animation. With a one-to-one mapping between two given surfaces, each point on the source surface has a unique corresponding point on the target surface, therefore, we can simply conduct linear interpolation between corresponding points and get a morphing trajectory from first object to the second object. On the other hand, different interpolation techniques can be applied for different morphing effects.

1.3.2 Surface Registration/Comparison, Shape Retrieval

With low distortion one-to-one mapping between two surfaces, we naturally get registration between the surfaces; then based on either curvature or some other property, we can compute their difference point-by-point for comparison purpose, and design a metric to measure distance (difference) between surfaces. This furthermore leads to the application of shape retrieval.

1.3.3 Cross-Surface Parameterization

Geometric processing and computation usually require heavy computation on surfaces with complicated geometry or topology. For example, to perform physically based simulation on convoluted surface data is very time consuming. With surface mapping, we can convert these computations into some canonical domains, where the computation efficiency and accuracy may be greatly improved. Examples include physically-based simulation based on surface mapping, and spline construction through canonical shape mapping such as polycube mapping, etc.

1.3.4 Texture/Information Transfer

One-to-one correspondence provides a tool to correlate information between two objects. Properties such as color, texture, density, or even strain/stress tensor can all be transplanted from one object to its corresponding region in the second object. An application in graphics is texture transfer, which transplants texture from the source object to the target object.

1.3.5 Spline Construction

Spline construction, although we list it separately, is an application of cross-surface parameterization (Section 1.3.3). Tensor product splines are usually built upon planar domains. With mapping from original surfaces to polycubes, we get the canonical rectangular domain (faces for the polycube) for the spline construction.

Volumetric Mapping Applications

1.3.6 Solid Object Comparison

Similar to surface registration, volumetric mapping can be used to align solid objects. Based on the physically-natural harmonic volumetric mapping, we are able to precisely compute the deformation energy required to deform the first object into the second one, as well as illustrate at each point where the stretching energy concentrates.

1.3.7 Tetrahedral Remeshing

As triangle meshes are used to represent surfaces, tetrahedral meshes can be used to represent solid objects. Given an input solid object, how to generate a regular tetrahedralization (or hexahedralization) is an interesting topic for mechanical engineering research, because the regular structure provides better efficiency and accuracy in physically based computation and geometric operation. Regular tetrahedralization (or hexahedralization) can be easily generated on regular solid domains such as solid polycubes; then with the volumetric mapping, such a regular structure can be transferred onto the given object with complicated geometry. We call this application "tetrahedral remeshing".

1.3.8 Solid Texture Synthesis

For visualization and virtual reality, people want to synthesize natural interior solid texture of a given object based on its boundary surface texture. Based on our volumetric mapping algorithm, we can generate smooth and natural-looking solid texture in real time.

1.3.9 Data Fusion

Volumetric mapping, as a shape registration tool, can be used for fusing medical data from different modalities. With the correspondence provided by domain experts, how to merge medical data obtained from different scanning devices is an important research direction for medical imaging. Our shape mapping naturally provides a powerful tool for this purpose.

1.3.10 Deformation Transfer

To reuse dynamic modeling data is an interesting topic in the geometric modeling field. Much research has been conducted based on surface data. Transferring the deformation among solid data also has important applications for reusing sequential volumetric data and is natural when the interior region of the solid data is anisotropic. When we have a deformation sequence, and a target object, we can map the starting source object to the target object, and then transfer all the deformation gradient tensors sequentially. Then we may be able to integrate the deforming sequence on the target object by integrating these gradient tensors while imposing some constraints.

1.4 Dissertation Organization

This chapter (Chapter 1) presents the motivation, central problems and the contribution of this dissertation. The remainder of this document is organized as follows. Chapter 2 reviews previous related work in shape mapping computation. In the following three chapters, we explain our three projects in surface mapping computation (in Chapter 3, Chapter 4, and Chapter 5 respectively). Chapter 6 focuses on our work for volumetric mapping computation. Chapter 7 illustrates the results and power of surface mapping through various applications. Chapter 8 shows the applications for volumetric mapping. Chapter 9 concludes our work and discusses both our contribution in shape mapping and limitations of our current methods. Future research directions are also elaborated in Chapter 9.

Chapter 2

Background and Related Work Review

This chapter reviews the background and work related to shape mapping research. It has been composed into three main parts dealing with data from 1*D* manifolds (curves), 2*D* manifolds (surfaces), to 3*D* manifolds (solid).

Surface mapping computation has deep roots in surface segmentation (Section 2.1) and surface parameterization (Section 2.2). We will briefly recap most related literatures in these two fields before we go into the review of existing piecewise surface mapping methods (Section 2.3, Section 2.4) and globally smooth mapping techniques (Section 2.5). In 3*D* cases, our volumetric mapping computation algorithm has better effectiveness and efficiency over existing volumetric mapping methods (Section 2.6), because it is based on a boundary method that we brought into the graphics and modeling community, called the Method of Fundamental Solution (MFS) (Section 2.7).

2.1 Curve Comparison and Surface Segmentation

Analysis of the one-dimensional manifold, curves on the plane has been a widely studied problem in the computer vision fields. Effective curve matching and comparison methods have focused on properties of the planar curve itself [6, 7, 16, 30, 53, 88, 95, 107, 117, 129]. Because of the different application purposes, very little work has been conducted to specifically study curves defined on surfaces so that people can use them to study the segmentation of embedded surfaces, and to aid in surface matching. Recent research on conformal geometry offers a new way to study curve in this direction. [108] modeled planar simple closed smooth curves by diffeomorphisms from a circle to itself via conformal mapping and proved the space of all such curves modulo scaling and transformation is isomorphic to the diffeomorphism group of the circle quotient Möbius transformations group restricted on the circle. Our work in Chapter 3 follows the same philosophy and generalizes this idea from planar curves to curves embedded on $2D$ manifolds, so that we can study the segmentation of given surfaces by studying curves.

2.2 Surface Parameterization

In order to build up a one-to-one mapping from one surface to another, we can use a straightforward yet effective method that uses a regular domain as the bridge. A common domain for surfaces is the plane. Mappings from a surface to a suitable planar domain is usually called *surface parameterization*. Surface parameterization itself has been extensively studied, and have been playing an important role in the modern graphics, modeling and geometric processing pipeline.

Surface parameterization can be treated as a special case of inter-surface mapping (in the sense that one of the surface is the plane). Despite surface mapping's connection with surface parameterization over canonical domains, techniques for computing an mapping between general surfaces are different and seems more challenging (please see Section 5.1 for more technical details). Therefore, a thorough review of surface parameterization is beyond the scope of this work. In the following, we only briefly summarize the most related work, and refer the reader to [29] and [111], which provide extensive surveys of state-of-the-art techniques in the field of surface parameterization.

The most developed and important parameterization task is to map a surface $M \subset \mathbb{R}^3$ of disk topology into the plane. The early driving force for the development of planar parametrization techniques mainly came from the need for texture mapping in Computer Graphics.

Floater [27] addressed that a *valid* parameterization of a disk-topology mesh

$\mathcal{M}$ is any *planar triangulation* which is isomorphic to $\mathcal{M}$. More specifically, each vertex of the mesh is mapped to a point in $\mathbb{R}^2$; each edge is mapped to a straight line segment in $\mathbb{R}^2$ and the edges only intersect at their common endpoints. In other words, the planar parameterization is valid when there are no intersections among the edges in the planar domain. This means that, locally, no triangle flip or degeneracy exists. Global degeneracy, however, might still occur.

One of the earliest methods for parameterizing an open genus-0 surface into the plane was first introduced to the graphics community by Eck et al. [24]. Pinkall and Polthier derived the discrete Dirichlet energy earlier in [96] for the computation of discrete minimal surface. The main advantage of the above method for computing discrete harmonic mapping is that it is a quadratic minimization problem and therefore reduces to solving a sparse linear equation system.

However, discrete harmonic maps in general do not always guarantee injectivity. Floater [27] describes a generic method, which is called the *convex combination map*, to map a 3D mesh into the plane without foldovers. Floater's method is a generalization of Tutte's method [123] for planar graph embedding in the 60's. Floater observed that the *'barycentric mapping'* method used by Tutte can be made much more general by allowing each interior vertex to be any convex combination of its neighbors and he also provided an algorithm for choosing the convex combinations so that the mapping locally preserves the shape of the given surface.

By discretizing the discrete Dirichlet energy defined in [96], Desbrun et al. [21] constructed free-boundary harmonic maps.

When input surfaces are closed genus-zero surfaces, a sphere is a natural parameterization domain. An important point for spherical parameterization is that according to Gu and Yau [43], harmonic maps from a closed genus zero surface to the unit sphere are conformal, i.e. harmonic and conformal maps are the same when the surfaces are topological spheres. Haker et al. [48] first mapped the given surface into the plane and then uses stereographic projection to subsequently map to the sphere. Gu and Yau [44] proposed an iterative method to approximate a harmonic spherical map. Gotsman et al. computed a spherical mapping that guarantees the one-to-one property in [34]. The angle flattening method of Shefer and de Sturler [109] for disk-like surfaces had been generalized to spherical case by Sheffer, Gotsman, and Dyn [110]. The stretch metric approach of Sander et al. [102]

had been generalized to spherical case by Praun and Hoppe [98].

As a criterion of surface parameterization, angle preservation is typically addressed in these above work either from the harmonic point of view (Dirichlet energy) [21, 24, 96] or from the conformal point of view (Cauchy-Riemann equation) [21, 79]. Conformal geometry has been studied for conformal surface parameterization [44, 101].

More recently, the hyperbolic structure of Riemannian surfaces has been introduced to surface parameterization. Thurston firstly introduced circle packing in [122]. An effective algorithm and implementation is presented by Stephenson in [113]. Circle packing has also been generalized to circle patterns [10] and used for surface parameterization in [70]. Hamilton first introduced Ricci flow on surfaces in [49]. Theoretical results of combinatorial Ricci flow are later generalized in [14], and applied in surface parameterization and mapping fields in [57] and [81].

2.3 Mapping between Genus-0 Surfaces

To compute surface mapping between genus-zero surfaces, people widely use the sphere (for closed surfaces) and the plane (for open surfaces) as intermediate domains. Existing planar parameterization of spherical parameterization techniques can be used to flatten surfaces onto above canonical domains.

Kent et al. [69] mapped star-shaped surfaces onto spheres, and merged them by clipping one sphere to the other. Kanai et al. [62] used harmonic maps to build correspondence from surfaces to the unit disk domain, therefore not only the star-shaped surfaces, but also all genus-zero closed or open surfaces can be mapped easily. However, it only allowed one constraint point from users. Alexa [1] proposed to match multiple feature points between genus-0 surfaces. His work wrapped two surfaces onto a unit sphere by minimizing a distance function, and feature points on the surface were aligned and the resultant embedding was used for the surface mapping. They started to aim for matching multiple feature points. However, its limitation is that no bijectivity is guaranteed and hard constraints may not be fully enforced. More recently, Asirvatham et al. [4] used their constrained spherical parameterization to map genus-zero surfaces onto the sphere, the progressive mesh was used to get a simple base mesh and to enforce constraints at certain positions on the sphere. This method allows multiple hard constraint points between genus-0 surfaces.

2.4 Piecewise Mapping between General Surfaces

For surfaces with more general topology, common canonical domains such as disks and spheres become unavailable. Directly solving intra-surface mapping usually fails. Most techniques follow the direction of piecewise surface mapping. First, they segment the surfaces with complicated topology into a consistent set of sub-regions, then, they compose or refine the global result from the mapping between the sub-regions.

DeCarlo and Gallier [18] designed a surface mapping framework based on user-specified base meshes. When base meshes are carefully designed, the framework is flexible, and mapping between surfaces with different topology can be computed. However, deep domain knowledge in topological surgery may be required to manually design consistent base meshes; and when the surface has high genus, the design are usually quite complicated. Only examples up to genus-2 were provided in their work. Gregory et al. [35] and Zöckler et al. [132] also used the base mesh approach. When the consistent "base mesh" have been manually designed, harmonic or barycentric mappings are used to correspond these sub-regions accordingly. More earlier surface mapping work for morphing applications can be found in the survey [77].

Recent work has been trying to seek more automatic methods to consistently generating the base mesh. Lee et al. [78] used their MAPS algorithm to hierarchically map fine meshes onto a common base mesh. Praun et al. [99] introduced a graph tracing algorithm to transfer the coarse base mesh from one surface to another with the same topology. Kraevoy and Sheffer [73] designed another algorithm to trace out base meshes consistently on different surfaces. To build up the base meshes, many feature points have to be provided by users for high genus surfaces. For example, at least four points are required for each topological handle to proceed the base mesh tracing algorithm. Schreiner et al. [106] first traced original surfaces into a corresponding set of triangular patches, with feature points as path endpoints, and created original surfaces' progressive mesh representations. Then they created a trivial map onto the base mesh, and iteratively refined the map back to the original surfaces.

Base mesh constructions (consistent segmentations) could consume a large amount of human labor. To improve this, a direction of recent research is the automatic generation of surface maps. This automation becomes very challenging when surfaces are with complicated topology, where far less work has been explored. Furthermore, when given surfaces with different topologies are present, it is even more difficult. Manual base mesh design [18] requires large effort and strong expertise from the user. This motivates us to seek an automatic method for consistent shape segmentation for surfaces with complicated topology.

2.5 Globally Smooth Mapping for General Surfaces

The common drawbacks of the above piecewise surface matching using aforementioned techniques are that constructing the patch layout oftentimes involves a number of fragile heuristic algorithms. Furthermore, the mappings are generally only C^0 continuous across the patch boundaries. In applications such as building domains for splines, a global continuity is critical. The work of [71] addresses the continuity problem by taking into account linear transition functions across patch boundaries. The manifold concept in mapping is introduced in [36], which primarily focuses on topology instead of geometry, thus it is difficult for designing optimization algorithms.

2.6 Volumetric Mapping

Similar to the harmonicity extensively studied in the literature of surface mapping, harmonicity in the volumetric sense is also addressed from the point of view of minimizing Dirichlet Energy. Therefore, people try to compute a function in $\mathbb{R}^3$ with the vanishing Laplacian, representing the smoothness of the mapping function. Due to its computational complexity, much fewer work has been conducted in the graphics and modeling fields. Gu et al. [42] studied the formula of harmonic energy defined on tetrahedral mesh and computed the discrete volumetric harmonic maps by a variational procedure. Ju et al. [61] generalized the mean value coordinates [28] from surfaces to volumes and built a smooth volumetric interpolation. More recently, Joshi et al. [59] presented harmonic coordinates for volumetric interpolation and deformation purposes, their method guaranteed to provide non-negative weights and therefore in concave regions, and led to a more pleasing interpolation result, compared to [61]. However, the technique introduced in [59] lacked the closed form expression for a given interior point, which increases the computational burden and reduces numerical accuracy. A latest work from Martin et. al. [83] presented a volumetric parameterization method for genus-zero tetrahedral mesh and generated trivariate B-spline based on it.

2.7 Boundary Method and Method of Fundamental Solution

The partial differential equations we try to solve are nonlinear and therefore can only be solved numerically. The numerical methods for solving PDEs can be classified into **domain discretization methods** and **boundary discretization methods**(also called boundary element methods). The most popular domain discretization methods include the Finite Difference Method(FDM) and the Finite Element Method(FEM). These methods reduce the continuous space and infinite degrees of freedoms to a finite set by discretizing the interior domain as well as the boundaries, which thus leads to large computation complexity. In the Boundary Element Method(BEM), only the boundary of the domain needs to be discretized and thus leads to a reduction of the problem dimension by one, saving substantial computational time and storage. BEM is of less general applicability than the domain discretization methods. But for some problems that it can be applied to, it has been proved to be more effective.

BEM have two different formulated ways. One is the direct formulation, including many existing work, where the functions appearing in the integral equations are the actual physical variables of the problem. The other approach is the indirect formulation, the solution is approximated by a function that satisfies the governing equation but has unknown source densities. The Method of Fundamental Solution (MFS) falls into this category.

MFS was first proposed by Kupradze and Aleksidze [75], and employed by Mathon and Johnston [84] for solving Dirichlet problems. Later it has been widely used in the mathematical physics and mechanical engineering fields. It was first used in potential problems, including Mathon and Johnston's work in the 2D case [84], and easily extended to 3D problems, e.g. [58] solved the computation of dipole fields. The MFS with fixed singularities has been also applied to several problems in elastostatics [97] and acoustics [74]. MFS methods were also developed for biharmonic problems [63]. There are more general operators that MFS can be applied to, as long as the fundamental solution of the differential equation governing the problem in question is available.

The harmonic volumetric mapping can be computed with the method of fundamental solution (MFS). Notable work among boundary methods for solving elliptic partial differential equations (PDEs) includes the classical boundary integral equation and boundary element method (BIE/BEM), which has been widely used in many engineering applications [5], and was introduced into computer graphics for the simulation of deformable objects in [56]. One of the major advantages of the BIE/BEM over the traditional finite element method (FEM) and finite difference method (FDM) is that only boundary discretization is usually required rather than the entire domain discretization needed for solving the PDEs numerically. The boundary method already reduces the original volume problem into the boundary surface scale, yet its algorithm efficiency might be able to further improved using the Fast Multipole Expansion techniques [13]. A comprehensive review of the MFS and kernel functions for solving many elliptic PDE problems was documented in [26]. Compared with the BIE/BEM approach, the MFS uses only the fundamental solution in the construction of the solution of a problem, without using any integrals over boundary elements. Furthermore, the MFS is a true meshless method, since only boundary nodes are necessary for all the computation.

"Meshless" has the advantage of simplicity since neither domain nor mesh connectivity is required for storage and computation; so it becomes very attractive in scientific computing and modeling especially in fracture simulation when the underlying topology is frequently updated. Several meshless methods have been reported in the literature, for example, the smooth particle hydrodynamics (SPH) method [87], the element-free Galerkin (EFG) method [8], etc. Meshless method was introduced into the physically-based graphical modeling by Desbrun and Cani in [19]. Later they applied Smoothed Particle Hydrodynamics (SPH) to simulate highly deformable bodies [20]. Müller et al. [90] presented a method for modeling and animating elastic, plastic, and melting volumetric objects based on the Moving Least Squares (MLS) approximation of the displacement field. Most recently, they presented a geometrically motivated approach in [89] for simulating deformable point-based objects. Pauly et al. [94] simulated volumetric meshless fracture with a highly dynamic surface and volume sampling method that affords complex fracture patterns of interacting and branching cracks. Guo et al. [47] applied meshless method to physically-based thin-shell simulation based on point-based global conformal parameterization of the surfaces.

2.8 Applications of Shape Matching

Shape mapping can be directly applied as a registration tool for shape matching and comparison. Unlike other comparison techniques (please see [55, 80, 118] for thorough surveys), for example, descriptors based on histogram ([3, 64–66, 92, 93]), descriptors based on spatial decomposition ([67, 68, 91, 103, 125, 126]), and descriptors based on spatial graphs or skeletons ([52, 115]),the comparison based on pointwise registration [131] provides us not just a distance value but also the intuitive error distribution for afterward effective analysis.

Surface mapping and surface parameterization are also widely used in many applications of graphics, modeling and vision fields, including texture mapping [34, 48, 98], shape classification [45], medical data matching [40, 41], surface and volumetric remeshing [23, 83], digital data reconstruction [2] and compression [37, 54], spline construction [38, 39, 127, 128] and so on.

Chapter 3

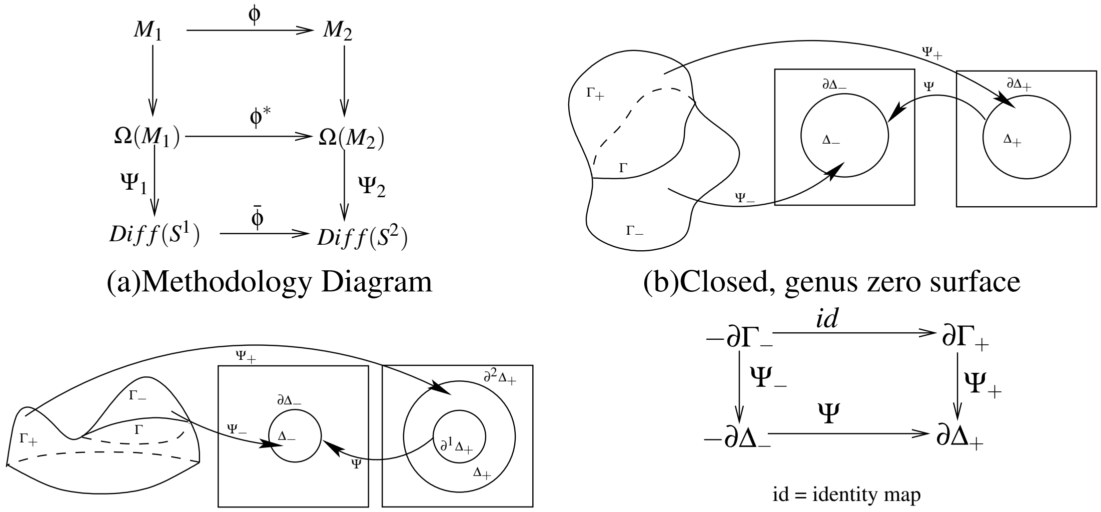
Curves-on-Surfaces Classification and Surface Matching

Surface segmentation can be studied through the curves that partition the given surface. In this chapter, we introduce a conformal invariant function to describe and analyze simple closed curves on surfaces. Such a function, also called the curve signature, provides a classification for all simple closed curves on surfaces, and can be utilized for segmentation matching and segmentation transfer.

3.1 Introduction

Our surface matching computation framework starts from an easier perspective ($1D$) by considering all the closed curves on a surface. Since surface segmentation plays an important role in piecewise surface mapping computation, and under any segmentation, boundaries of sub-regions are simple closed curves.

We define the set of all closed curves on surface M as *curve space* and denote it as $\Omega(M)$. The *curve space* on surfaces contains the relevant geometric information of the surface and is easy to process. This philosophy of analyzing shapes via their associated curve space has a technically sound foundation in algebraic topology, infinite dimensional Morse theory and Teichmüller space theory in complex geometry. Milnor [86] pointed out that $\Omega(M)$ is an infinite dimensional manifold, the length of curves on a surface is a *Morse function*, and its critical points are *geodesics*. Morse theory is used to analyze the topology of $\Omega(M)$, which determines the topology of M. From the point of view of differential geometry, we know that the local surface shape is completely determined by all the curves defined in its neighborhood. Motivated by the above technical advances, our research in this chapter naturally follows the same methodology, and is specifically based on Teichmüller space theory.



$$\begin{tikzcd}
M_1 \arrow[r, "\phi"] \arrow[d] & M_2 \arrow[d] \\
\Omega(M_1) \arrow[r, "\phi^*"] \arrow[d, "\Psi_1"'] & \Omega(M_2) \arrow[d, "\Psi_2"] \\
Diff(S^1) \arrow[r, "\bar{\phi}"] & Diff(S^2)
\end{tikzcd}$$

(a)Methodology Diagram

(b)Closed, genus zero surface

(c)Open, genus zero surface

$$\begin{tikzcd}
-\partial\Gamma_- \arrow[r, "id"] \arrow[d, "\Psi_-"] & \partial\Gamma_+ \arrow[d, "\Psi_+"] \\
-\partial\Delta_- \arrow[r, "\Psi"] & \partial\Delta_+
\end{tikzcd}$$

id = identity map

(d)Curve signature

Figure 2: Theory foundation of curve signatures

Consider two surfaces M_1 and M_2 (Fig 2(a)), $\phi : M_1 \to M_2$ is a diffeomorphism between them, any curve (By curve, we mean closed curve in the remainder of this chapter.) $\Gamma_1 \in \Omega(M_1)$ will be mapped to a curve in $\Omega(M_2)$ under this mapping: $\Gamma_2 = \phi \circ \Gamma_1$. Therefore, ϕ induces a one-to-one mapping ϕ^* from $\Omega(M_1)$ to $\Omega(M_2)$ by $\phi^* : \Omega(M_1) \to \Omega(M_2)$. The intrinsic relations between surfaces can be analyzed by studying ϕ^* instead of ϕ.

Furthermore, we map the curve space $\Omega(M)$ to a canonical Lie group $Diff(S^1)$, where $Diff(S^1)$ denotes the group of all diffeomorphisms from the unit circle S^1 to itself. We denote this map as $g_i : \Omega(M_i) \to Diff(S^1)$. Consequently, $\phi^* : \Omega(M_1) \to \Omega(M_2)$ induces a mapping from $Diff(S^1)$ to itself by $\bar{\phi} := g_2 \circ \phi^* \circ g_1^{-1}$.

The diagram in Fig 2(a) conveys our methodology in a graphical means: three mappings $\phi, \phi^*, \bar{\phi}$ are closely related; any one of them determines the remaining two. In another word, for the purpose of studying surfaces M_1, M_2 and the maps ϕ between them, we can study their curve space $\Omega(M_1), \Omega(M_2)$, signatures of the curves $Diff(S^1)$, and the mappings $\phi^*, \bar{\phi}$. We propose the following theoretic results to further re-enforce the above geometric intuition.

Theorem 1 *If M is a metric oriented genus zero surface, then the curve space $\Omega(M)$ and $Diff(S^1)$ are equipped with L^2 metric, the map Ψ from its curve space $\Omega(M)$ to $Diff(S^1)$ is a homeomorphism.*

Therefore, in order to measure the distance between two curves on a surface, it is sufficient to measure the distance between two signatures defined in $Diff(S^1)$.

Theorem 2 *M_1 and M_2 are two oriented metric surfaces, ϕ is a conformal map if and only if $\bar{\phi}$ is the identity map of $Diff(S^1)$.*

We also believe that, the mapping from ϕ to $\bar{\phi}$, $F : \phi \rightarrow \bar{\phi}$ reveals a lot of geometric information about M_1 and M_2. By choosing appropriate metrics, F is continuous. The kernel of F is all the conformal mappings between M_1 and M_2.

3.2 Signatures in Curve Space

This section outlines our theoretical results on how to compute curve signatures for curves defined on a surface.

3.2.1 Theory and Algorithm Overview

Given a simple closed curve Γ on genus zero surface M, the central idea to compute its signature is illustrated in Fig 2(b), (c), (d).

Case 1: If M is closed, as shown in Fig 2(b), then Γ partitions M into two components Γ_+, Γ_-, both are topological disks and can be conformally mapped onto planar unit disks Δ_+, Δ_- by Ψ_+, Ψ_-. Γ is the boundary of Γ_+ and Γ_-, denoted by $\partial\Gamma_+ = \Gamma$ and $\partial\Gamma_- = -\Gamma$, and is mapped to the disk boundary, which are unit circles $\Delta = \partial\Delta_+ = -\partial\Delta_-$. The mapping induced by Ψ_+ and Ψ_- on the boundaries $\partial\Delta_+$ and $\partial\Delta_-$ is a diffeomorphism (differentiable and has a differentiable inverse). This diffeomorphism $\Psi : \partial\Delta_+ \to \partial\Delta_-$ is the *signature* of Γ.

Case 2: If M is open, as shown in Fig 2(c), then Γ partitions M into a topological disk Γ_- and a topological annulus Γ_+. Γ_- can be conformally mapped onto a unit disk Δ_-, while Γ_+ can be conformally mapped onto an annulus Δ_+ with unit inner radius. We denote such an annulus with unit inner radius as the *canonical annulus*, the inner boundary of Δ_+ as $\partial^1\Delta_+$, and use the diffeomorphism $\Psi : \partial\Delta_- \to \partial^1\Delta_+$ as the signature of Γ.

In [108], Sharon and Mumford used Teichmüller theory to prove that any simple closed planar curve can be represented with such a diffeomorphism from a unit circle to itself uniquely up to scaling and translation. In this work, we generalize this idea to arbitrary genus zero surfaces using Riemann surface theory.

In technical essence, we compute the conformal mapping for each component segmented by the curve, and take the boundary mappings Ψ as shown in Fig 2(d) as the signature. Some landmarks and constraints are used to eliminate the so-called Möbius ambiguity.

3.2.2 Conformal Map from an Open Genus-zero Surface to a Disk

We seek a conformal map Φ from a disk-like surface M to a unit disk. The map does exist according to Riemann mapping theory. Extensive relevant work has been done on finding a good *parameterization* for disk-like surfaces. However, complete conformality is usually not guaranteed. Based on the fact that the harmonic map from a closed genus zero surface to a sphere is also conformal, we use the *double covering* technique [45] to convert an open surface to a closed one, and reduce computing Φ to computing a harmonic map from a *double covering* of M onto a sphere.

For an open surface M, we compute the double covering of M and then compute its harmonic mapping onto a sphere. Due to the exact symmetric property of double covering, the boundary ∂M is harmonically mapped onto the equator of the sphere and M is conformally mapped onto a hemisphere. Then we compose a stereo graphic projection to get a conformal map from M to the unit disk.

3.2.3 Conformal Map from a Closed Genus-zero Surface to a Sphere

To compute a conformal map Φ from a closed genus-zero surface M to a sphere, we initiate a map between them and minimize the harmonic energy by diffusing the heat-flow on the sphere surface. This process is introduced and proved to converge to a harmonic/conformal map [41].

The process is given in the following algorithm of spherical harmonic mapping:

3.2.4 Conformal Map from a Topological Annulus to a Canonical Annulus

For curves on an open genus-zero surface, we need to compute a conformal map Φ from a topological annulus M (with $\partial M = \Gamma_1 - \Gamma_2$, where Γ_1 and Γ_2 are two boundaries) to a canonical planar annulus. First, we double-cover the surface to get a closed genus-one surface; next we compute a conformal map from a closed genus-one surface onto a rectangle planar domain by integrating a holomorphic 1-form [45] which describes two vector fields perpendicular to each other everywhere on surface; finally, we compose the conformal map from the rectangle to the canonical annulus using $e^{\frac{2\pi}{b}z}$ to get the Φ.

The algorithmic flow to map a topological annulus to a canonical planar annulus is detailed as follows:

3.2.5 Eliminating the Möbius Ambiguity

Conformal mappings between surfaces are not unique; e.g., all conformal mappings from a unit disk D^2 to itself form a Möbius group, with the form:

$$\tau : z \rightarrow w, w = e^{i\theta}\frac{z - z_0}{1 - \bar{z}_0 z}, z, z_0 \in \mathbb{C}, \theta \in [0, 2\pi),$$

where z_0 is a constant point, θ is a constant angle. All such τ form a 3 real dimensional group. Two mappings from a topological disk to a unit disk differ by a Möbius transformation, this ambiguity affects the signature and has to be eliminated via certain extra constraints.

Spherical Harmonic Mapping

In: Closed genus-0 surface M.

Out: Harmonic mapping Φ from M to unit sphere s^2.

1. Compute the normal vector for each triangle face. For each vertex, compute its normal $\vec{n}(v)$ as the weighed sum of normals on the adjacent faces weighed by their areas. Then set the initial map as the Gauss map: $\Phi(v) = \vec{n}(v)$.
2. Compute the Laplace-Beltrami operator at each vertex:

   $$\Delta(v) = \sum_{[u,v]\in M} w_{u,v}(\Phi(v) - \Phi(u)),$$

   the weight $w_{u,v}$ associated with edge $[u, v]$ is the well known harmonic weight, calculated as

   $$w_{u,v} = \frac{1}{2}(\cot(\alpha_{u,v}) + \cot(\beta_{u,v})),$$

   where $\alpha_{u,v}$ and $\beta_{u,v}$ are two angles opposite to the edge $[u, v]$ in the two triangles sharing the edge.
3. Project Laplacian $\Delta(v)$ onto the tangent space of $\Phi(v)$,

   $$\Delta_{s^2}(v) = \Delta(v) - (\Delta(v) \cdot \Phi(v))\Phi(v).$$

4. Update the map Φ, $\Phi(v) = \Phi(v) - \varepsilon\Delta_{S^2}(v)$ where ε is a small constant to assure the numerical stability. In our experiment we set $\varepsilon = 0.1$.
5. Normalize the map Φ by

   $$\Phi(v) = \frac{\Phi(v) - \mathbf{c}}{|\Phi(v) - \mathbf{c}|},$$

   where $\mathbf{c}$ is the weighted mass center:

   $$\mathbf{c} = \frac{\sum_v k_v \Phi(v)}{\sum_v k_v},$$

   where k_v is the summation of the areas of all faces adjacent to vertex v.
6. Repeat step 2 through 5 until it converges.

Mapping a Topological Annulus to a Canonical Planar Annulus.

```
In: Topological annulus $M$
Out: Conformal map $\Phi$ from $M$ to canonical planar annulus.
1. Double-cover $M$ to a closed genus one surface $\bar{M}$.
2. Compute a holomorphic 1-form basis of $\bar{M}$ by using the method introduced
   in [44]. Denote the basis as $\bar{\omega}$.
3. For an arbitrary path $\tau$ connecting $\Gamma_0$ and $\Gamma_1$, compute a holomorphic 1-form
   $\omega$ such that $\int_{\Gamma_0}\omega = 1, \omega = \frac{1}{\int_{\Gamma_0}\bar{\omega}}\bar{\omega}$.
4. Trace a vertical trajectory $r$ of $\omega$, such that $r$ is an integration curve of $\omega$ along
   imaginary direction. Namely, $r$ is iso-$u$ in the $(u,v)$ domain.
5. Slice $M$ along $r$ to get a fundamental domain $\widetilde{M}$, by integrating $\omega$, where $\widetilde{M}$
   is conformally mapped to a rectangle on the plane.
6. Conformally map the rectangle to an annulus with unit inner radius by $e^{\frac{2\pi}{b}z}$.
```

For closed genus-zero surfaces, we first fix a marker point p on the surface and define a tangent direction $\vec{t}_p$ going out from p. A closed curve Γ separates M into two disk-topology patches, the patch containing p is denoted as Γ_+. We require that Ψ_+ maps p onto the origin, and $\vec{t}_p$ onto the positive x-axis direction. These constraints uniquely determine Ψ_+.

For open genus-zero surfaces, we fix the marker p on the boundary. Ψ_+ maps Γ_+ to Δ_+, where Δ_+ is a canonical annulus with unit inner radius. The outer radius of Δ_+ is denoted as R, which is uniquely determined by the surface Γ_+. Furthermore, we require that $\Psi_+(p) = R$. Such Ψ_+ uniquely exists.

Through the above construction pipeline, every closed curve $\Gamma \in \Omega(M)$ corresponds to a diffeomorphism $\Psi \in Diff(S^1)$. Γ corresponds to two signatures Ψ_1, Ψ_2 if and only if there exists a Möbius transformation $\tau : D^2 \to D^2$, such that

$$\Psi_2 \circ \Psi_1^{-1} = \tau|_{\partial D^2}. \tag{3}$$

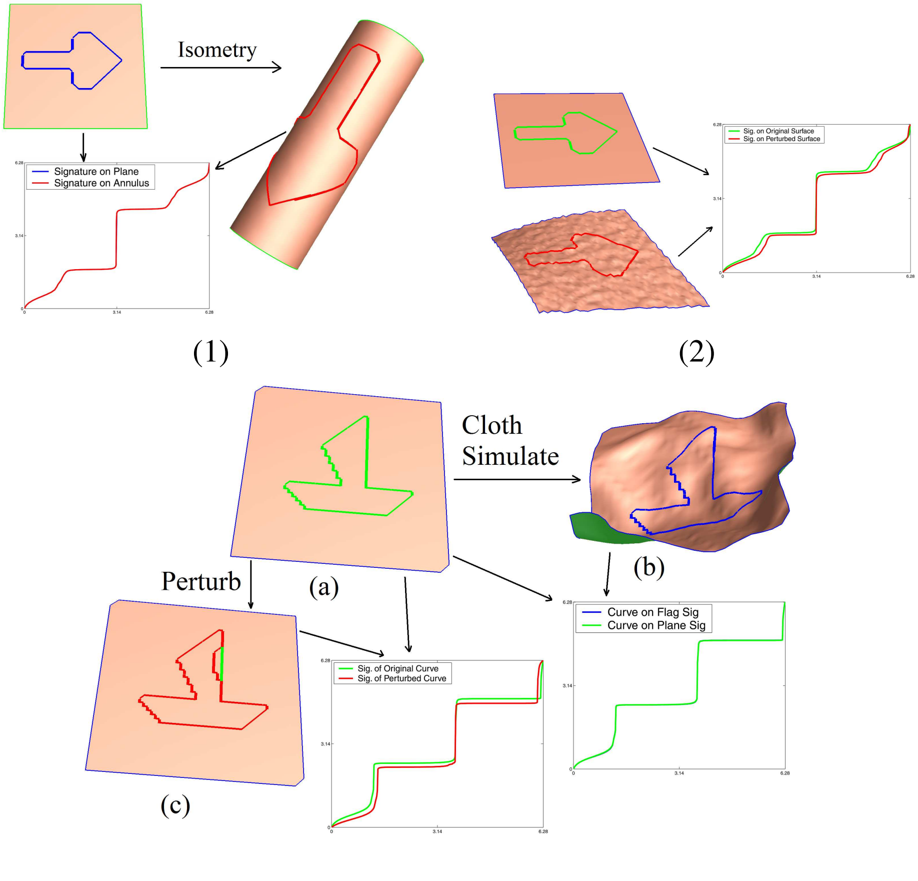
The above equation defines an equivalence relation $\sim$ in $Diff(S^1)$. We claim that the mapping

$$\Psi : \Omega(M) \to Diff(S^1)/\sim$$

is a one-to-one map. With appropriate metrics on $\Omega(M)$ and $Diff(S^1)$, it is a homeomorphism. In another word, each closed curve on M corresponds to an equivalence class of diffeomorphisms from the unit circle to itself.

In some scenarios, we might want to completely eliminate the ambiguity of signatures. For this purpose, we can further eliminate the Möbius ambiguity using more markers. To uniquely reconstruct a curve, Ψ and three markers are sufficient for the closed genus-zero surfaces while for the open genus-zero surfaces, Ψ and two markers are sufficient.

3.2.6 Distances between Curves



(1) (2)

(3)

Figure 3: The stability of curve signatures under isometry, perturbation and bending of embedded surfaces.

For a genus-zero surface M, we create signatures for curves defined on M. The deviation between two curves can be measured by the distance between their signatures using *Weil-Peterson metric* on $Diff(S^1)$ as introduced in [108].

If surfaces M_1 and M_2 are with similar Riemannian geometries in $\mathbb{R}^3$, then there exists a diffeomorphism $\phi : M_1 \to M_2$ close to an isometry, the induced map $\bar{\phi}$ between the signatures is close to the identity map from $Diff(S^1)$ to itself. In another word, if the curve $\Gamma_1 \subset M_1$ corresponds to $\Gamma_2 \subset M_2$ with $\Gamma_2 = \phi(\Gamma_1)$, then Γ_1 and Γ_2 have similar signatures. Hence, the signatures of curves have a property of strong stability under the Riemannian metric perturbation of their embedded surfaces and can be used to analyze curves on different surfaces as a robust tool.

Fig 3 demonstrates the stability of the signatures. All the curves and their corresponding signatures are drawn in the same color. Note that the signature is a diffeomorphism from a circle to itself, thus it can be considered as a periodic real function from $[0, 2\pi)$ to $[0, 2\pi)$, and only one period is shown in our figures. In (1), a planar rectangle is isometrically deformed to a cylinder, our computation shows that the corresponding curves have exactly the same signatures. In (2), the planar rectangle is perturbed about 6% in z direction, and about 1% in x, y directions, signatures of the corresponding curves are very close to each other. In (3), the planar surface in (a) is simulated as cloth and deforms as shown in (b), namely, it allows large bending but little stretching, the signatures of the corresponding curves are also almost identical(i.e., undistinguishable); also, the curve on surface in (a) is perturbed a little and shown in red curve in (c), the signature perturbs little.

Therefore, curves on different surfaces, which are close to each other in terms of geometry or differ by a near-isometric mapping, can be robustly and accurately compared and analyzed using their signatures.

3.3 Surface Matching

Based on analysis of curve space, we design our surface matching framework for curve alignment, surface registration, and shape comparison.

3.3.1 Feature Alignment for Surface Segmentation and Matching

We decompose the entire surface matching problem into two steps:

1. segmenting a surface via a set of feature curves and their alignments;
2. matching boundary curves and surface patch interiors.

The general framework is as follows.

Assume M_1 and M_2 are the two surfaces to be matched, if they share similar Riemannian geometries, meaning there exists a mapping $\phi : M_1 \to M_2$ close to an isometry, then the following algorithm can be used for matching.

Surface Matching Pipeline.

1. Extract the feature curves set $\{\Gamma^1_1, \Gamma^1_2, \cdots, \Gamma^1_n\}$ on M_1, which can be either marked by users as certain meaningful features, or automatically computed based on geometric information of M_1 such as the extremals of the principal curvatures along the corresponding principal directions.
2. Compute the curve signature of each Γ^1_i on M_1 and get the signature set $\{\Psi_1, \Psi_2, \cdots, \Psi_n\}$.
3. Compute the curve set $\{\Gamma^2_1, \Gamma^2_2, \cdots, \Gamma^2_n\}$ on M_2, such that the curve signature of Γ^2_i equals to Ψ_i.
4. The curve set $\{\Gamma^k_i\}$ segments M_k to several connected components $\{c^k_1, c^k_2 \cdots, c^k_m\}$, $k = 1, 2$, such that the boundaries of c^1_i correspond to the boundaries of c^2_i. Match c^1_i with c^2_i pairwise on the planar domain.

If in the step 3 above, users may prefer to label the meaningful feature curve set on M_2, we can change this step accordingly so that we compute, compare their signatures, and find the nearest one-to-one matching between these two sets of feature curves.

3.3.2 Surface Comparison in 2D Planar Canonical Domains

When all feature curves are matched, we segment the surfaces into several patches, each of which can be matched on the planar domain with many existing surface parameterization techniques [29].

A possible technique is to use the *conformal representation* [40], which consists of two functions $\lambda(u,v), H(u,v)$ defined on canonical domains. λ is called *conformal factor*, representing the area stretching of the mapping from the original surface to the planar domain and H is the mean curvature implying the bending information of the surface. In our experiments, we normalize the original surface and then compute its conformal factor at each vertex by dividing its one-ring-neighbor area on the surface by its counterpart on the planar domain. The conformal representation is complete, stable, and also convenient to compute in our framework. The matching energy E between two corresponding surface patches M_0 and M_1 is defined on their common canonical planar domains D by

$$E = \int_{(u,v)\in D} ||\lambda_0(u,v) - \lambda_1(u,v)||^2 dudv + \int_{(u,v)\in D} ||H_0(u,v) - H_1(u,v)||^2 dudv.$$

We can visualize the matching result by color-coding the difference distribution measured with this term. These three reasons that we use conformal representation are specialized as follows.

First, the conformal representation is complete in the sense that it allows us to fully reconstruct the original surface from the representation according to [40].

Theorem 3 (Conformal Representation) *If a surface $S(u,v)$ is parameterized by some conformal parameter (u,v) on a domain D, then the conformal factor function $\lambda(u,v)$ and mean curvature function $H(u,v)$ defined on D satisfy the Gauss and Codazzi equation. If $\lambda(u,v)$ and $H(u,v)$ are given, along with the boundary condition $S(u,v)|\partial D$, then $S(u,v)$ can be uniquely reconstructed.*

Second, according to [40], conformal representation stably represents the geometry distance between surfaces in $\mathbb{R}^3$; the perturbation in geometry leads to stable and continuous perturbation in their conformal representations.

Third, as a by-product, the computation process of curve signatures has already computed conformal maps from most $3D$ patches to the planar domains, so the surface matching based on these mappings can be done without further computation cost.

Figure 4 shows an example on how to make use of conformal representation for surface comparison. A unit disk planar surface M_0 as shown in (a) is compared with a center-bulb surface M_1 shown in (b). The conformal factor and mean

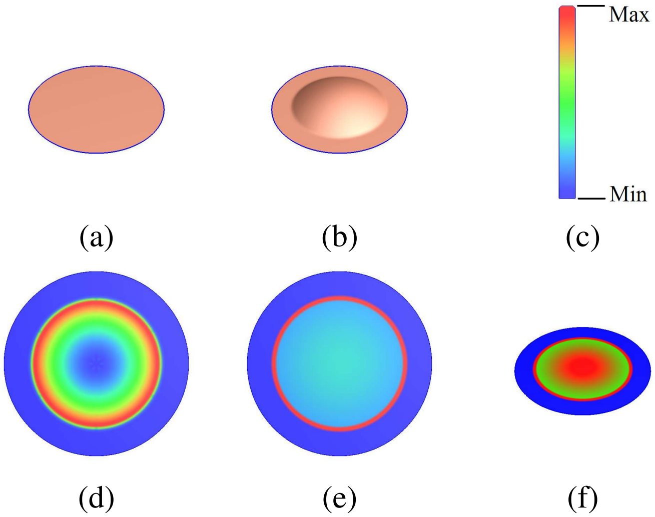


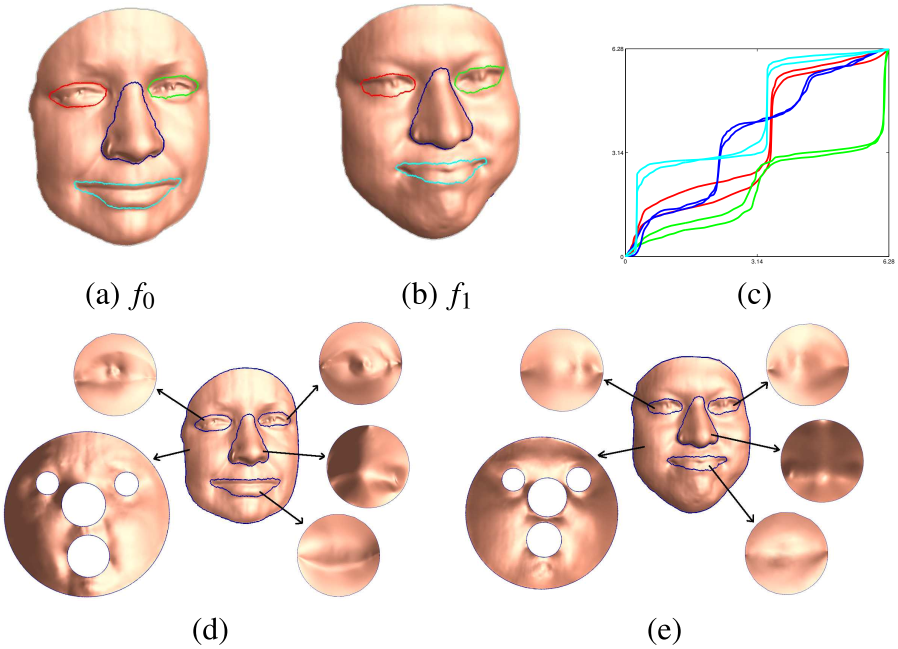
Figure 4: Conformal representation. (a) and (b) are surfaces to be compared, (c) is the color scheme we use in this work, (d) and (e) are conformal factor and mean curvature of (b) drawn in 2D planar domain, and (f) is the conformal representation difference distribution between (a) and (b).

curvature of planar surface is constant everywhere; the conformal factor and mean curvature of surface (b) in 2D planar domain are color coded and shown in (d) and (e); the deviation $d(u,v) = (\lambda_0(u,v) - \lambda_1(u,v))^2 + (H_0(u,v) - H_1(u,v))^2$ between the matched surfaces are color-coded in surface M_0 and shown in (f).

3.4 Experimental Results

3.4.1 Human Faces

To illustrate our framework, we firstly use a human face matching example. Two human faces(f_0(female) and f_1(male) as shown in Fig 5 (a) and (b)) are compared by aligning feature curves enclosing eyes, noses and mouths. Assuming that the geometries of human faces are similar, namely, there exist mappings $\Phi : f_0 \rightarrow f_1$ that are close to isometry, we manually label on each face four feature curves and compute their signatures. The curves and their signatures are highlighted with the same color. For example, curves enclosing the right eyes and their signatures are colored in red. As shown in Fig 5 (c), signatures with the same color are quite



(a) f_0 (b) f_1 (c)

(d) (e)

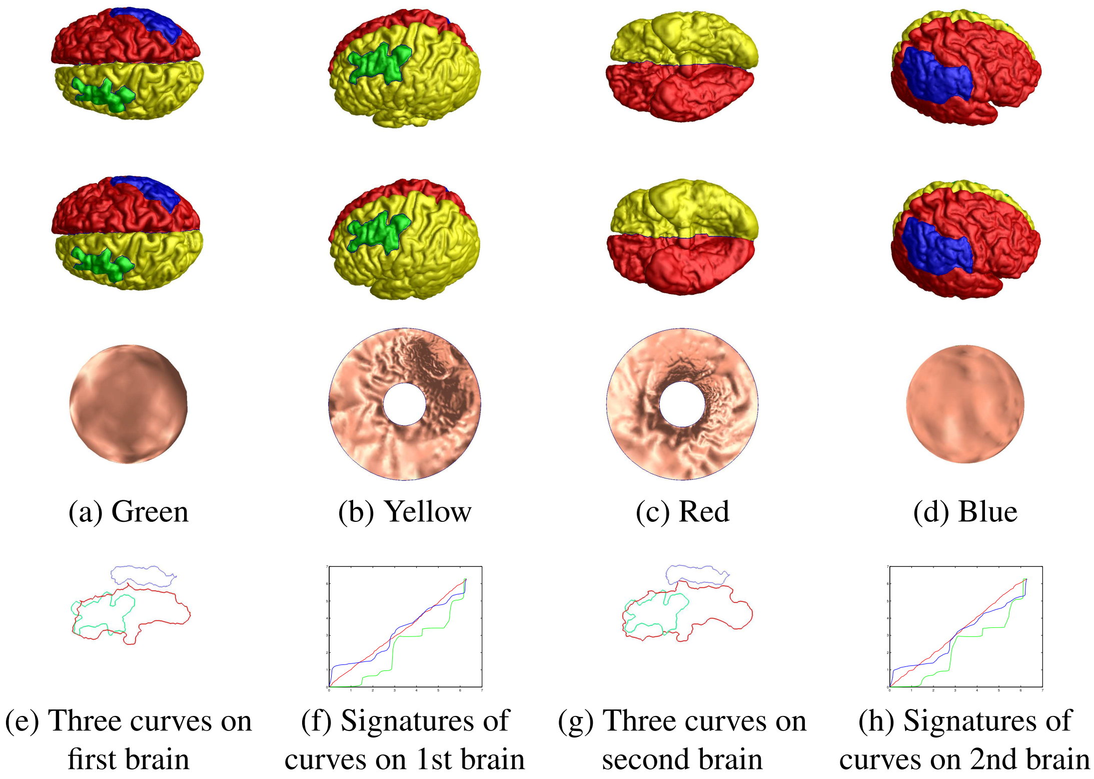
Figure 5: Curves on faces((a),(b)), their signatures(c), and the segmentations for the matching purpose((d),(e)).

similar to each other.

The experiment shows that similar feature curves on two faces have similar signatures, while different feature curves on the faces have dramatic different signatures. Therefore, the curve signature is a reliable tool to align the same features across different faces. The faces can then be segmented and mapped onto common canonical planar domains for subsequent registration and comparison, as shown in Fig 5 (d) and (e).

3.4.2 Brain Cortex Analysis

Another example is brain cortex comparison, we locate feature curves segmenting the whole surface into disks and annuli. These features are functional “landmarks” given by users. Our practical example is for medical imaging: The two cortex surface data are reconstructed using MRI images of one paralytic acquired at different times. The feature curves are manually labeled by the clinical doctor who is monitoring the recovery of this patient’s brain. It is desirable to compare


(a) Green

(b) Yellow

(c) Red

(d) Blue

(e) Three curves on first brain

(f) Signatures of curves on 1st brain

(g) Three curves on second brain

(h) Signatures of curves on 2nd brain

Figure 6: Surface match on brains using curve features similarity comparison. The first two rows show the first and second brain from four different view directions. The third row show the mapping results from four different subregions of the first brain to canonical domains. The last row shows the curves on two brains and their signatures.

the cortex surfaces such that the corresponding curves and regions are matched. Noted here although these two brains are similar in terms of Riemannian metric, they cannot be matched simply via a rigid transformation. [1]

We apply our matching procedure as explained above. The feature curves for the first and the second brain are shown separately by the first row and the second row in figure 6 from different view directions. Feature curves and their corresponding signatures are shown in (e),(f) and (g),(h). By comparing their signatures, each curve on the first brain is mapped to the curve on the second brain with the same color. The curves segment the cortex surfaces to four components, each of which is

[1] Because the cortex surfaces are highly convoluted, two points on the surface with small Euclidean distance in $\mathbb{R}^3$ may have huge geodesic distance on the surface.

either a topological disk or an annulus; the segmentation is color encoded as shown in the first two rows in Figure 6.

Each component on the cortex surface is conformally mapped to either the unit disk or the canonical planar annulus. Figure 6 (a)-(d) show the conformal mappings for the 4 components of the first cortex surface. Similarly, the components on the second cortex surface are conformally mapped to the unit disks or canonical annuli. By matching these canonical planar domains, the map between two cortex surfaces can be easily induced using the existing method such as [40].

3.4.3 Elephant Gallop

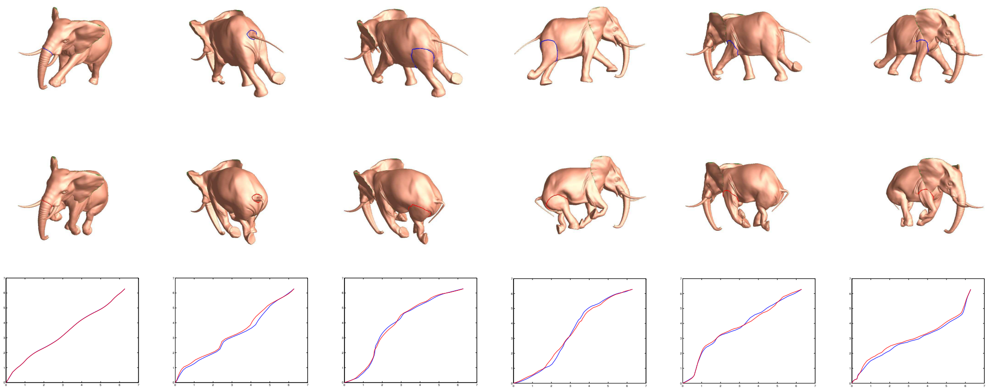


Figure 7: Feature curves and their signatures on two elephant models with different postures. The first row shows one geometric configuration of the elephant from different view directions; the second row shows another model from different view directions; the third row shows the signatures. Note that, each column shows a special curve on two models and their signatures, and the curve is depicted with the same color as its signature. In each column, the red and blue signatures are almost identical so that they overlap and are not distinguishable.

We use an elephant gallop example to further evaluate our curve signatures and our surface comparison framework. As shown in Figure 7, there are two models of one elephant in different postures. Suppose we want to compare these two models, we first label feature curves which segment the elephants into several parts. We compute signatures for all curves on both surfaces, as shown in the third row. Every signature of curve on one surface is matched to the most similar signature

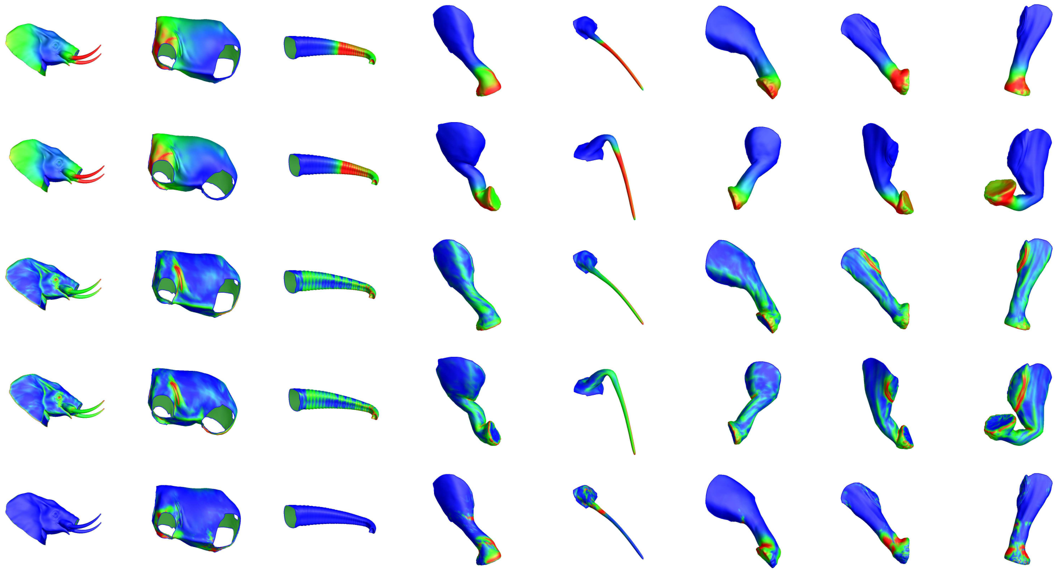


Figure 8: Segmented parts from the elephant model and their color-coded function distributions, highlighting their intrinsic differences. The first row shows the conformal factor function distributions mapped on the original surface of the first model; the second row shows the conformal factor distributions of the second model; the third row shows the mean curvature distributions of the first model; the fourth row shows the mean curvature distributions of the second model; the last row color-codes the difference of conformal representation between two models.

of curve on another surface. The matched pairs are all placed in the same column, and each feature curve and its corresponding signature are drawn in the same color. The experiment results demonstrate that the correct matching can be induced automatically without human intervention. This attractive property on curve signatures results from the fact that the signatures for corresponding curves are very similar, and underlying reason of this fact is that the skin deformation is very close to isometry because the stretch of skin under these kinds of deformation is relatively small.

Once the corresponding feature curves are matched, the surfaces are segmented into several parts with explicit correspondence established by the segmenting curves. These parts are then considered separately on their own canonical planar domains, as shown in Figure 8. On each domain, we use the stretching and bending functions to compare their differences. The conformal factor and mean curvature are computed and colorized in the original surface to show the function value distributions. We color-code the conformal factor of the first model in the first row, and color-code this model's mean curvature in the second row; similarly we color-code the conformal factor and the mean curvature of the second pose in the third and fourth row. The matching difference between two surfaces based on the functions on 2D domains are color-coded on the first pose and shown in the last row. The color-code scheme is the same as in the previous example (Figure 4 (c)), where red represents the max value while blue is for the min value. Note that, the last column is color coded in one uniform scheme. And the results shows that largest stretching and bending differences locate on leg joints and ankles.

3.4.4 The Collapsing Horse

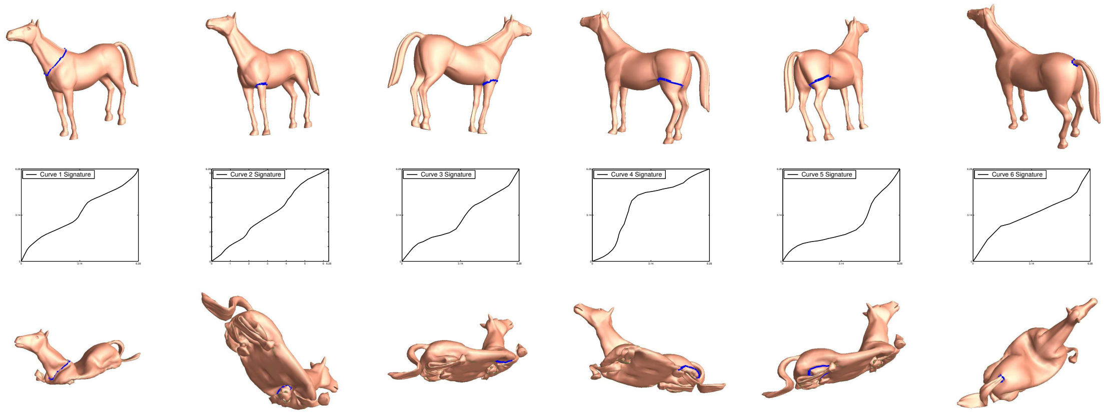


Figure 9: The first row views feature curves on the standing-horse model; the second row shows their corresponding signatures; the third row displays the accordingly reconstructed curves on the collapsed-horse model.

The next experiment is to compare a horse and its collapsed pose. Users first mark feature curves on one pose. With the signatures we could reconstruct the curves on the second surface. Techniques introduced in [108] can be used to reconstruct the curve on the complex domain, which corresponds to a unique curve on the spherical domain. Combined with three predefined markers introduced in section 3.2.5 and the mapping from the original surface to the sphere, the unique curve on the original surface can be reconstructed. With this process, feature curves can

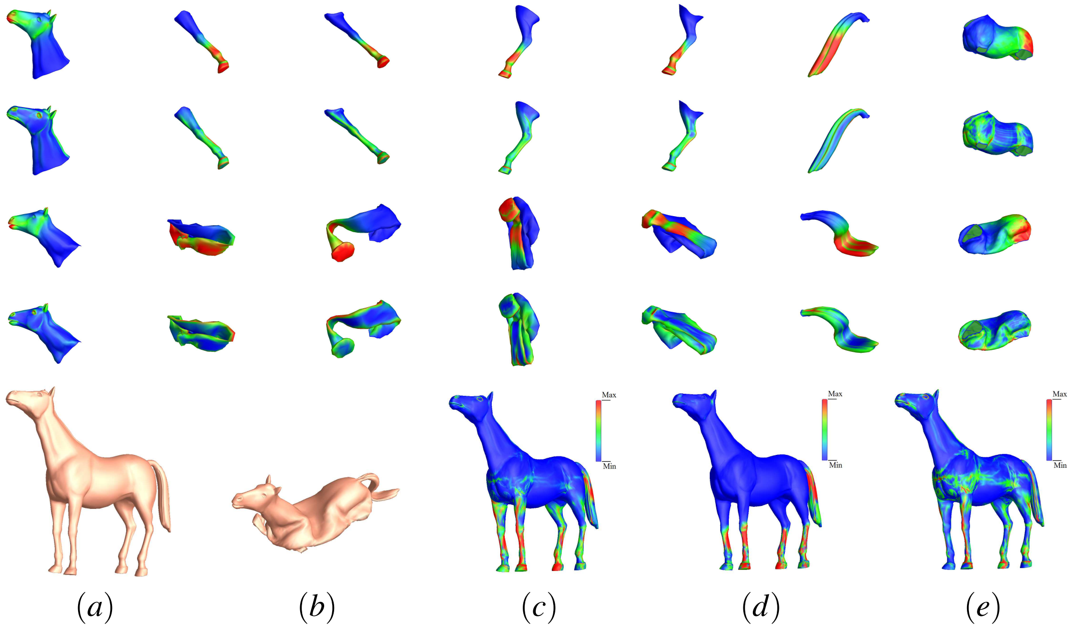


Figure 10: The first and the second row color-code conformal factor λ and mean curvature H of the standing horse model; the third and fourth rows are of the collapsed horse model; the last row shows the final matching results between the standing model (a) and the collapsed model (b), with (c)-(e) color-coding differences on conformal representation, λ, and H respectively. (Mesh size: 17k Triangles)

be transferred onto the second object as shown in Fig 9. The original feature curves on rest pose, their signatures, and the transferred curves are shown in the three rows in Fig 9 respectively.

The conformal factor and the mean curvature distributions of all parts are computed and color-coded in the first four rows of Fig 10 (the first two rows are for the standing pose, while the third and fourth rows are for the collapsed pose).

The surface comparison framework can be interactively controlled by changing weights of the two terms in our matching energy. For example, if isometry-invariant comparison is preferred, only stretching factor needs to be considered. So by ignoring the mean curvature, a metric invariant under bending is designed, which naturally leads to a bend-invariant or pose-invariant result. The conformal representation difference between the two horse models (a) and (b) is color-coded on the first model as shown in Fig 10 (c) and the difference ignoring the bending term is shown in Fig 10 (d); also, the difference with only the bending term is color-coded in (e). As shown in the above examples, our matching algorithm finds between two complicated objects a difference distribution which can be flexibly adjusted for different goals such as the bending-invariant purpose. Since it can determine the difference on the metric ignoring the embedding of the surface in $\mathbb{R}^3$, it becomes a useful tool for non-rigid matching applications. One example is colon matching and analysis in medical imaging. People with different poses under CT scans might have large bending differences on their colons with little changes in metric, in which case such a bending-invariant matching is ideal for the analysis purpose.

3.5 Chapter Summary

We have designed a metric space for simple closed curves on genus-zero surfaces via conformal mappings. Curves on surfaces are represented by equivalence classes of diffeomorphisms of the unit circle to itself. The proposed curve signature corresponds uniquely to the curve defined on a surface. It also includes information of both the curve's shape and its embedding on the surface, which are invariant under isometry and stable under near-isometric transformation of surfaces, thus enables a powerful practical tool for the effective analysis of curves and surfaces among geometrically similar objects.

Besides the above theoretical results, we develop a framework for shape registration and comparison guided by feature curves alignments. After curves with the most similar signatures are correctly identified and aligned, genus-zero surfaces are then segmented into several parts and registered separately. This automatic process accurately forces the alignment of feature curves and alleviates the difficulties of 3D surface matching by reducing it to the simple comparison of functions defined on canonical planar domains. Also, the algorithm can be flexibly adjusted to provide a pose-invariant shape descriptor.

Constructing shape space of curves on surfaces with arbitrary topology is promising and challenging. We plan to explore further along these directions.

The shape mapping framework based on our curve signatures is semi-automatic and far from perfect. Even segmentation can be consistently transferred from one surface to another, we will need segmentation to be provided on one surface at least. We will discuss this in the next chapter, where we present a more automatic and flexible framework that generates consistent segmentation on different surfaces canonically.

Chapter 4

Canonical Pants Decomposition for Piecewise Surface Mapping

In Chapter 3, we studied curves on surfaces so that segmentation among different surfaces can be consistently **matched** or **transferred**. In this chapter, we present an automatic decomposition scheme so that segmentation on different surfaces with complicated topology can be consistently **generated**.

4.1 Introduction

As we discussed in Chapter 2 and Chapter 3, current state-of-the-art surface mapping techniques (including the method we presented in Chapter 3) are far from adequate and perfect. A more desirable and powerful surface mapping method is needed and should have the following properties.

The first one is **generality**. The mapping methodology should be general, i.e., it should be able to handle surfaces with arbitrary topology, with or without boundaries. The generality also includes another important issue – being capable of accommodating topology changes. We can see the importance of topology change in surface mapping from its applications. When we use surface mapping for shape comparison and difference analysis, the data to be registered could easily have different topologies due to shape variations and accompanying noise (e.g., small boundaries and tiny handles). Moreover, when we use surface mapping to drive the animation of a morphing sequence, we usually transform one object to another based on their intrinsic semantics, regardless of whether they have the same topology or not (see Fig. 20(a)).

Many existing surface mapping techniques primarily focus on genus-zero surfaces, most recent works start to aim at general surfaces, and much fewer techniques have been devised to flexibly work for arbitrary topological changes. In this work, we aim at a general framework that can handle arbitrary mesh inputs.

The second property is **automation**. Most current surface mapping techniques heavily rely upon a large amount of user intervention. Although in many applications, the requirements of object semantics forbid us from entirely ignoring user intentions. The primary reason for the lack of fully automatic methods in this research field still stems from technical difficulties. Real-world shapes could be complex in both topology and geometry. To our best knowledge, if the given surfaces are topologically non-trivial (neither sphere-like nor disk-like), even with the same topology, no existing techniques are able to compute the mapping in a fully automatic way. A key difficulty stems from that, although most current mapping methods depend on a preprocessing stage of mesh segmentation, few surface segmentation techniques have been devised for automatically providing consistent segmentation on different surfaces.

When mapping is used in applications dealing with large amount of data, such as analysis and comparison on shapes in database, user involvement on every registration trial could not be practical. Therefore, we definitely need a surface mapping technique that works for general inputs, yet is as automatic as possible.

The third property is **controllability**. Although automation makes the mapping process much less labor-intensive, in real applications where the semantics plays a critical role, such as morphing (requiring feature points matching), automatic methods based on pure topology and geometry inevitably fail. We must have a new mechanism that can provide an easy way to let the user manage the behavior according to semantics-specific requirements. Indeed, current surface mapping techniques oftentimes provide limited control to the user; but for surfaces with complicated topology, they either require a large number of markers [73] or need user's great efforts to design the base mesh as a good starting point [18], [35]. In principle, a good mapping framework should provide an intuitive and easy-to-use human computer interaction.

Fourth, it is also important to emphasize **rigorousness**. The global continuity is typically required for the underlying mapping. However, between given surfaces, there may exist many continuous yet topologically different mappings, i.e., mappings could have different homotopy types (see Fig. 24). Two surface mappings belong to the same homotopy type if and only if they can continuously deform to each other without degeneracy. Among so many legitimate choices, there are no viable ways to select the best ones from all candidates, since different homotopy may represent different semantics. In such a case, being able to let user easily and intuitively determine arbitrary topological type of a mapping not only demonstrates the rigorousness of the mapping algorithm, but also has practical importance.

In this chapter, we present a piecewise surface mapping framework in order to unify the above four properties. We conduct our experiments on several challenging examples to demonstrate the power and potential of our method. Our **contributions** are as follows.

1. Our framework efficiently handles surfaces with arbitrary topology, with or without boundaries. It also handles surface mapping with topology changes.
2. Our framework has great automation. Our pants decomposition can automatically compute consistent segmentation on a set of surfaces with same but complicated topology. Besides the selection towards the semantics purposes at the preprocessing stage, the afterward framework can proceed without any user intervention, and therefore provides canonical decomposition for automatic matching among a large number of acquired or synthetic datasets.
3. When user interaction is necessary for any semantics reasons, our framework coherently aligns constraint points or curves to enforce constraints, and provides users a simple and intuitive mechanism to control the mapping behavior.
4. In practice, our framework generates and enumerates any different homotopy types of mappings. It shows not only the flexibility but also the rigorousness and completeness of our mapping framework from the mathematical point of view.
5. Our algorithm is simple and efficient. As we will elaborate later, the technical core of the decomposition algorithm primarily relies on the Dijkstra algorithm, and only the triangular metric of given surfaces is employed.

The remainder of this chapter is organized as follows. In Section 4.2, we introduce theoretic background as well as the terms and definitions necessary for this work. The fundamental idea of our framework is illustrated in Section 4.3, which is a two-step pipeline, as discussed in Section 4.4 and Section 4.5, respectively. Finally, we demonstrate our experimental results in Section 4.6.

4.2 Theoretical Foundation

4.2.1 Definition of Pants Decomposition

We briefly introduce the related background in topology and geometry and make necessary definitions in this section.

A *surface* M is a topological Hausdorff space in which each point has a neighborhood homeomorphic to either the plane or the closed half-plane. Points with closed half-plane neighborhood are defined as the *boundary* of M.

A *path* is a continuous map $p : [0,1] \to M$. A *loop* (cycle) is a closed path, meaning that the endpoints $p(0)$ and $p(1)$ coincide. The *concatenation* of two paths p and q, with $p(1) = q(0)$ is the path $p \circ q$ defined by

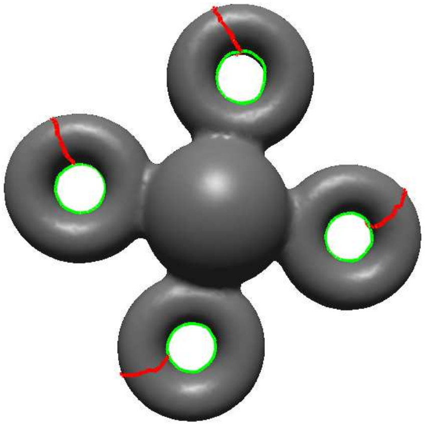
$$(p \circ q)(t) = \begin{cases} p(2t), & t \leq 1/2; \\ q(2t-1), & t \geq 1/2. \end{cases}$$

When we say two paths are *homotopic*, it means one path can continuously evolve to the other one through a family of paths on the surface. Rigorously speaking, a homotopy between paths p and q is a continuous map $h : [0,1] \times [0,1] \to M$ s.t. $h(0,\cdot) = p$, $h(1,\cdot) = q$, $h(\cdot,0) = a$, $h(\cdot,1) = b$, where a and b are two paths joining $p(0)$ with $q(0)$ and $p(1)$ with $q(1)$, respectively. We denote the homotopy equivalence class of path p as $[p]$.

All homotopy classes under the product $[p] \circ [q] = [p \circ q]$ form a group called the *fundamental group*, denoted as $\pi_1(M)$. Suppose $f : M \to M'$ is a continuous map, p is a loop on M, then $f \circ p$ is a loop on M'. f maps the homotopy class $[p]$ to the homotopy class $[f \circ p]$, and f induces a homomorphism $f_* : \pi_1(M) \to \pi_1(M')$. Suppose $f^1, f^2 : M \to M'$ are two continuous maps between M and M', we say f_1 and f_2 are homotopic, if and only if they induce the same homomorphism between the fundamental groups $f_*^1 = f_*^2$.

A pair of *pants* is a genus-0 surface with 3 boundaries. A *pants decomposition* of M is an ordered set of simple, pairwise disjoint cycles that splits M into pairs of pants. Every compact orientable surface, except the sphere, disk, cylinder, and torus, admits pants decomposition. If M is of genus G and has B boundaries, a pants decomposition is made of $3G+B-3$ cycles [50]. In this work, we present an automatic decomposition algorithm to cut surface apart iteratively along certain non-trivial loops. The $3G+B-3$ cycles segment the given surface M apart to $2G-2+B$ pairs of pants $(M_0,\dots,M_{2G-2+B-1})$. Each pair of pants M_i (for simplicity, we also call such a surface patch a *pants patch* in the remaining part of this document) has three boundaries, which are denoted as the waist Γ_i^0, and two legs Γ_i^1,Γ_i^2. Two pants M_i and M_j are adjacent if they share boundaries.

4.2.2 Handle and Tunnel Loops



Suppose a closed embedded surface $M \subset \mathbb{R}^3$ separates $\mathbb{R}^3$ into a bounded space $\mathbb{I}$ and an unbounded space $\mathbb{O}$. Handle and tunnel loops of M can be defined as follows (see also [22]). A loop b_i is a handle if it spans a disk in the bounded space $\mathbb{I}$; if one cuts M along b_i and fills the boundary with that disk, one eliminates a handle. A loop a_i is a *tunnel* if it spans a disk in the unbounded space $\mathbb{O}$; and its removal eliminates a tunnel. The handle and tunnel loops characterize important topological information of the surface, and we use them to determine the homotopy types of our mappings. An intuitive illustration is shown in the above figure. Red curves represent the handle loops while the green ones are tunnel loops. More details about handle and tunnel loops as well as their automatic computation algorithm can be found in [22]. All handle loops form a basis of $\pi_1(\mathbb{I})$, and all tunnel loops form a basis of $\pi_1(\mathbb{O})$. The union of their homology classes form a basis of $\pi_1(\mathbb{M})$. Tunnel loops and handle loops can be effectively computed using techniques presented in [22]. Our algorithm takes surfaces, their handle and tunnel loops as input.

4.3 Overview of Key Ideas

The core of our mapping framework is consistent pants decomposition. Given an arbitrary genus-G surface with B boundaries, pants decomposition provides a canonical segmentation, partitioning this surface into $2G+B-2$ pants patches (see the followings for more detail). Given a set of arbitrary surfaces of the same topology, our pants decomposition scheme can automatically segment all of them into consistent sets of patches with "pants" topology.

Decomposing a Surface. The first step of our surface mapping is pants decomposition. It segments the given surface into a set of pants. This decomposition can process canonically in an automatic way, i.e., once the indexed handle and tunnel loops ($a_i, b_i, 0 \leq i < G$) are provided in a preprocessing stage, then the decomposition is unique and we will obtain an ordered set of pants.

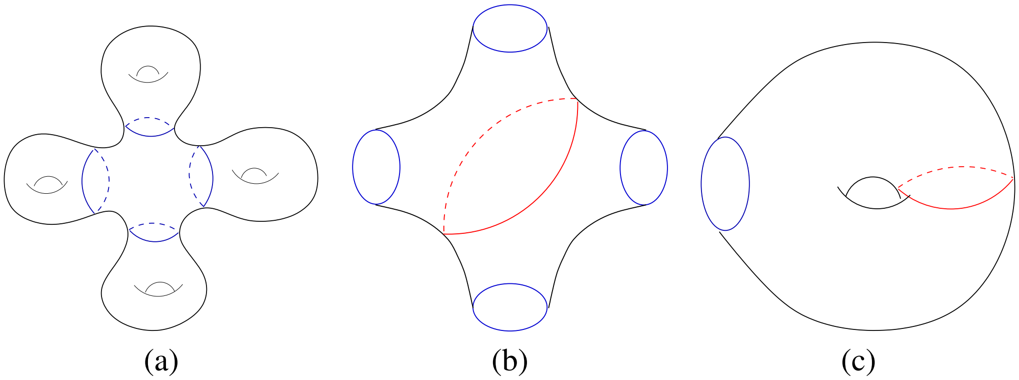


Figure 11: Pants Decomposition Pipeline. (a) Find and remove "waists" of handles. (b) and (c) Decompose the base patch and handle patches.

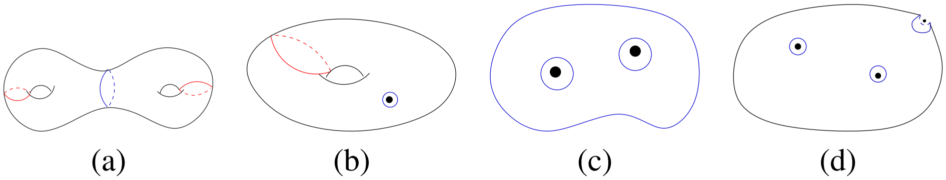


Figure 12: Pants Decomposition on Surfaces with Simple Topology. (a) A genus-2 surface has 2 handle patches, no base patch. (b) A closed genus-1 surface ($\chi = 0$) needs one punch. (c) A topological disk ($\chi = 1$) needs two punches. (d) A topological sphere ($\chi = 2$) needs three punches.

To give an intuitive overview, we start from a closed genus-G ($G \geq 2$) surface M. In Fig. 11, a genus-4 torus example is used to illustrate key steps of our pipeline. First, we remove G handle patches from M (a), and get a set of genus-one surfaces $M_i, (0 \leq i < G)$ with one boundary and (if $G \geq 3$), a topological sphere M' with G boundaries. We call M' the *base patch* and these boundaries *waists*. Second, we decompose the base patch M' and all the handle patches M_i into pants patches ((b) and (c)). For genus-2 surfaces, no base patch exists, 2 handle patches M_0 and M_1 compose the segmentation (Fig. 12(a)).

When surfaces have high genus ($G \geq 2$) with boundaries, we leave boundaries on the base patch M', treat them as usual “waists”, and apply base patch decomposition similarly.

We can also decompose surface M with easier topology ($G < 2$) with some extra “holes”. The basic idea is, the Euler number of a pair of pants is -1, so if M’s Euler number $\chi = 2 - 2G - B$ is negative (for example, $G \geq 2$ will guarantee $\chi < 0$), M can be decomposed directly. Otherwise, we punch holes on M until M gets a negative Euler number. One punch decreases the Euler number by 1, and since the Euler number of a surface can not exceed $+2$, we at most need 3 punches.

More specifically, if M is genus-1 and has a boundary, $\chi_M = -1$, it is directly processed as a handle patch M_i. If the surface M is genus-1 and closed (Fig. 12(b)), $\chi_M = 0$, one marker is required from the user. We punch a hole on the marker, get a boundary and make $\chi_M = -1$ so that it can also be decomposed like M_i. If M is a genus-0 mesh with a boundary (Fig. 12(c)), like a disk, then $\chi_M = 1$. We already have one “waist”, and need two more punches as “legs”. If the surface is a closed genus-0 mesh (Fig. 12(a)), three markers are required to form a pair of pants.

When a genus-zero or genus-one surface has more than one boundary, similarly we compute its Euler number and check whether extra punches are necessary. As we will discuss in the later part of this chapter, these markers can be used as feature points in the surface mapping because their exact correspondence is guaranteed.

Pants Mapping. When surfaces are decomposed into a set of sub-regions, each with “pants” topology, the mapping computation becomes easier. To make

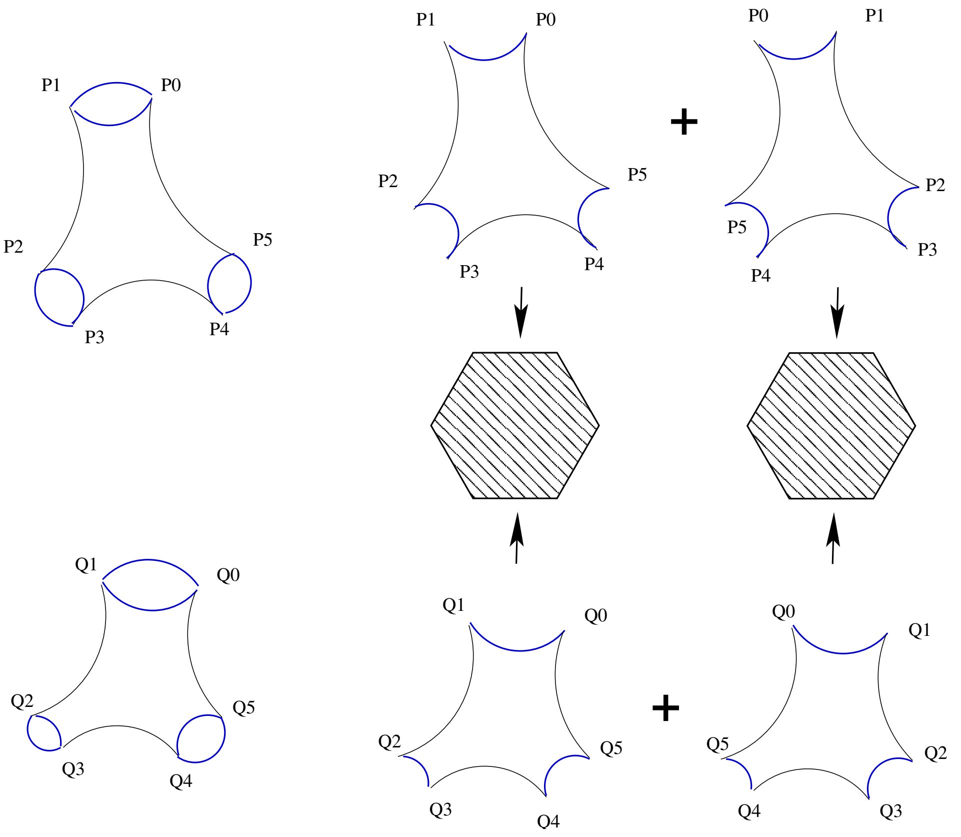


Figure 13: Mapping Two Pants Patches. Each pair of pants is decomposed into two topological hexagon patches. Harmonic maps from these patches to regular hexagons are used to compose the pants mapping.

sure the map has global continuity and bijectivity, boundary mappings on neighboring sub-regions have to be consistent. Many mapping techniques with fixed boundary conditions, such as harmonic map [24], mean value coordinate map [28] and so on can all be the choice for pants mapping. Free boundary mapping techniques are not preferred for sub-regions mapping here because if we cannot control the boundary mapping behavior, it will fail to satisfy continuity and bijectivity along the sub-regions' boundaries.

In this work, we use the harmonic map, because it is physically natural and can be computed efficiently. Like other fixed boundary mapping techniques, the shape of the target regions should be convex to guarantee the map's existence and validity. Such a direct convex domain for shapes with pants-topology does not exist; therefore, we simply decompose the pants into two patches with disk-like topology to make the mapping computation stable. As illustrated in Fig. 13, each pants patch is decomposed into two topological hexagons, and each hexagon is harmonically mapped to a regular hexagon. The pants mapping is then composed through these hexagonal domains.

4.4 Consistent Pants Decomposition

This section introduces our algorithm and implementation of the consistent pants decomposition on surfaces. The algorithm pipelines are introduced in Section 4.4.1, Section 4.4.2, and Section 4.4.3, respectively. To allow topology changes in mapping, surgery points are introduced in our decomposition framework (Section 4.4.4). Users can control the mapping by setting feature points or curves, we describe the related issues in Section 4.4.5 and Section 4.4.6. Our decomposition process is very robust, as addressed in Section 4.4.8.

4.4.1 Removing Handles

The first step of the pipeline is to remove handle patches from a given surface M. We iteratively trace a special shortest cycle and slice M along it to separate a handle patch (which becomes a pants patch M_i later) from M. We denote such shortest cycle as w_i, indicating it is the "waist" of M_i.

Suppose the handle loop and tunnel loop of the current handle are denoted as a_i and b_i, then the cycle w_i is homotopic to $c_i = a_i^1 \circ b_i^1 \circ a_i^{-1} \circ b_i^{-1}$. The computation of w_i is not trivial, we illustrate it using the following example.

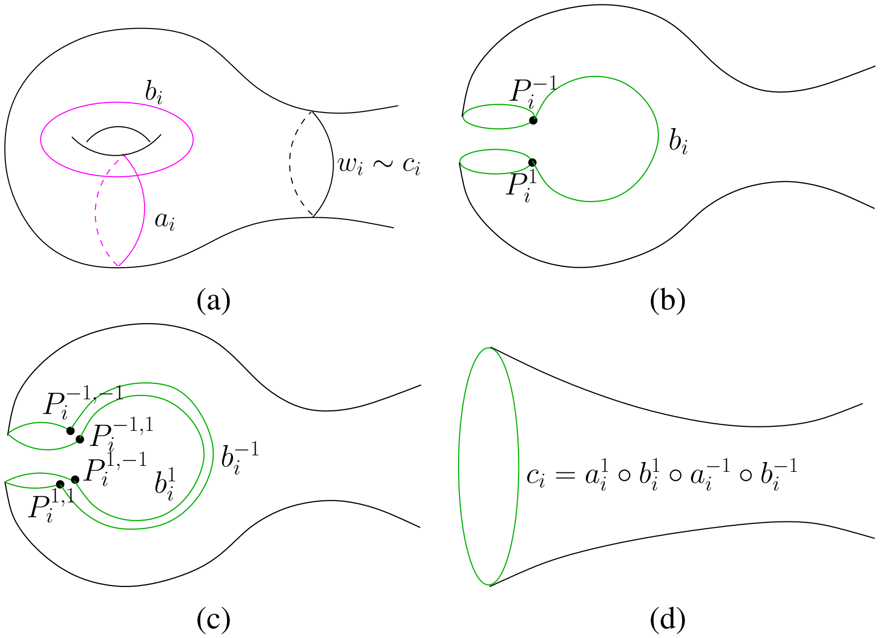


Figure 14: Computing c_i of the handle i. (a) w_i is the waist, but we need to compute c_i first. (b) Slice a_i apart, get P_i^1 and P_i^{-1}. (c) Slice b_i apart, P_i^1 and P_i^{-1} split to $P_i^{1,1}$, $P_i^{1,-1}$, $P_i^{-1,1}$ and $P_i^{-1,-1}$ separately. (d) The newly generated boundary is c_i.

Step One: Compute c_i.

Fig. 14 shows a surface patch near the handle and illustrates the idea. In this step, we compute the curve $c_i = a_i^1 \circ b_i^1 \circ a_i^{-1} \circ b_i^{-1}$ which is homotopic to w_i. As (b) and (c) show, we slice a_i and b_i apart along their intersection point, and get the resultant green boundary in (d): $c_i = P_i^{1,1} \rightarrow P_i^{1,-1} \rightarrow P_i^{-1,-1} \rightarrow P_i^{-1,1} \rightarrow P_i^{1,1}$.

Step Two: Shrink c_i to the "Waist" w_i.

As shown in Fig. 15, in step one, we sliced apart M ((a)) and get all its c_i ((b)). Now, we iteratively shrink each c_i to its shortest homotopic cycle w_i. It is computed through the following algorithm 4.4.1:

When we get w_0 from c_0, we remove the sub-region bounded by c_0 and w_0 from M, and denote it as M_0. We go on processing the above algorithm on other $c_i, i = 1 \ldots G-1$ on the new M, only substituting c_0 by w_0 as shown in (d).

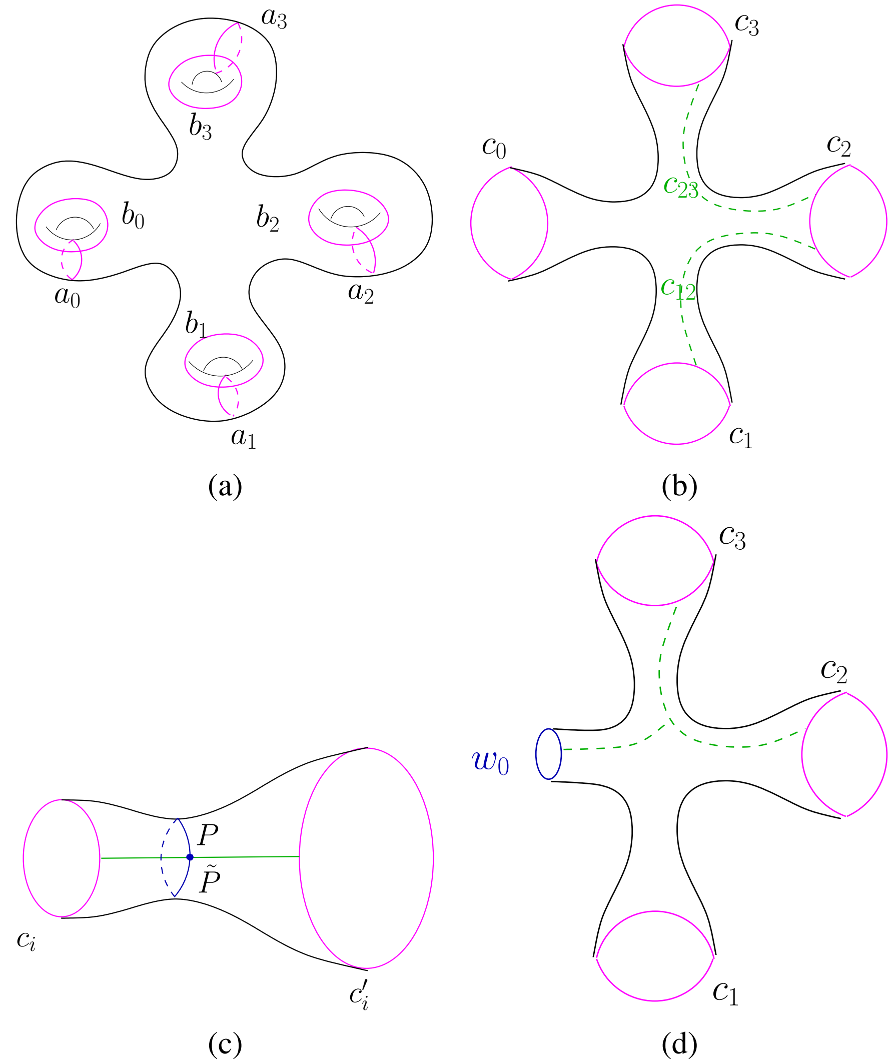


Figure 15: Computing the "Waist". (a) Slice M apart along its handle/tunnel loops, get boundaries c_i. (b) Connect all other boundaries $j, (j \neq i)$ to a large boundary c'_i, and get a topological cylinder. (c) Slice apart γ (green) connecting c_i and c'_i; get a topological "trapezoid"; compute w_i (blue) as the shortest path connecting boundary point pairs. (d) Continue the process on other handles.

This process ends after G steps, and we get G handle patches $M_0 \ldots M_{G-1}$. Each waist w_i is the shortest cycle on $M \setminus \cup_{j=0}^{i-1} M_j$, making the segmentation geometrically optimal under this order. We also get a leftover patch M, which is a

Shrink c_i to w_i.

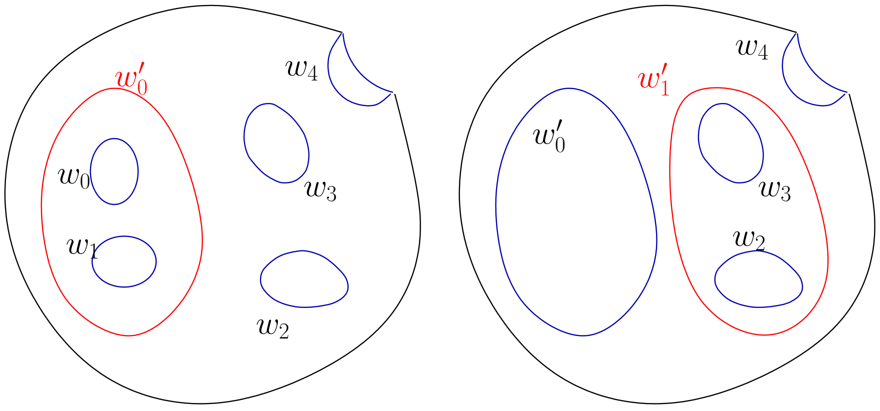
In: Surface M, c_i.

Out: The shortest loop w_i homotopic to c_i.

1. Connect all existing boundaries except c_i together using shortest paths (dashed green curves in Fig. 15(b)).
2. Slice these paths apart, we get one new large boundary c'_i (Fig. 15(c)). M becomes a topological cylinder.
3. Compute the shortest path γ (green curve in Fig. 15(c)) connecting the cylinder's two boundaries.
4. (The shortest cycle w_i at least intersects once with γ) Slice γ apart, every point $p \in \gamma$ splits to a pair $(P, \tilde{P})$. Find the point pair $(P, \tilde{P})$ having the minimal length of shortest connecting path. This shortest path is w_i (blue curve in Fig. 15(c)).

topological sphere with G holes, denoted as the base patch M'. We further decompose M' and all the M_i into pants in the following sections.

4.4.2 Decomposing the Base Patch



(a) (b)

Figure 16: Decomposing the Base Patch. (a) Slice w'_0, get a new pair of pants. Boundary number decreases by 1. (b) Set w'_0 as a new boundary, go on to compute w'_1.

The base patch M' is a topological sphere with $G+B$ holes (G is the genus, B is the number of original boundaries on surface M). As we mentioned previously, when there are less than 3 boundaries (for example, $G < 3, B = 0$), there is no base patch. In those cases, this step is skipped. Fig. 16 illustrates an example of our base patch decomposition, the algorithm is as follows.

Base Patch Decomposition.

```
In:  Base Patch $M'$ and all the waists $w_{i=0...G+B}$.
Out: $G+B-2$ pants patches.
  0. Put all boundaries $w_i$ of $M'$ into a queue $Q$.
  1. If $Q$ has $\leq 3$ boundaries, end; else goto Step 2.
  2. Compute a shortest loop $w'$ homotopic to $w_i \circ w_j$. (Red curves in Fig. 16(a)
     and (b))
  3. $w'$, $w_i$ and $w_j$ bound a pair of pants, remove it from $M'$. Remove $w_i$ and $w_j$
     from $Q$. Put $w'$ to $Q$. Goto Step 1.
```

During each iteration, we decrease the number of boundaries on M' by 1 because two boundaries w_i and w_j are removed, one new boundary w' is created. Therefore, this algorithm will process for $G+B-3$ iterations, and we get $G+B-2$ pants patches ($G+B-3$ from iterations, and base patch becomes the last one).

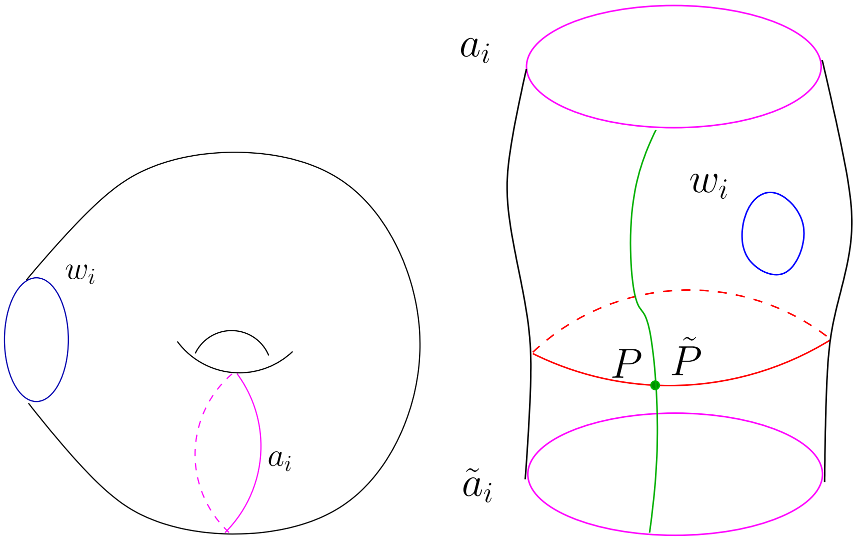
In step 2, we need to trace a shortest loop w' homotopic to $w_i \circ w_j$. The computation follows the idea of the previous algorithm of shrinking waists (Fig. 15(b) and (c)).

Compute Shortest Loop w' Homotopic to $w_i \circ w_j$.

```
In:  Surface $M'$ and two waists $w_i$, $w_j$ on it.
Out: Shortest (under the given metric) loop $w'$ homotopic to $w_i \circ w_j$.
  1. Connect $w_i$ and $w_j$ together by a shortest path $w_{ij}$.
  2. Connect all other loops in $Q$ together with shortest paths without intersecting
     $w_{ij}$. These loops together form a new big boundary $w'_{ij}$
  3. $w_{ij}$ and $w'_{ij}$ bound a cylinder (same as in Fig. 15(c)). Compute the shortest
     cycle $w'$ using the same idea of step 3 and step 4. in Algorithm 4.4.1.
```

4.4.3 Decomposing Handles Patches

After we find each waist w_i in the pipeline's step one, we remove the handle patches M_i from M, each of which is a genus-1 surface with one boundary. We can



(a) (b)

Figure 17: Decomposing a Handle Patch. (a) Slice a_i apart. (b) Slice the green curve that connects two outer boundaries. Then find the shortest path (red) connecting corresponding point pairs on the green curve.

simply find a shortest cycle homotopic to the handle loop a_i and slice it apart to make M_i a pants patch.

The idea is illustrated in Fig. 17. We first slice a_i to get a cylinder with an inner boundary w_i; then we find the shortest path γ (green curve in (b)) connecting two outer boundaries. Then we slice γ, and find the shortest "shortest paths" connecting P and $\tilde{P}$, $P \in \gamma, \tilde{P} \in \tilde{\gamma}$. Now we make M_i a pants patch by slicing this shortest cycle (red cycle in (b)).

4.4.4 Topological Surgery and Evolution

The pipeline introduced in the above sections provides a canonical (therefore consistent) decomposition for surfaces with the same topology. When the given surfaces are with different topologies, our framework can also flexibly handle it. The user only needs to provide some marker points (denoted as *surgery points*).

As shown in Fig. 18(a), when we want to evolve a region to a handle. For example, we want to match the bottom area of the left torus to the bottom handle on 2-torus. We easily pick a pair of points, and punch two holes there. Their

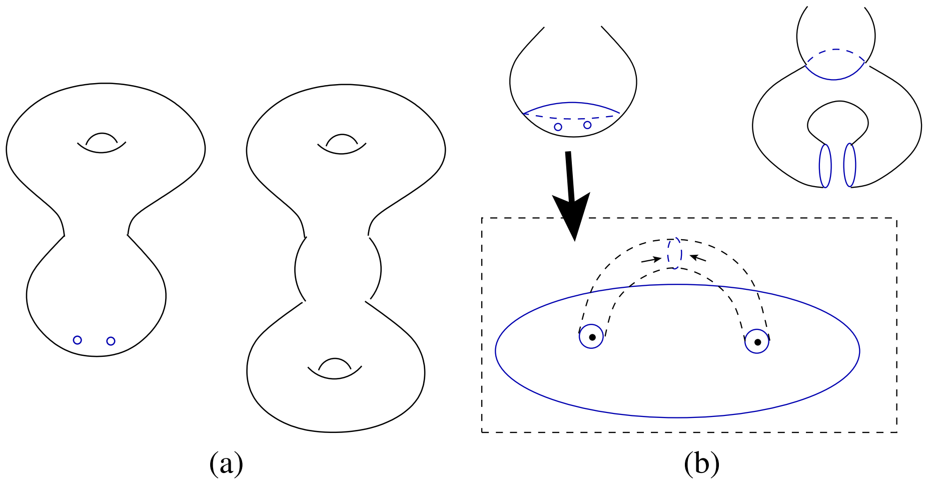


Figure 18: Consistent Decomposition for Surface with Different Topologies. (a) User specifies a pair of markers to correspond to a handle. (b) Each pair of markers generate a new pants patch, which is matched with a handle patch from the second surface.

one-ring neighbors become boundaries c_1 and c_2. Then similar to the previous process introduced in Algorithm 4.4.2, we find a cycle c_3 homotopic to $c_1 \circ c_2$, these three curves bound a pants patch (as shown in (b)), which is matched with the corresponding handle patch on the second surface. Many real examples using surgery points to handle topology change in surface mapping will be presented in the experimental section later.

4.4.5 Matching Feature Points

When the semantics need to be considered during surface mapping, users usually set up corresponding feature points or curves on both surfaces and require them to be matched exactly. To enforce the feature points correspondence as hard constraints, on each feature point, we also punch a hole, and make its 1-ring neighbor a boundary. During the decomposition, we treat these boundaries generated from feature points as usual boundaries. As we will discuss in Section 4.5, since all the boundaries in the pants set are consistently matched, **hard constraints** are guaranteed. We can easily cut feature curves into boundaries, and guarantee their hard constraints in the same way.

4.4.6 User-Guided Segmentation

Our pants decomposition follows a topological consistency and all the segmentation paths are determined by geometry (shortest paths) of the surface. Taking semantics into account is sometimes important in segmentation. We therefore also allow users to easily sketch some curve segments to guide decomposition.

Decomposition should be topologically correct to assure validity and consistency of segmentation. Therefore, our system takes user's sketches as **soft constraints**, and tries to follow the guidance while at the same time guarantee the newly traced cycles are homotopic to the original ones. This can ease user's operations, eliminate the necessity of user's expertise, and greatly improve our system's robustness.

Users can sketch some guiding curves on the mesh for a specific handle or boundary. When we compute the waist of this handle or boundary, we use a special metric m_M to attract the shortest cycle towards the guiding curves. To design the metric, we first compute each vertex's distance from the guiding curves. A scaling function is defined according to such a distance. The smaller this distance is, the smaller the scale is. Intuitively speaking, regions close to guiding curves shrink while the metric of the distant area preserves or even expands. Under such a metric, the shortest paths will be attracted towards the guiding curve segments:

4.4.7 Decomposition Sequence

The correspondence of handle patches between two surfaces is determined by indexing of handle/tunnel loops. Our system generates a default index sequence for handle and tunnel loops on one surface for its canonical decomposition. The user can reorder the indexing to change the handle correspondence when necessary.

By default, when M and all its handle/tunnel loops (homology group bases) are given as input. We simply project the object onto a plane (e.g. $X-Y$ plane); for each basis (a pair of handle and tunnel loops), we compute their "center" on this plane; then we enumerate the index for each basis according to its center's slope on this plane. As we will show later, an arbitrary indexing order determines a homotopy type of decomposition (and mapping). Users can easily change this order through the interface of our system.

Compute the Guiding Metric.

In: Surface mesh M, threshold D, parameter α.

Out: Guiding metric m_M defined on each edge of M.

1. Set an initial metric, $m_M(e) = 1$ for all edges. Set all vertices on guiding curves as starting points.
2. Perform the Dijkstra algorithm, using $m_M(e)$ as the edge length of the graph. We get the hop distance $d(v)$ from all vertices to the guiding curves.
3. Set the weight function:

   $$w(v_i) = \begin{cases} 1, & d(v_i) > D; \\ (\frac{d(v_i)}{D})^\alpha, & o/w. \end{cases} ,$$

   where D is a hop distance threshold, α is a parameter to control the strength of the attraction.
4. Set

   $$m_M(e_{ij}) = (\frac{w(v_i)+w(v_j)}{2}) * |e_{ij}|,$$

   where $|e_{ij}|$ is the original edge length of e_{ij}.

The second issue is the sequence used to decompose the base patch. The default sequence is to decompose waists from small indices to large ones. Users can also provide their decomposition sequence script, if desired. Under a specific base patch decomposition sequence, all the pants adjacency relationship can be deduced easily. For a surface with G topological handles (including virtual handles introduced by surgery points) and B boundaries, we can get $2G-2+B$ pairs of pants, and totally $6G-6+2B$ adjacency relations.

4.4.8 Robust Shortest Path Tracing

We always generate pants by tracing shortest cycles. To assure that the pants do not become degenerate, we should prevent shortest cycles from intersecting boundaries. Therefore, here we slightly update the Dijkstra algorithm so that it prevents the shortest paths from reaching boundaries (or from reaching specified curves).

In the Dijkstra algorithm, when a vertex v is visited, we enqueue it and relax its neighbors ([17], page 595). Now if v is on a boundary (or on some specific curves

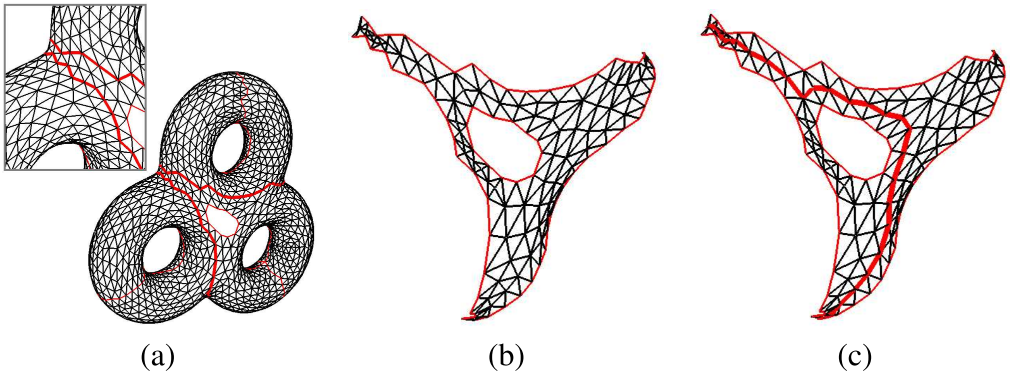


Figure 19: Adaptive Edge-Split. (a) Two waists (thick red curves) are close to each other. No path can cross through the upper left region. (b) Edge-split on base patch before tracing. (c) Shortest cycles pass through successfully.

we want to circumvent), we do not enqueue v nor update its neighbors. Using this updated algorithm, shortest cycles' intersection with boundaries will be prevented.

The original Dijkstra algorithm guarantees to trace a shortest path for an arbitrary vertex on a connected mesh. Our modified algorithm only fails when two boundaries are too close to each other. An example is illustrated in Fig. 19. (a) shows a three-hole torus with a boundary, and its waists w_0 and w_1 (thick red curves) are close to each other, therefore the upper left region is error-prone: there are some edges spanning these two boundaries. So although topologically a path should be able to go through this region between two boundaries without any intersections, a real go-through path will inevitably intersect boundaries under the current mesh connectivity. We call this kind of spanning edges *dangerous edges*. We perform the edge split on all *dangerous edges* before computing shortest cycles/paths, as shown in (b). This then robustly guarantees the success of our path tracing (c).

There is another technical issue here. Usually we are not only satisfied with the correct topology, but also want the cycles to be apart from boundaries so that the pants will not be too skinny. We may also need to handle similar situation around surgery/feature points and boundaries on the base patch, so that degenerate pants will not be created. Therefore, we also apply a modified metric (as discussed above in Section 4.4.6) to push shortest paths away from boundaries or some specific curves.

4.5 Matching Pants Patches

After we perform consistent pants decomposition on two surfaces M and M' respectively, we get two consistent pants sets $M_i, (i = 0 \ldots n)$ and $M'_i, (i = 0 \ldots n)$. In this section, we map each corresponding pairs of pants patches: $f_i(M_i) \to M'_i$.

To assure global continuity, mappings across the pants' boundaries should be consistent. Rigorously speaking, if a curve segment γ is on two adjacent pants M_i and M_j: $\gamma \subset \partial M_i, \gamma \subset \partial M_j$, then we should have $f_i(\gamma) = f_j(\gamma)$. Therefore, as we discussed in Section 4.3, we slice a pair of pants into two patches and compute their boundary-fixed harmonic maps.

As shown in Fig. 13, slicing a pants patch M_i into two hexagons needs 3 curves connecting M_i's boundaries. We simply use the shortest paths to connect each boundaries pair. Then these three paths slice M_i into two patches. The 6 intersection points among these curves and pants' boundaries are mapped to 6 corners of the regular hexagon. To assure the mapping is continuous across the boundary, when corners of M_i have been determined, its adjacent pants' corners on their shared boundary should be consistently determined. The following algorithm computes all corners on a set of pants consistently.

Computing Corners for Pants Set $M_i, i = 0 \ldots n$.

In: A set of pants patches M_i.

Out: All corners on each of M_i.

1. For handle patches $M_0 \ldots M_{G-1}$:
   (1.1) Connect shortest cycles between legs, we get corners P_3, P_4 (the index follows Fig. 13).
   (1.2) Set P_2 as the point on Γ_i^1 farthest from P_3. The farthest point on Γ_i^2 from P_4 is P_5.
   (1.3) Trace the shortest path from P_5 to Γ_i^0; its end point on Γ_i^0 is P_0. The farthest point on Γ_i^0 from P_0 is P_1.
2. Propagate existing corners to adjacent pants: check every M_i's adjacent pants M_j; if corners on M_j's shared boundaries have not been determined, fix them.
3. For each newly propagated M_j, if M_j's corners on Γ_j^0 have not been decided. Use Step (1.3) above to fix them.
4. Stop if all corners have been fixed, otherwise GOTO Step 2.

We first go through all handle patches because their corners are freely determined. Then we propagate their corners to the adjacent pants. Each step of the base patch decomposition combines two waists to generate a new pants patch, so the above propagation will not get stuck, and it ends within several iterations with all corners consistently fixed.

Now each pants patch M_i can be sliced into two patches M_i^0 and M_i^1, as Fig. 13 shows. We compute their harmonic maps to the regular hexagon $\mathcal{H}$, $h_j(M_i^j) \rightarrow \mathcal{H}, (j = 0, 1)$ with the following boundary conditions: set each patch's 6 corners' UV coordinates to the regular hexagon's corners; for other boundary points between each pair of corners, interpolate their UV coordinates using the arc-length ratio. Each harmonic map is computed after solving a sparse linear system [24]. On the pants M_i', the same harmonic maps $h'_j(M'^j_i) \rightarrow \mathcal{H}, (j = 0, 1)$ are computed. Then we can immediately get the final composed patch mapping $f(M_i^j) = h'^{-1}_j \circ h_j$ by barycentric point locations. Mapping on boundaries across neighboring patches and pants is consistent. Because each boundary point's image is determined by the corners and corresponding arc length ratios, and both neighboring regions arrive at the same result.

4.6 Experimental Results

We demonstrate our mapping framework using several examples. The generated surface morphing sequences can be found in our accompanying video.

Automatic Mapping Genus-9 Mechanical Parts. Consistent pants decomposition is automatically performed on three genus-9 mechanical parts. As shown in Fig. 23, although all models in this example have very complicated topology and geometry, once their homotopy group bases are indexed, all the following decomposition is performed in a canonical way without any user involvement. This demonstrates the great automation capability of our framework, and it shows our consistent decomposition is a powerful tool for registering complicated objects. Two morphing sequences generated using our mapping are shown.

Deforming Hands: "Five" to "Okay". In this example, we map a human hand (Fig. 20(a) (left)) to another hand (right). This example shows that how semantics are well handled in our framework with intuitive user intervention.

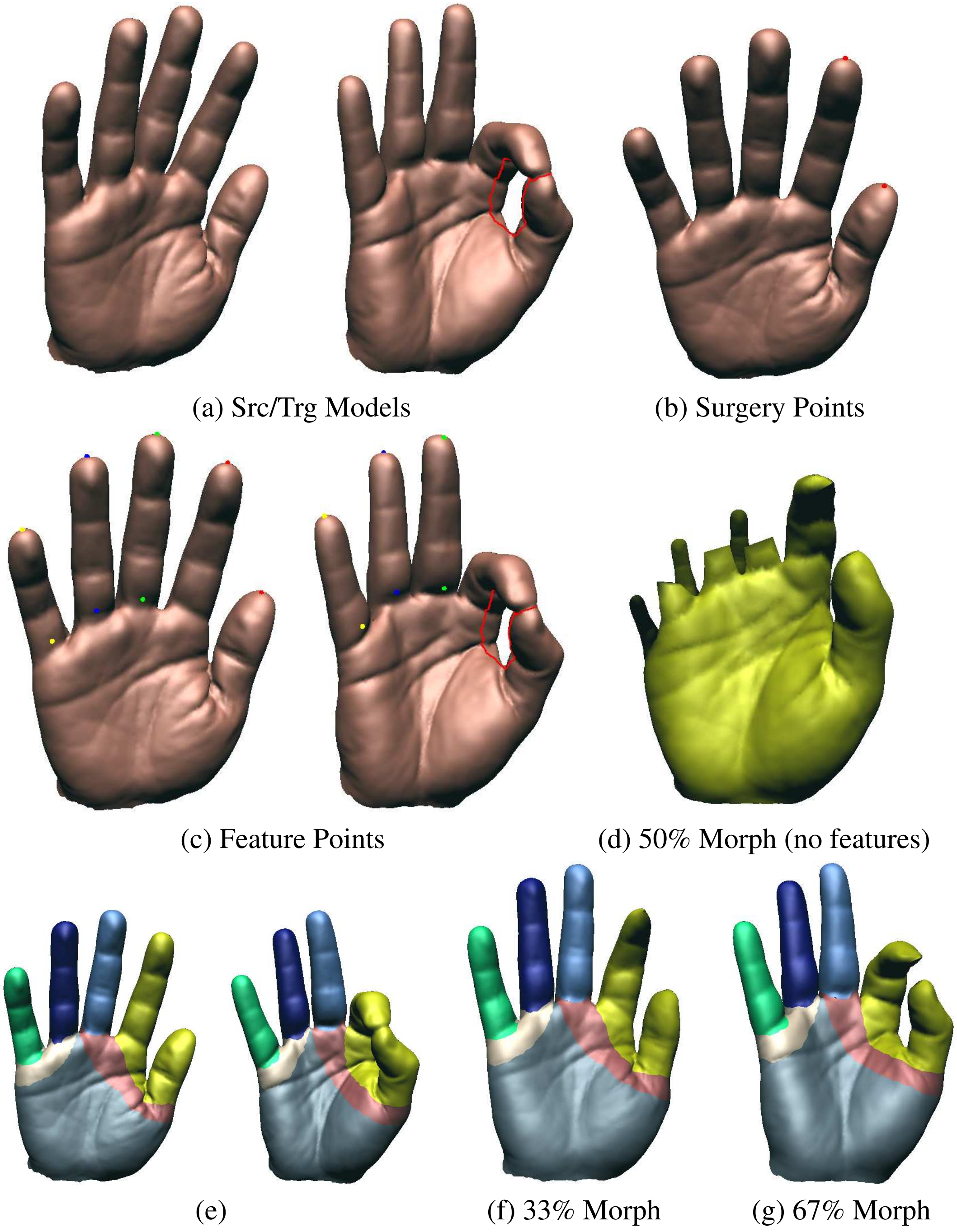

(a) Src/Trg Models
(b) Surgery Points
(c) Feature Points
(d) 50% Morph (no features)
(e)
(f) 33% Morph
(g) 67% Morph

Figure 20: Mapping Hands: "Five" to "Okay". (a) Source and target surfaces. (b) Two surgery points are the least requirements due to the topological difference. (c) Users define more feature points for semantics purpose. (d) Without feature points in (c), 3 fingers are not matched, the morphing is not satisfactory semantically. (e) The refined decomposition results (with feature points). (f) and (g) show the newly generated morphing.

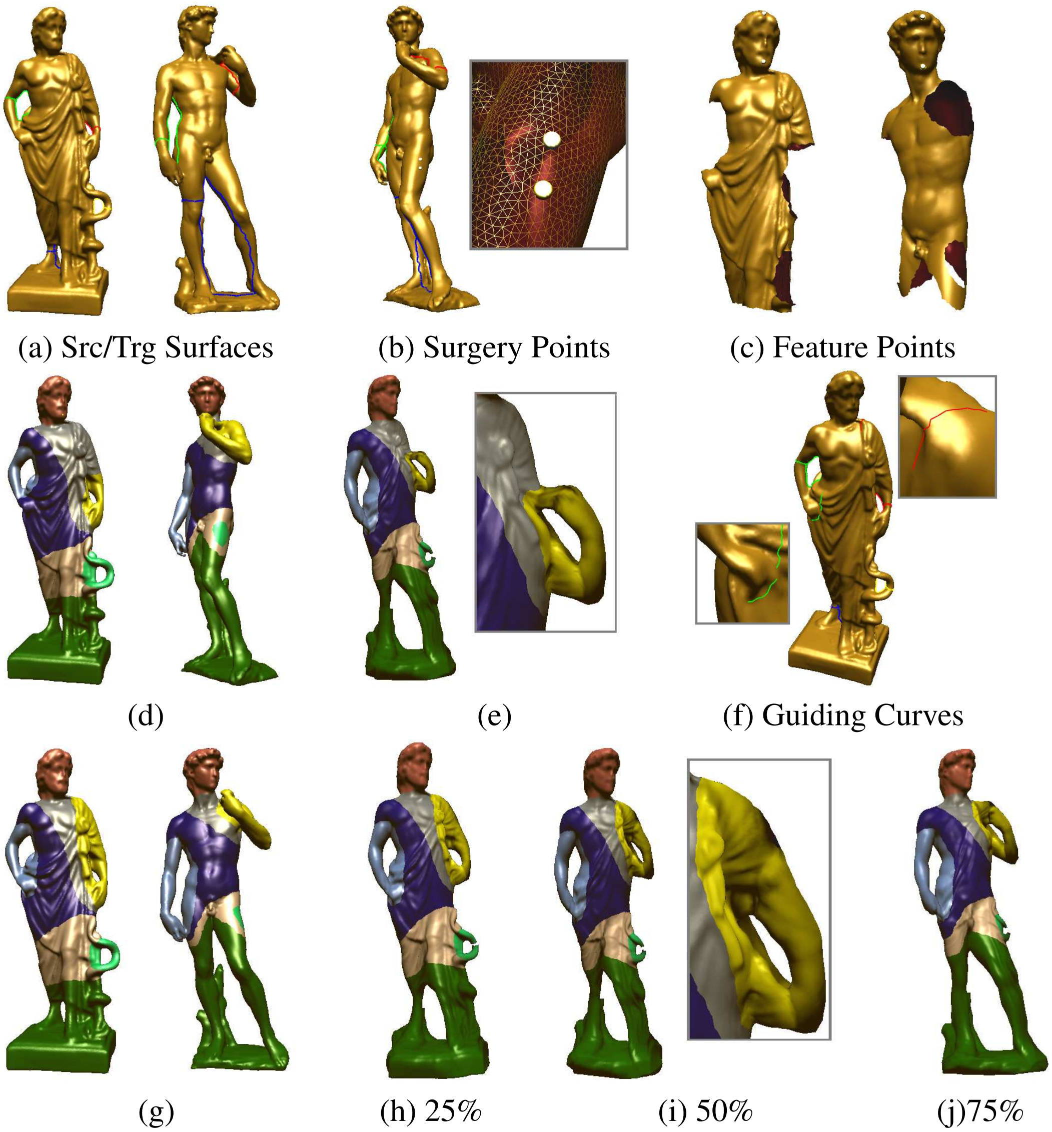


Figure 21: Mapping the Greek Sculpture to David. (a) Two surfaces and their homotopy group bases. (b) Two surgery points (matched with the lower right green handle on the Greek). (c) Base patches of both models, and two feature points to assure correspondence on head regions. (d) The decomposition result without further user involvements. (e) Geometrically optimal decomposition may have poor semantics effect (yellow regions). (f) Users sketch some guiding curves. (g) The new decomposition result with guided segmentation. (h)-(j) A more visually natural morphing sequence.

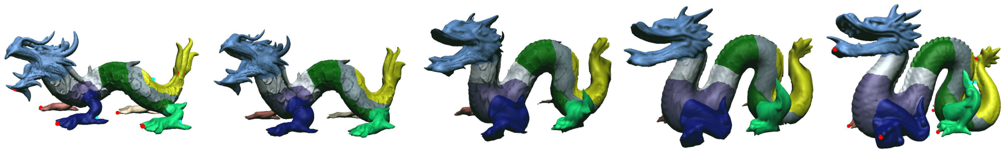
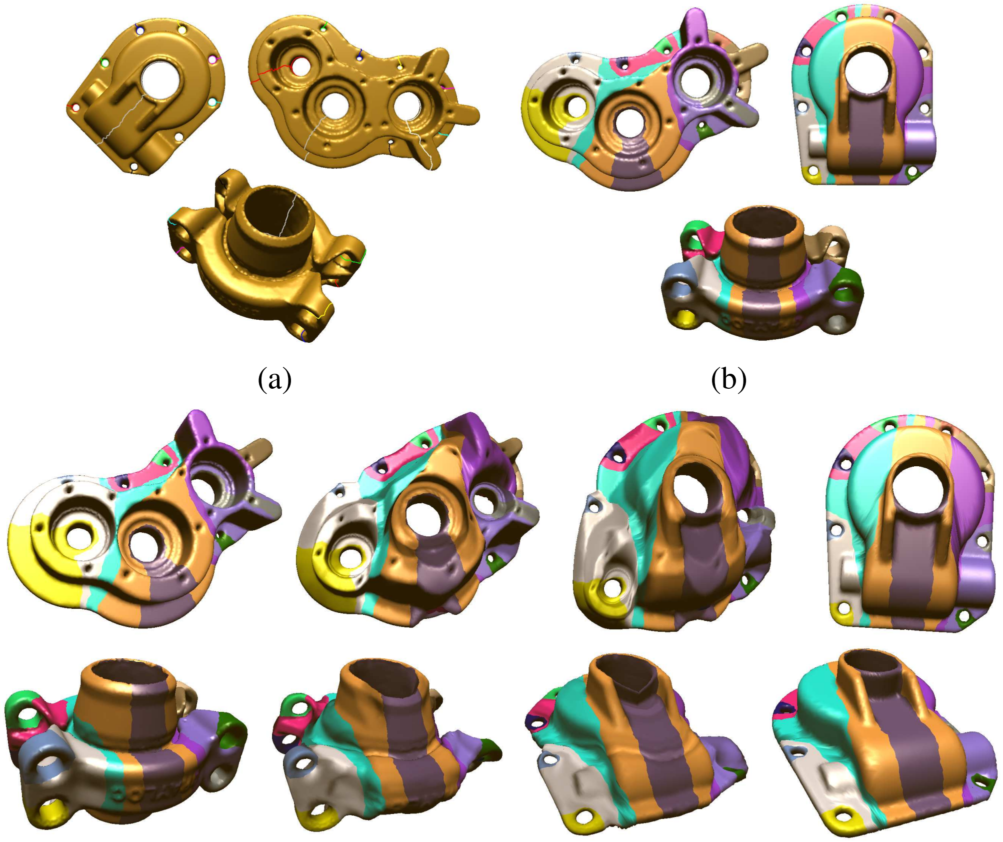


Figure 22: Mapping two Dragons. Feature/surgery points are placed on both dragons (red and green markers on the head and legs). The morphing sequence is generated.



(a)

(b)

Figure 23: Mapping Genus-9 Mechanical Parts. The initial homotopy group bases on each models are color-encoded in (a). (b) illustrates the canonical decomposition result. The next two rows visualize mappings through morphing sequences.

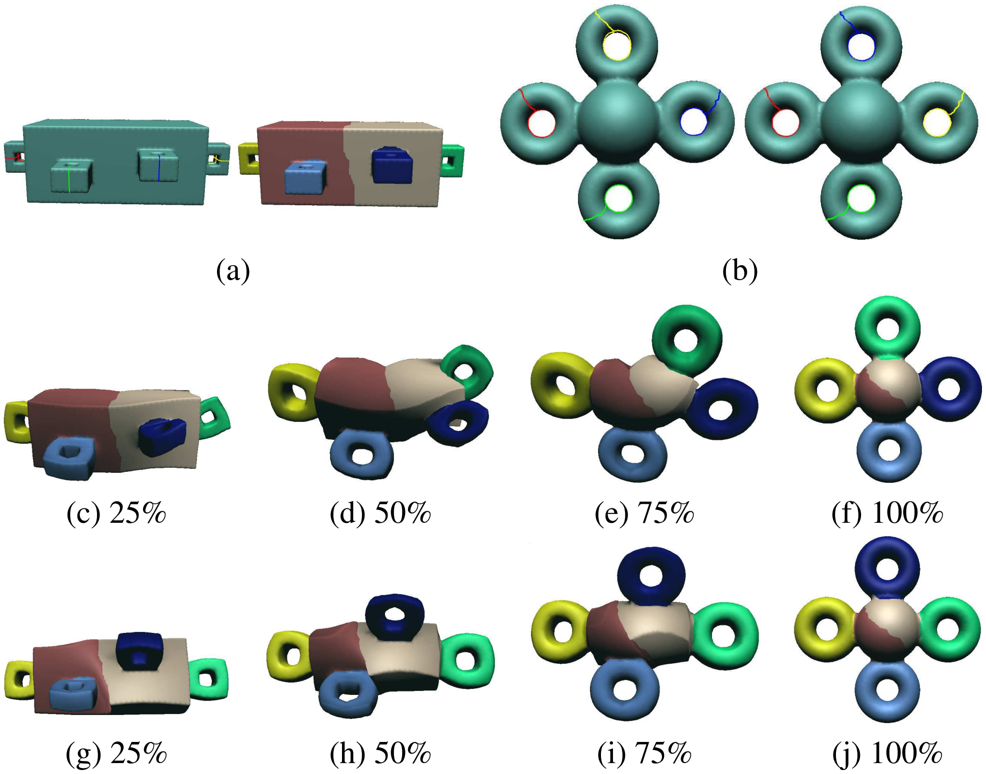
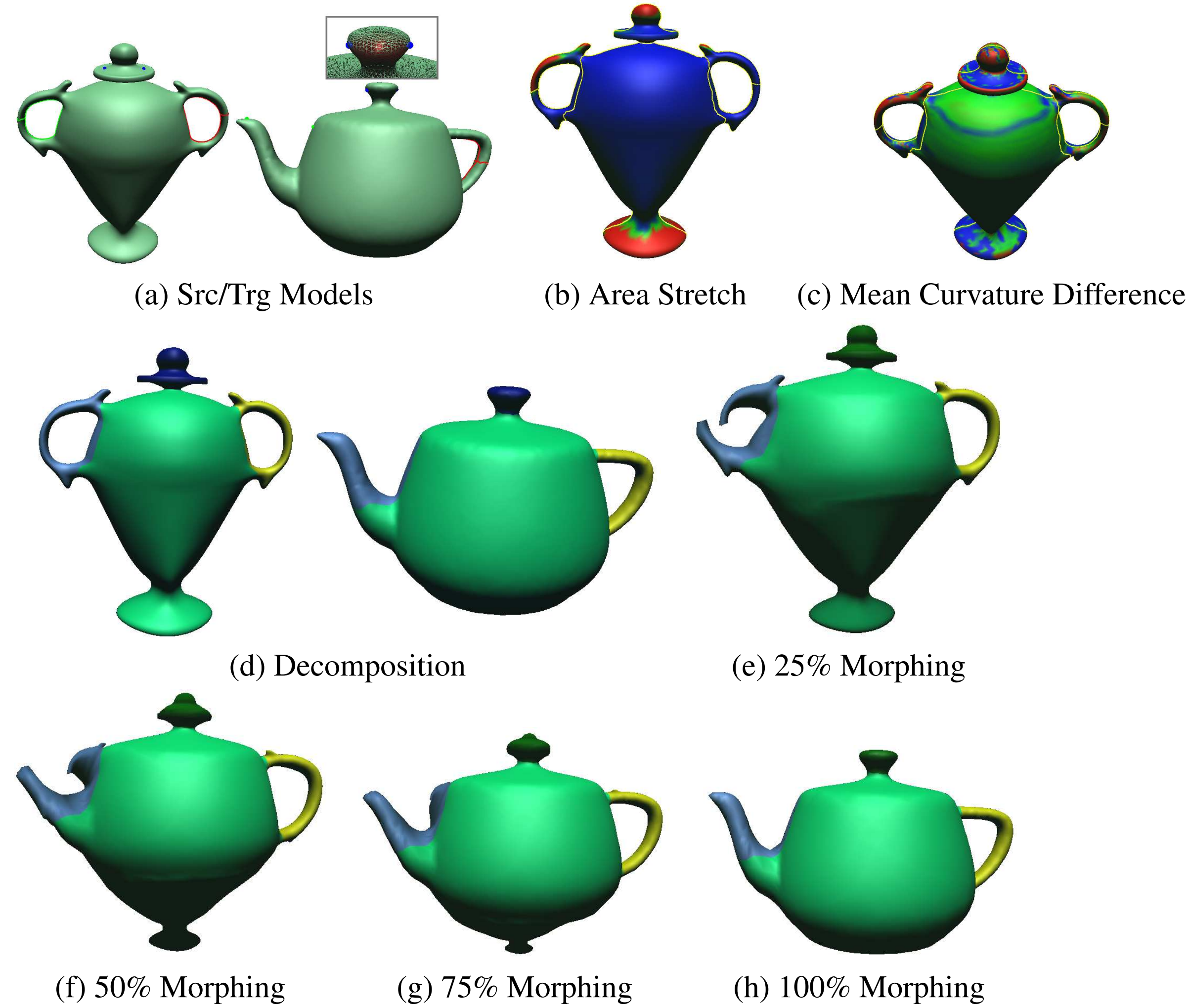


Figure 24: Design Surface Mapping with Arbitrary Homotopy Types. (a) The source surface and its canonical decomposition. (b) User chooses different homotopy types by changing the index of handle/tunnel loops. (c)-(f) First homotopy type: the "green" handle goes up. (g)-(j) Second homotopy type: the "blue" handle goes up.



(a) Src/Trg Models

(b) Area Stretch

(c) Mean Curvature Difference

(d) Decomposition

(e) 25% Morphing

(f) 50% Morphing

(g) 75% Morphing

(h) 100% Morphing

Figure 25: Vase vs. Teapot. (a) Surfaces with handle/tunnel loops and surgery/feature points. The matching's area stretching (b) and mean curvature difference (c) are color-coded. (d) Pants Decomposition Result. (e)-(h) Morphing.

The source hand is an open genus-0 surface, while the target hand is genus-1 with a boundary (red curves are its handle/tunnel loops). To achieve the topological evolution, at least a pair of surgery points is required on the first hand. Naturally, we can set them on tips of the indexing finger and the thumb (Red points in (b)). If we do not provide any other feature points, each hand is decomposed into one pair of pants. The direct mapping between them can be easily computed. However, such a mapping may not follow shape semantics, which can be visualized using a linearly-interpolated morphing sequence generated from this mapping, as shown in (d): three fingers shrink while three new fingers come out from some places elsewhere. This means those fingers are not matched to each other properly. Our framework allows users to set a few corresponding markers in order to match these fingers, as shown in (c). This results in a finer decomposition as shown in (e). The new mapping is computed using this decomposition result, and guarantees the matching between regions of corresponding fingers; therefore, it generates a more natural morphing sequence as illustrated in (f) and (g).

Genus-4 **Greek Model to Genus-**3 **David Model.** In Fig. 21, we map a genus-4 Greek model to a genus-3 David model. The original surfaces and their homotopy group basis correspondence are shown in (a) (each curve's color indicates its index). A pair of surgery points is set on the target model (b), corresponding to the small handle in the lower right part of the Greek sculpture. In (c), when all the handle patches are removed, we get the base patch of both models. Also, we want to semantically guarantee correspondence between head regions, two feature points are placed on each model. We show the canonical decomposition result in (d). From the semantics aspect, we do not like the segmentation around the right hand (blue patch) of the Greek because the shortest cycle goes through his wrist. The segmentation of the left arm (yellow patch) is even worse; it cuts through the elbow. An unnatural morph mapping the forearm of the Greek to the whole arm of David is shown in (e). This can be easily remedied if users sketch two short guiding curve segments on the Greek model as shown in (f). The new decomposition result is shown in (g), where we get a morph with much better visual effects as shown in (h)-(j).

Morphing Dragons. In Fig. 22, we perform decomposition on two dragons. Several surgery points and feature points are used to guide the mapping, as depicted on the source and target models. The morphing sequence is shown to demonstrate the mapping effect.

Surface Mappings with Different Homotopy Types. This example demonstrates the rigourousness and completeness of our mapping framework. As shown in Fig. 24, different homotopy classes can be chosen arbitrarily by users, they only need to switch the indexing of the homotopy group basis (as shown in (b)). The morphing sequences from the source surface (a) to the target surface based on different mappings are illustrated in the next two rows. They are both reasonably good, and it is up to the user to select which one they really want. Our framework provides a rigorous way for users to decide an arbitrary homotopy type of the mapping.

Shape Matching and Error Analysis. With one-to-one correspondence between two matched surfaces, we can measure at each point the shape difference using some geometric properties, and color-code the error distribution, which is potentially useful for shape comparison and visual analysis.

Fig. 25 illustrates our mapping from a genus-2 vase model to a genus-1 teapot model. (a) shows the models and their handle/tunnel loops; and user-provided surgery/feature points are also depicted. The decomposition results are shown in (d). (e)-(h) show the morphing sequence generated by our mapping.

In (b) and (c), we color-code the shape matching error distribution. We use two terms, one is the area stretching ratio while the other is mean curvature difference. In (b), we compute total area of one-ring triangles around each point on the vase model; when the vase is mapped onto the teapot, we also compute each point's corresponding one-ring area. The ratio of these two areas represents the *stretching* of the mapping, and it is color-coded on the surface: red represents the maximum while blue represents the minimum. From this figure, we can see that the cap, the left handle, the tips of handles, and the bottom of the vase have larger stretching values, because these regions shrink to a relatively small area on the teapot model. In (c), we color-code the mean curvature difference on every point: the regions with large curvature difference (for example, left handle, the rim of the cap) are red. Integration of these two terms over the whole surface has been proved ([41]) that it provides an intrinsic energy for measuring the shape difference. Therefore, our surface mapping/registration can be used for automatic shape comparison and shape analysis. Before applying our mapping for registering models in a database,

Model	Topology	# of Vertices	Run Time	# of Pants Patches
Hand-1	$G = 0, B = 1$	19832	24.5s	7
Hand-2	$G = 1, B = 1$	21055	26.1s	7
Teapot	$G = 1, B = 0$	22012	27.0s	4
Vase	$G = 2, B = 0$	10014	10.1s	4
4-Torus	$G = 4, B = 0$	7994	6.3s	6
David	$G = 3, B = 0$	26138	140.3s	8
Greek	$G = 4, B = 0$	43177	380.5s	8
Asian Dragon	$G = 0, B = 0$	26562	110.1s	10
Casting	$G = 9, B = 0$	34116	423.4s	16

Table 1: Canonical Decomposition Performance for Consistent Surface Segmentation.

each model should go through a preprocessing procedure. For each model, first, its handle and tunnel loops are automatically computed and indexed; second, if necessary for any semantic reason, user can simply reorder indexing of these loops. After this preprocessing, all the follow-up shape comparison and retrieval can be performed automatically.

The complexity of our algorithm is very low, and the performance of our algorithm on most real examples presented here is given in table 1.

4.7 Chapter Summary

We have developed a consistent pants decomposition framework for matching surfaces with arbitrary topology. The consistent generation of sets of pants is a key component to ensure the subsequent high quality surface mapping. Our novel mapping framework has been demonstrated to be efficient, robust, and powerful on examples with arbitrary types of surfaces. Also, the mapping framework simultaneously provides great automation and accommodates intuitive user control. Therefore, our new modeling framework can serve as an enabling tool for many visual computing tasks.

Besides surface mapping, we believe our pants decomposition framework has many other potential applications. First, pants decomposition provides a piecewise representation for any given surface. When we have the semantically meaningful patch segmented from the original surface, we can easily perform the “cut-and-paste” operation from a “part” database ([31]) to produce more meaningful shapes from examples. Since all our segmented patches are pants-like shapes, we could streamline many modeling and simulation tasks. Also, pants decomposition can be extended to a consistent segmentation of volumetric data. Compared with directly computing harmonic maps between volumetric shapes with complicated topology and geometry [82], this decomposition should make the process more robust and general, and it will also provide more semantics control.

Chapter 5

Globally Optimal Surface Mapping

In Chapter 3 and Chapter 4, we presented two works in piecewise surface mapping, which improve the efficiency and effectiveness of the piecewise mapping computation. As we will address in Section 5.1, piecewise mapping has its advantages; however, in many applications, globally continuous and optimized mapping is more desirable. This chapter reveals the key challenges that prevents most existing techniques from conducting the global optimization directly, and introduces our algorithm that computes the globally optimal surface mapping.

5.1 Introduction

Piecewise Mapping has the following advantages:

1. Segmenting surfaces makes the surface mapping computation simpler and more efficient. After segmentation, subregions matching with simple topology can be easily and efficiently mapped through planar domain.
2. Its behavior can be easily controlled, users can control the mapping behavior, for example, homotopy type, or subregion matching can be easily achieved.
3. Feature alignment can be conveniently implemented. By segmenting surfaces while preserving the feature points or curves on the boundary, the feature alignment can be guaranteed in an easier way.

However, piecewise mapping also has its limitations:

1. The mapping result is heavily determined by the initial segmentation. Intuitively speaking, regions that are close to each other may be mapped separately only because they are partitioned into different sub-patches.
2. The smoothness along the cutting boundaries is usually not guaranteed. C^0 continuity can usually be achieved, but higher order smoothness are not necessary near these regions.
3. Global distortion can not be measured and controlled. Unless we consider the mapping entirely, we cannot design a numerical way to measure the global distortion of the map.

In many applications such as scientific computation/simulation, surface comparison, spline construction, remeshing and so on, greater global smoothness and lower global distortion are usually more desirable. Therefore we need to consider the globally smooth surface mapping through some global approaches.

We start from the definition of the distortion of surface mappings. Numerous applications such as shape registration, matching and comparison, shape morphing, and texture/attribute/motion transfer all benefit from such a bijective correspondence between two given surfaces. Researchers usually measure the mapping quality using angular or area distortions, because such criteria dictate the end effect of the enabling applications (e.g., texture mapping). Given two surfaces with different geometry, distortions are usually inevitable; we naturally want to seek the mapping that can minimize distortions as much as possible. However, if two given surfaces are not isometric to each other, there does not exist a mapping that can eliminate the angle and area distortion simultaneously.

In this work, we choose the harmonicity (measuring angular distortion) as the criterion because it is most physically meaningful. If we assume surfaces are made of elastic materials; then when surfaces deform and are mapped to others, the stretching energy caused by the elastic distortion can be formulated as harmonic energy. Among all possible mappings, a harmonic map minimizes the stretching energy and has a direct physical meaning. Also, the harmonic map minimizes angular distortions. For example, conformal mappings are harmonic, and are free of angular distortion.

Besides the physical intuition, harmonicity and conformal mapping have other merits that are critical for real-world applications. First, the dimension of conformal mappings between two given surfaces is finite; by fixing the images of finite points, the mapping can be uniquely determined. Therefore, these kinds of mapping are easy to control. Second, conformal mappings can transform arbitrary surfaces to several canonical domains, and convert all geometric processing into these domains. This greatly simplifies the complexity of these geometric processing algorithms. Third, the theoretic foundation and algorithms of conformal mappings are relatively mature.

In contrast, area-preserving mappings lack physical meaning, lack analytic methodology, as well as practical computation algorithms, and are hard to control. Therefore, in our current work, we use harmonic maps and try to minimize the angular distortion.

The criteria of mapping quality on angular distortions have been analyzed and optimized in the aforementioned surface parameterization research area. Surface parameterization aims to find a bijective map between surfaces and planes (or other simple canonical domains such as spheres), thus it can be treated as a special case of surface mapping since its target surface is usually just a plane or a sphere. Parameterization arises from texture mapping and synthesis, where the angular distortion is the most critical factor when quantifying the mapping quality.

Despite its earlier connection with surface parameterization over canonical domains, finding a minimally-distorted surface mapping between two general surfaces is much more technically challenging. There are three key reasons as follows.

First, there are topological differences. Surface parameterization usually "flattens" a surface onto the plane, specifically, the surface is sliced apart into a topological disk, and the parameterization refers to a map from that disk to the plane, so it has only one topological type. For mapping between general surfaces especially with high genus, which handle of one surface is mapped to which handle of the second surface needs to be considered. This topological factor has to be explicitly determined and it gives rise to the mapping complexity for shapes with nontrivial topology. Rigorously speaking, mappings between two given surfaces are classified into infinite *homotopy* classes [12]. Two maps are isotopic to each other, i.e., belonging to a same homotopy class, only if one can deform to another smoothly. A rigorous surface mapping framework should be able to handle an arbitrarily given homotopy type. On the other hand, only topologically equivalent mappings can be compared together; mappings from different classes should be considered separately since a best mapping may exist in each class. In the following discussion, we shall consider maps that are within the same homotopy class.

Second, due to the topological inequivalence between closed surfaces and the plane, as we mentioned above, parameterizing surfaces onto planer domains cuts the surface along a boundary. The simplified target canonical shape not only leads to some well-established numerical solving techniques, but also unavoidably pushes distortions towards its cutting boundary or a collection of some singularity points. In contrast, mappings between surfaces with same topology should prohibit the cutting, and find a "seamless" result. Therefore, we are not pursuing a map from a topological disk to the plane, but a continuous map between two surfaces with complicated topology.

Third, the most important reason for the lack of globally optimized surface mapping techniques is the complex geometry of the general target surfaces. The non-smoothness of the target shape actually leads to a technical obstacle in finding the global optimum among all possible mappings. A natural way is to follow ideas in surface parameterization: we can optimize the map between surfaces by simply constructing an initial map, and then locally adjusting it using a variational procedure until the distortion energy is reduced to the minimum. We can call this technique "the naive method". When the target surfaces are genus zero (e.g., parameterization onto the sphere or plane), this approach can reach a globally optimized result. However, for mapping surfaces with non-trivial topology, due to the nonexistence of a canonical target domain (see Section 5.2.3), any local optimization process will inevitably get stuck at a local minimum. This is the primary reason that other state-of-the-art methods use base meshes or hierarchical structure to circumvent this problem, while giving up searching for the global optimum.

Our Novel Solution

In this work, we introduce a novel computational framework to tackle the aforementioned challenging problems. Our method, based on the theories of Riemannian uniformization and harmonic maps, is both theoretically rigorous and practically efficient.

Considering two general surfaces with nontrivial topology, under their induced Euclidean metric, the target shape may have complicated geometry, and the harmonic maps are usually not globally unique. Some harmonic maps are local minima of the stretching energy.

To globally reduce the stretching distortion without getting stuck locally, we propose to use the so-called *uniformization metric*. Under their uniformization metric, surfaces with nontrivial topology have constant non-positive Gaussian curvature everywhere, so that the harmonic map becomes **globally unique** [105]. (Please refer to Section 5.2.1 and the Appendix for more theoretic details and Section 5.6.4 for experimental demonstrations). Uniformization theory states that for all surfaces, such a uniformization metric does exist; and we can compute this metric efficiently by using existing techniques.

Under surfaces' uniformization metric, we conduct our optimization process. It is guaranteed to converge to unique global harmonicity under surfaces' uniformization metric. Specifically, our algorithm has the following important merits.

- **Optimality.** Harmonicity under uniformization metric can be globally optimized without worrying about any local optima. The resultant map minimizes the stretching energy and distortion.
- **Uniqueness.** The global harmonic map in hyperbolic space is unique; our algorithm converges to the same result starting from arbitrarily different initial mappings, as long as they belong to the same homotopy class.
- **Conformality.** For genus-zero surfaces, an arbitrary harmonic map is free of angle distortions. In the genus-one case, our optimized map minimizes the angle distortion among all possible maps. Between two general surfaces, if there exists an angle-distortion-free map between them, our method is guaranteed to find such a conformal map.
- **Efficiency.** Harmonicity relaxation under the uniformization metric is performed in 2D, which is much more efficient and robust compared with any other iterative methods directly conducted over curved surfaces.

The main contributions of this work are:

1. We propose and articulate a novel approach to computing a globally optimal map that minimizes distortions between two surfaces with the same non-trivial topology. This process is fully automatic and requires no user interaction. To the best of our knowledge, this is the first attempt to compute a surface mapping with globally minimized energy for arbitrary high genus ($g \geq 1$) models.
2. Using the intrinsic geometric structure of surfaces, the convergence of our algorithm is guaranteed. We quantitatively evaluate its performance, and then design toolkits to clearly visualize the mappings, as well as analyze their converging effects.
3. We use our surface mapping as a powerful tool for data and texture transfer, shape morphing, cross-surface parameterization onto canonical shape domains, shape matching, and shape comparison. Our globally optimized mapping demonstrates its efficacy in these graphics and visualization applications, with potentials in a broader scope of applications.

The remainder of this chapter is organized as follows. In Section 5.2, we introduce the theory and algorithm of our method. Our algorithm proceeds in three main steps, as discussed in Section 5.3, Section 5.4 and Section 5.5, respectively. We then discuss our mapping performance and properties in Section 5.6. Finally, we conclude this chapter in Section 5.8. In the appendix of this thesis, we prove the existence, global uniqueness, and the one-to-one property of the harmonic map and we also show our algorithm will converge to such an optimized map uniquely.

5.2 Theory and Algorithm

5.2.1 Uniformization Metric

On a surface, a *metric*, or *Riemannian metric* is a tensor that defines the inner product on the tangent plane at each point. With the metric, the length of a tangent vector can be determined, and the angle between two tangent vectors can be explicitly computed.

Suppose S is a smooth surface embedded in $\mathbb{R}^3$; it has the induced Euclidean metric $\mathbf{g}$. We denote the surface S together with its equipped metric $\mathbf{g}$ as $(S, \mathbf{g})$.

If $\lambda : S \rightarrow R$ is a scalar function defined on the surface, then $\bar{\mathbf{g}} = e^{2\lambda}\mathbf{g}$ is another metric on S. Any angles on the surface measured by $\mathbf{g}$ equal to those measured by $\bar{\mathbf{g}}$, therefore, we say $\bar{\mathbf{g}}$ is conformal to $\mathbf{g}$, meaning that changing between these two metrics is angle-preserving.

Given two surfaces S_1 and S_2, the uniqueness of the harmonic map from S_1 to S_2, as we will discuss in the upcoming section, is determined by the distribution of the Gaussian curvature K of S_2. It is important to note that K is fully determined by the equipped metric of the surface. The relation between the curvatures K and $\bar{K}$ under $\mathbf{g}$ and $\bar{\mathbf{g}}$ is $\bar{K} = e^{2\lambda}(-\Delta\lambda + K)$.

Riemann uniformization states that for an arbitrary closed surface, there exists a unique λ such that $e^{2\lambda}\mathbf{g}$ induces constant Gaussian curvature. Furthermore, the constant is one of the three choices $\{+1, 0, -1\}$ for surfaces with zero, one, and higher genus, respectively. Such a metric, $e^{2\lambda}\mathbf{g}$, is called the *uniformization metric* of the surface. The uniformization metric can be computed using the Ricci flow method (see Section 5.4).

5.2.2 Euclidean Harmonic Map and Conformal Map

Given two surfaces embedded in $\mathbb{R}^3$ with the induced Euclidean metrics $(S_1, \mathbf{g}_1)$ and $(S_2, \mathbf{g}_2)$, $f : S_1 \rightarrow S_2$ is a map between them, the harmonic energy (stretching energy) is defined as

$$E(f) = \int_{S_1} |\nabla f|^2 dA_1, \tag{4}$$

where ∇f is the gradient of the map. A harmonic map is a *critical point* of the harmonic energy. Harmonic maps depend on the Riemannian metrics. However, if $f : (S_1, \mathbf{g}_1) \rightarrow (S_2, \mathbf{g}_2)$ is a harmonic map, then $f : (S_1, e^{2\lambda}\mathbf{g}_1) \rightarrow (S_2, \mathbf{g}_2)$ is also a harmonic map.

If a map preserves angles, then the map is called a conformal map. Analytically, if the *pull back* metric $f^*\mathbf{g}_2$ on S_1 is conformal to $\mathbf{g}_1$, $e^{2\lambda}\mathbf{g}_1 = f^*\mathbf{g}_2$, then f is conformal. A conformal map must be harmonic. For closed genus-zero surfaces, harmonic maps are also conformal. In the general case, if S_1 and S_2 are with complicated topology, then there may not exist a conformal map. But there is a special map, which minimizes the maximum of the angle distortion; such a map is called the extremal quasi-conformal map. For the genus-one case, such an *extremal quasi-conformal map* is the harmonic map under uniformization metric. Therefore, if the given surfaces are genus-one, our algorithm converges to the extremal quasi-conformal map.

5.2.3 Uniqueness of Harmonic Maps

The uniqueness of harmonic maps between surfaces is determined by the shape of the target objects. For genus-zero surfaces, there are infinite harmonic (conformal) maps, all with zero angular-distortion. Each two of these maps differ by a möbius transformation on the sphere domain.

Harmonic maps between surfaces with non-trivial topology are also not unique if the Gaussian curvature of the target surface is positive somewhere. However, if the target surface has non-positive Gaussian curvature everywhere, then the harmonic map exists and is **unique**. For example, if the Euler number $\chi(S_i) < 0, i = 1,2$, and we apply uniformization metric $\mathbf{g}_2$ on S_2, then harmonic map f exists and is unique, with its energy $E(f)$ reaching the global minimum.

Therefore, between two arbitrary surfaces with genus ≥ 2, there uniquely exists such a stretching-minimized harmonic map. For genus-one surfaces, $\chi = 0$, under uniformization metric, the harmonic maps are not unique, but only differ by rigid translations on the $\mathbb{R}^2$ universal covering space, and we can use one feature point to uniquely determine it.

5.2.4 Poincaré Disk Model and its Harmonic Maps

If the given surfaces are with higher genus, their uniformization metrics can only be embedded in hyperbolic space. We have to carry out our computation in this space, which can be modeled by the Poincaré disk as follows.

The Poincaré disk is the unit disk on the complex plane $z\bar{z} \leq 1$, with the Riemannian metric $ds^2 = \frac{4dzd\bar{z}}{(1-z\bar{z})^2}$. Our goal is to compute a map $f : (S_1, \mathbf{g}_1) \rightarrow (S_2, \mathbf{g}_2)$. We use their uniformization metrics and compute a harmonic map $\bar{f} : (S_1, \bar{\mathbf{g}}_1) \rightarrow (S_2, \bar{\mathbf{g}}_2)$. The computational algorithm of hyperbolic harmonic maps is based on theoretic results in [104].

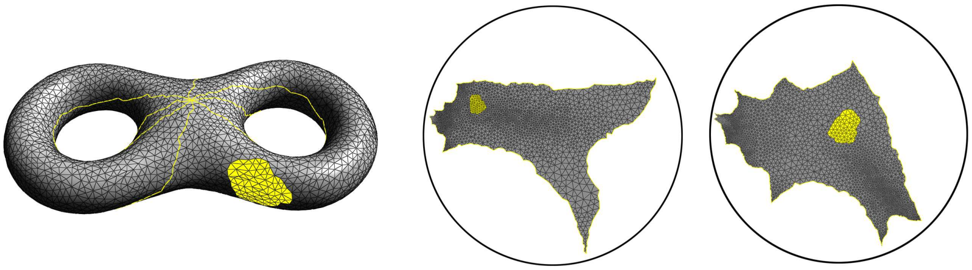


Figure 26: (Left) The yellow patch represents a chart on the two-hole torus model; (Middle) Embed the two-hole torus model in the Poincaré disk; (Right) A möbius transformation moves the chart to the center of the Poincaré disk.

We denote the parameters of S_1 on the Poincaré disk as (x,y), the parameter of S_2 as (u,v), then the map $\bar{f}$ is represented as $\bar{f}(x,y) = (u(x,y), v(x,y))$. The harmonic energy is

$$E(\bar{f}) = \int_{S_1} 4\frac{|\nabla u|^2 + |\nabla v|^2}{(1-u^2-v^2)^2} dxdy, \tag{5}$$

where ∇u is $(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y})$ and ∇v is $(\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y})$.

The harmonic energy in hyperbolic space (5) has a more complicated form compared to harmonic energy in Euclidean space (4). We simplify the problem using the following two merits of hyperbolic harmonic energy:

1. In a small neighborhood of the origin $u^2 + v^2 < \varepsilon$, since $(1-u^2-v^2)^{-2} \to 1$, the hyperbolic metric is close to the Euclidean metric, the hyperbolic harmonic energy is close to the Euclidean harmonic energy. We can optimize the hyperbolic energy by minimizing the Euclidean energy.
2. If ϕ is a Möbius transformation of the Poincaré disk, then the composition $\phi \circ \bar{f}$ and $\bar{f}$ have the same hyperbolic harmonic energy. This is because the Möbius transformation is the rigid motion in the hyperbolic space, and the harmonic energy is invariant under such isometries of the target surface.

We describe our computational methodology for hyperbolic harmonic maps as follows.

1. The surfaces are tessellated into many small triangular patches, $S_1 = \bigcup_i T_i$, where T_i is a triangular patch, then the harmonic energy is decomposed into

the summation of the energy of the map restricted on these patches, the sub-maps, $E(\bar{f}) = \sum_i E(\bar{f}_i), \bar{f}_i : T_i \to \mathbb{H}^2$.

2. Each sub-map $\bar{f}_i$ is composed with a Möbius transformation ϕ_i, such that the image $\phi_i \circ \bar{f}_i(T_i)$ is in the neighborhood of the origin.
3. If the tessellation is refined enough, T_i is small, and the diameter of its image under the corresponding sub-map is within an ε-threshold, the hyperbolic energy can be approximated by Euclidean harmonic energy with high accuracy.

Therefore, computing the harmonic map under the hyperbolic metric, which is equivalent to minimizing the hyperbolic harmonic energy, is now converted to optimizing a collection of Euclidean harmonic energies of sub-maps. We can use the *mean value property* of the harmonic function to minimize the Euclidean harmonic energy.

5.2.5 Discrete Algorithm

We summarize our approach as the following discrete algorithm:

Global Optimal Surface Mapping.

In: Source surface S_1, target surface S_2.

Out: Harmonic map f under the uniformization metric of S_2.

1. Construct an initial map $\tilde{f} : S_1 \to S_2$ (See Section 5.3).
2. Compute the conformal deformation (uniformization metric) of S_2 using the technique introduced in [57], then embed S_2 in the canonical domain $\mathbb{C}$ or $\mathbb{H}^2$, $\phi_2 : S_2 \to \mathbb{C}$ or $\phi_2 : S_2 \to \mathbb{H}^2$. (See Section 5.4)
3. Compose the $\tilde{f}$ and ϕ_2 to get $\omega : S_1 \to \mathbb{C}$ or $\omega : S_1 \to \mathbb{H}^2$, and apply heat diffusion on dynamic charts to minimize the harmonic energy:

$$\frac{d\omega}{dt} = -\Delta\omega. \tag{6}$$

(See Section 5.5)

4. When ω converges to the global minimum, let $f_{1H} = \omega$ and get the final map $f = \phi_2^{-1} \circ f_{1H}$.

5.3 Initial Mapping

We first build up an initial mapping between the given surfaces. The initial mappings determine the homotopy class of the resultant surface mapping (which will be discussed later in Section 5.6.2). Our pipeline for creating the initial map has two steps: (1) we unfold both surfaces to disks through a cut graph called the *system of loops* (See Section 5.3.1); (2) we map two surfaces via the disk domain (See Section 5.3.2).

5.3.1 Cutting a Surface into a Topological Disk

An orientable closed surface of genus $g, (g \geq 1)$ can be cut into a single topological disk by removing a so-called *cut graph*. Computing a special case of cut graphs which passes through a common given base point, called *systems of loops*, is studied in computational geometry. One of the state-of-the-art techniques, [25], uses an efficient greedy algorithm to get an optimal cutting loop. We refine their algorithm for our surface cutting.

We first briefly describe their algorithm for computing a system of loops L on the given mesh S and the base point x:

1. Compute the shortest paths tree T of S from x.
2. For each edge $e \in S \backslash T$ (i.e. $e \notin T$), compute the shortest loop that contains e, denoted as $\sigma(e)$, which consists of two shortest paths from x to the endpoints of e plus the e itself.
3. Compute the dual graph of $S \backslash T$, denoted as $(S \backslash T)^*$. Compute its maximum spanning tree T^*, where the weight of each dual edge e^* is $\sigma(e)$.
4. Get the set E' which contains every edge that is neither in T nor crossed by T^*.
5. E' has $2g$ edges $e_1, e_2, \cdots, e_{2g}$. Compute shortest loop $\sigma'(e_i)$ containing each e_i. These loops constitute the system loop L.

In [25], they assumed that all the shortest paths from each point on the cut path to the base point only intersect at the common base point. This assumption holds in the smooth case but often fails for triangular mesh representations. Thus in step (5), shortest paths on triangle meshes may intersect each other, especially

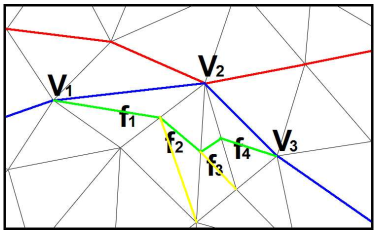


Figure 27: Local Refinement on the System of Loops Computation. When the blue cut path intersect with the existing red path in V_2, we apply a local refinement. The intersected path $[V_1, V_2, V_3]$ segment is replaced by the new green path. Yellow segments are new edges inserted during edge splits in the refinement.

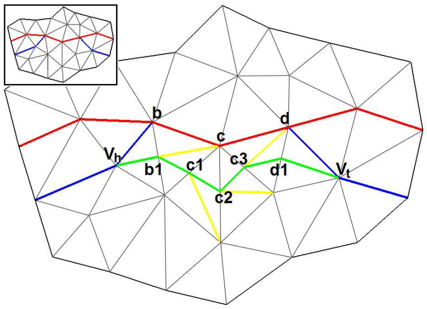


Figure 28: Refinement on the System of Loops Computation.

for high genus surfaces. For example, on a genus-six surface, 12 loops will go through the base point, meaning that the valence of the base point should at least be 24 to prevent the paths' intersections outside the base point. Such high density connectivity is hardly satisfied in ordinary mesh data. Therefore, a robust algorithm has to adaptively change the connectivity.

As shown in Figure 27, locally, if a cut path (blue) intersects with an existing path (red) in one point. We apply the following algorithm on the blue curve to make it bypass the red one:

After applying this algorithm, we replace the intersected path $[V_1, V_2, V_3]$ segment by the new path (as shown in green). The yellow segments are edges newly inserted onto the mesh during the edge split procedure.

In general, if the intersected parts have more than one point, we apply the adaptive bypassing algorithm iteratively on each intersected vertex. Figure 28 illustrates

Adaptive Bypassing Algorithm:

```
1. Spin around the intersected vertex $V_2$, enqueue all faces between $[V_1, V_2]$ and
   $[V_2, V_3]$(For example, $f_1, f_2, f_3, f_4$ here).
2. Set $V_1$ as the current point $p$.
3. Pop face $f$ out from the head of queue. If $[V_2, V_3]$ is in $f$, add edge $[p, V_3]$ to
   the new path and STOP; else GOTO 4.
4. Split the edge opposite to the current vertex $p$. The new split point is denoted
   as $q$. Add the edge $[p, q]$ into the new path, move to $q$: set $p := q$. GOTO 3.
```

this. In the small picture (upper left), a cut path (blue) passes through an existed cut path (red). We apply the following algorithm on the intersected segments:

Refinement Algorithm:

```
1. Find out the point right before the intersection ($V_h$) and the first point right
   after this intersection ($V_t$). Push all vertices on the current path between $V_h$
   and $V_t$ into a queue $Q$.
2. Pop each vertex in $Q$, and apply adaptive bypassing algorithm on it.
```

Intersections usually happen near the base point because cutting paths are dense in this region. The V_t in such case is the base point, and the same refinement process is applied.

5.3.2 Initial Mapping via $4g$-gon

With the system of loops, we slice each surface onto a topological disk. For a genus g surface, the cut graph passes through the base point $4g$ times, making the disk a topological $4g$-gon. We map two given surfaces via this $4g$-gon, as the procedure illustrated in the Figure 29. The following is the algorithm.

By the above algorithm, we get an initial mapping from S_1 to S_2. This initial mapping is only used to determine the homotopy type. In the following sections, we will prove and demonstrate that if two initial cuts induce two maps belonging to the same homotopy class, then the final results are identical.

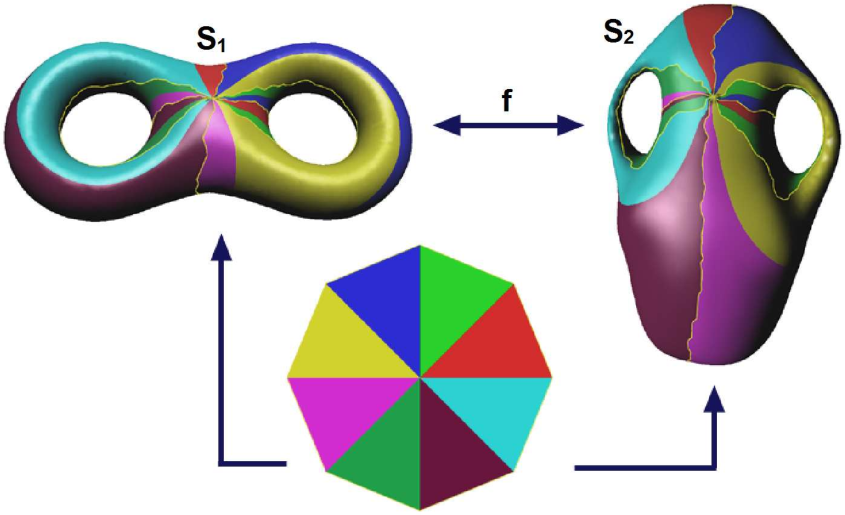


Figure 29: Slice both meshes open and map them to a canonical $4g$-gon; compose these two maps; get the initial mapping. Mappings of different (color-coded) regions are shown respectively with different colors.

Initial Mapping via $4g$-gon:

In: Source surface S_1, target surface S_2, and their systems of loops.

Out: Initial mapping from S_1 to S_2.

1. Slice each surface along its system of loops to open it up onto the $4g$-gon.
2. Flatten each sliced surface to the canonical $4g$-gon, using the harmonic map with fixed boundaries.
3. On the canonical planar parameter domain, map S_1 to S_2 via barycentric coordinates. Unlike [12], we do *not* extract a meta-mesh by overlaying the two planar domains. Instead, we use an approximation mesh S_2' with only the connectivity of S_1 (though its shape is like S_2), and we may later employ an adaptive remeshing procedure (Section 5.6.5) for mapping refinement in areas where under-sampling occurs.
4. Stitch the topological disk S_2' along the original cutting boundary back to the closed surfaces.

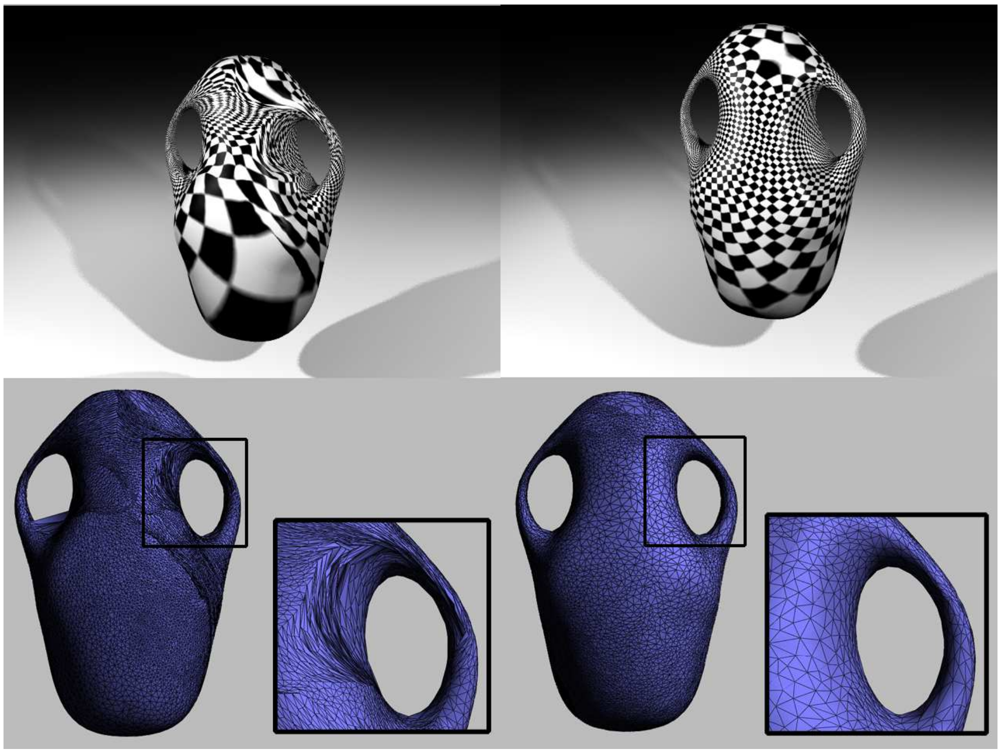


Figure 30: Side-by-side Comparison between Distortions of Initial Map (left) and Optimized Map (right).

5.4 Computing the Uniformization Metric

According to our previous discussion, given a surface $S \subset \mathbb{R}^3$ and its induced Euclidean metric (represented by its first fundamental form $\mathbf{g}$), let $u : S \to \mathbb{R}$ be a globally defined function on S, then $e^{2u}\mathbf{g}$ is another Riemannian metric on S, which is a *conformal metric* to the original induced Euclidean metric.

Riemann uniformization theorem [60] states that for any S, there exists a unique conformal metric, such that it induces constant Gaussian curvature K and zero geodesic curvature,

$$K = \begin{cases} +1 & \chi(S) > 0 \\ 0 & \chi(S) = 0 \\ -1 & \chi(S) < 0 \end{cases},$$

where χ is the Euler characteristic. Such a metric is called the *uniformization metric*.

We compute the uniformization metric $e^{2u}\mathbf{g}$ using the Ricci flow method [49]. Ricci flow is defined as

$$\frac{du(t)}{dt} = -2K(t), \tag{7}$$

where $K(t)$ is the Gaussian curvature induced by the metric $e^{2u(t)}\mathbf{g}$, under the area preserving constraint

$$\int_S d\sigma = \int_S e^{2u(t)} d\sigma.$$

In practice, all surfaces are represented as triangular meshes. Basically, for a triangular face ABC on the mesh with edge lengths a,b,c, we do *not* treat it as a planar triangle in the **Euclidean space**, but rather a triangle in **Hyperbolic space**. All the angles in the triangle can then be calculated using hyperbolic cosine law, and the discrete Gaussian curvature on each vertex is defined as the difference between 2π and the summation of all the corner angles surrounding the vertex.

We associate each vertex v_i with a circle of radius γ_i. Two circles centered at the end vertices of an edge e_{ij} intersect at an angle Φ_{ij}. The edge length of e_{ij} equals $l_{ij} = \sqrt{\gamma_i^2 + \gamma_j^2 + 2\cos\Phi_{ij}}$.

Conformal maps transform infinitesimal circles to infinitesimal circles and preserve the intersection angles among the circles. Therefore, we only modify the circle radii γ_i and keep the intersection angles Φ_{ij}. Let

$$u_i = \begin{cases} \ln\gamma_i & \chi(S) = 0 \\ \ln\tanh\frac{\gamma_i}{2} & \chi(S) < 0 \end{cases},$$

The discrete Ricci flow is similar to the continuous Ricci flow in the form:

$$\frac{du_i}{dt} = -K_i,$$

where K_i is the Gaussian curvature at v_i.

The Ricci flow will converge [57], such that all discrete Gaussian curvatures are constant, and the edge lengths approximate the uniformization metric.

If the surface S is equipped with the uniformization metric, then S can be isometrically and periodically embedded in the following three canonical spaces, the unit sphere for $\chi(S) > 0$, the plane for $\chi(S) = 0$, and the hyperbolic space $\chi(S) <$

0. When $\chi(S) = 0$ the metric is called *flat metric* since curvature is zero everywhere, and when $\chi(S) < 0$, it is called the *hyperbolic metric*. In Figure 26, we demonstrate the hyperbolic embedding of the two-hole torus model.

We use the Poincaré hyperbolic disk model to represent the hyperbolic space $\mathbb{H}^2$. The Poincaré hyperbolic disk is a two-dimensional space defined in the unit disk $\{z \in \mathbb{C} : |z| < 1\}$ on the complex plane $\mathbb{C}$ with hyperbolic metric. The hyperbolic metric is defined as

$$ds^2 = \frac{dzd\bar{z}}{(1-\bar{z}z)^2}.$$

The geodesics (hyperbolic lines) in the Poincaré disk are Euclidean circular arcs perpendicular to the boundary $|z| = 1$. The rigid motions in the hyperbolic plane are the Möbius transformations $z \to w, z \in \mathbb{C}$ with the form

$$w = e^{i\theta}\frac{z - z_0}{1 - \bar{z}_0}, \tag{8}$$

where z_0 is an arbitrary point inside the unit disk, and θ is a rotation angle. This formula rigidly transforms the hyperbolic disk so that the point z_0 is moved to the origin (the middle and right in Figure 26).

5.5 Map Optimization

With the uniformization metric defined on the target mesh S_2, we can perform the heat diffusion procedure to optimize the initial map. Because of the constant curvature distribution under the uniformization metric, our relaxation will not get stuck in local minima. An arbitrary initial map can be used as the start of our optimization procedure; it can be stretched and distorted, or even contain local flip-overs. Our optimization procedure (Section 5.5.2) converges to a unique bijective global optimum robustly; more discussion about this will be given in Section 5.6.4, and the rigorous proof is given in the Appendix.

In Figure 30, we visualize the distortion of the initial mapping from the amphora model to the vase model by texture mapping and displaying the connectivity. The checkerboard texture mapped is distorted (irregular pattern as shown in the top left image) by this initial mapping. This initial mapping, like all methods based on cutting, induces great distortions near the boundary. By relaxing each vertex on the dynamic charts discussed in the following section, we alleviate the distortions all over the mesh and reach a global minimum.

5.5.1 Chart Construction

In order to smooth the mapping between S_1 and S_2, we need to redistribute vertices of S_1 on the domain of S_2 following the heat diffusion flow. We can either embed the whole S_2 onto $\mathbb{C}$ (genus one) or $\mathbb{H}^2$ (higher genus) and perform the redistribution globally; or directly flow over local charts equipped with uniformization metric. In this work, we use the second method: dynamically constructing a set of overlapping local charts on S_2 and perform relaxation within these charts. Compared with using one global patch, the dynamical local charts method has two important **advantages**:

- The vertices may need to flow across the cutting boundary to relax the energy. On one parameter patch domain, it is difficult to perform the relaxation across the boundary.
- Globally embedding the target mesh onto a large patch is numerically less accurate, especially for hyperbolic metric. The local embedding of small charts is more precise.

For each vertex v_1 on S_1, we create at least one chart on S_2 that covers the images of its 1-ring on S_1, meaning that the chart contains all faces onto which the v_1's 1-ring are mapped.

As shown in Figure 31, to construct a covering chart for the 1-ring of a vertex on S_1, we first map the vertices of the 1-ring to S_2. Each vertex in this 1-ring is mapped to a face on S_2. Given this set of faces, we find a patch on S_2 that contains these faces and is homeomorphic to a disk. We first compute an approximate geodesic distance from the 'center' face (red) to all other faces. Then we add the faces to the chart through Breadth First Search (BFS) while maintaining disk topology. Faces which are closer to the 'center' face are given higher priority during the BFS. After the chart has been constructed, we tile it in $\mathbb{C}$ (or $\mathbb{H}^2$, according to the genus of the mesh). In this way, we get a **locally constructed**, yet **globally parameterized** chart, extracted as a small subset of the continuous global parameter domain. During the relaxation, the mapping of vertices and their 1-ring can change,

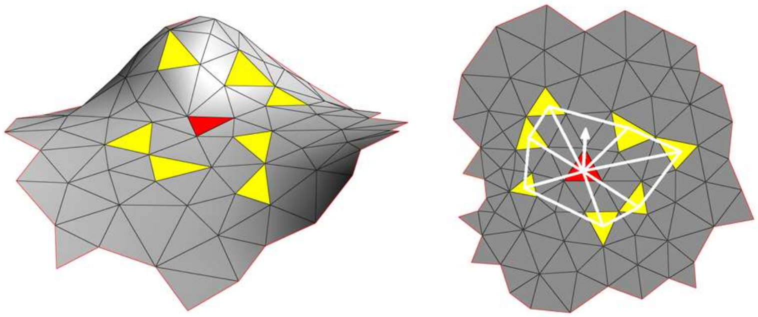


Figure 31: The Dynamic Covering Chart on S_2. Given a vertex V and its one-ring on S_1, the left figure shows a covering chart on S_2: the vertex V is mapped to the red face; its one-ring neighbors are mapped to the yellow faces. The right figure shows the domain of the chart. The white arrow indicates the gradient direction of the harmonic energy.

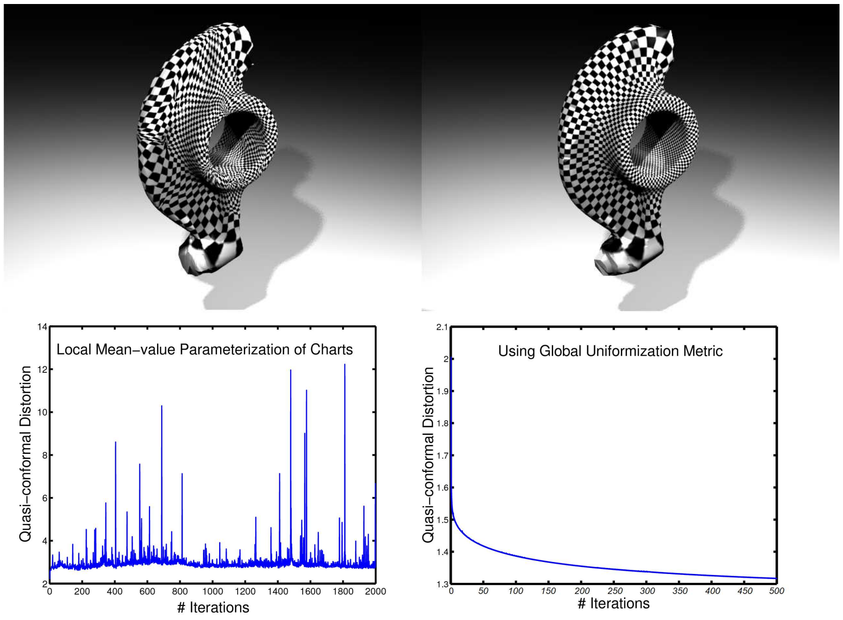


Figure 32: Local Parameterization VS Global Uniformization Metric (Map the Torus to the Rocker Arm). Top row: side-by-side comparison between local approach and global approach. Bottom row: temporal statistics of convergence performance.

new charts are dynamically created when necessary; old charts which are unused for a user-specified amount of time are removed from memory on the fly.

This is Not Local Parameterization. Note that chart-based approaches have been used in local-parameterization-based remeshing [116,124]. And our approach is fundamentally different from them in that we are not *locally parameterizing* these one-ring charts, but directly *embedding* the pre-computed uniformization metric. Local parameterization computes the flattening of charts every time separately, while we use the global metric so that a globally consistent covering is achieved. With the uniformization metric, we trivially get the flattening of each local chart by tiling it in a proper local patch domain (Figure 26). In other words, the pre-computed metric already defines all the edge lengths in the mesh of the given chart; we only conduct a tiling of this triangular mesh.

The relaxation result demonstrates the key difference between local parameterization and our approach: relaxation based on local parameterization will get stuck locally, while using a globally consistent uniformization metric guarantees the global convergence. To demonstrate this, we perform experiments as shown in Figure 32. Compared with the relaxation on the uniformization metric (right), the relaxation using local parameterization[1] (left) will get stuck in some local optimum, and cannot produce the desirable result.

5.5.2 Relaxation

We let the skin 'flow' on the target planar domain so that the harmonic energy is minimized. This is performed via an iterative heat diffusion (relaxation) procedure. The discrete harmonic energy of a map f is defined as

$$E(f) = \sum_i E(f) = \sum_{i,j} w_{ij} |f(v_j) - f(v_i)|^2,$$

where $|\cdot|$ is the norm with respect to Euclidean metric and w_{ij}'s are the discrete harmonic cotangent weights. We use the gradient descent method to minimize the harmonic energy. In each single relaxation step, a vertex is moved in the domain following the gradient of harmonic energy by the Laplacian operator, which is defined as

$$\Delta f = \sum_{j \in N_i} w_{ij} (f(v_j) - f(v_i)),$$

[1] Local parameterization of the charts onto circular disks

where N_i is the index of neighboring vertices and $f = \omega$ is the composed map as given in Equation 6. Therefore, the vertex in the domain is moving towards the new position:

$$f'(v_i) = f(v_i) + \Delta f.$$

During the iteration procedure, the harmonic energy (from the source mesh to the target domain) monotonically decreases. For genus one meshes, which are embedded in $\mathbb{C}$ under uniformization metric, these operators can be used directly. For higher genus meshes embedded in $\mathbb{H}^2$, we still use this Euclidean Laplacian operator to relieve the harmonic energy after an isometric transformation in $\mathbb{H}^2$, which is also called the *Möbius transformation*. The reason that we can approximate hyperbolic Laplacian operators using Euclidean Laplacian operators had been discussed in the previous Section 5.2.4. With the Möbius transformation we rigidly transform the domain of the local chart so that the parameterization of the vertex being relaxed coincides with the center of the Poincarè disk(Figure 26). Near the origin, the hyperbolic metric $ds = \frac{2|dz|}{1-|z|^2}$ only differs by a constant factor from the Euclidean metric and thus our Euclidean Laplacian operator is a linear approximation to the Hyperbolic Laplacian operator in this relaxation region. (The local chart is usually small, so the approximation has high precision.) For numerical issue, we change the f value on each vertex to the target using a step size 0.5, i.e. $f'(v_i) = f(v_i) + \frac{\Delta f}{2}$.

5.6 Discussions on Mapping Performance and Property

5.6.1 Mapping Quality Measurement

Harmonic energy is a natural energy to measure the stretching energy induced by the mapping. A physically meaningful mapping in reality ought to minimize the harmonic energy.

When the conformal mapping from S_1 to S_2 does not exist, the quasi-conformal mapping f maps circular regions around a local point into ellipses. The ratio of the major to the minor axis is called the *dilatation D* at this point. We use a discrete variance D' to measure the conformality of this mapping. The definition is as follows.

Given a local triangle $(q_1,q_2,q_3), q_i = (x_i,y_i,z_i)$ of the original mesh mapped onto a triangle $(p_1,p_2,p_3), p_i = (u_i,v_i)$ on $2D$. The interior discrete mapping $S(p) = S(u,v) = q$ is represented by

$$S(p) = (\langle p,p_2,p_3\rangle q_1 + \langle p,p_3,p_1\rangle q_2 + \langle p,p_1,p_2\rangle q_3)/\langle p_1,p_2,p_3\rangle,$$

where $\langle a,b,c\rangle$ denotes the area of triangle abc. The partial derivatives of the Jacobian are

$$S_u = (q_1(v_2 - v_3) + q_2(v_3 - v_1) + q_3(v_1 - v_2))/(2\langle p_1,p_2,p_3\rangle).$$

and

$$S_v = (q_1(u_3 - u_2) + q_2(u_1 - u_3) + q_3(u_2 - u_1))/(2\langle p_1,p_2,p_3\rangle).$$

The larger singular value Γ and smaller singular value γ of the Jacobian are given respectively [102]:

$$\Gamma,\gamma = \sqrt{\frac{E+G \pm \sqrt{(E-G)^2+4F^2}}{2}},$$

where E,F,G are terms for the first fundamental form.

We compute D' on each triangle using $D' = \frac{\Gamma}{\gamma}$. The maximal value of D' of the mapping on the surfaces is determined by their geometry. As we mentioned above, in the genus one case, the harmonic map we get is the extremal quasi-conformal map that minimizes the angular distortion. We test our mapping performance against the theoretical bound using the following experiment. Given two tori T_1 and T_2; T_1 has minor and major radii 0.5 and 2 respectively, while T_2 has these radii 0.5 and 1.5. T_1 and T_2 can be conformally mapped onto two $2D$ rectangles $R_1(a_1,b_2)$ and $R_2(a_2,b_2)$, where (a_i,b_i) are the width and length of the rectangle. The extremal quasi-conformal mapping between T_1 and T_2 has the lowest theoretical bound given by the modules ratio of T_1 and T_2. In our setting above, these two modules are 0.3531 and 0.5762, meaning the theoretically optimal D' bound between T_1 and T_2 is 1.632. We plot the performance of our mapping in Figure 34(c) (red curve). The x-direction shows the iteration numbers and the y-direction shows the global quasi-conformal distortions.

5.6.2 Homotopy Classes of Initial Mappings

When an arbitrary initial mapping is built up, the homotopy class of the mapping is determined. The subsequent optimization procedure (Section 5.5) reaches a unique optimized result in this same class. The slicing order of loops in the two systems of loops decides the homotopy class of the mapping. Usually, if we arbitrarily pick an order, an optimized result will be reached in that corresponding homotopy class; but this kind of arbitrary surface mapping may not be what we intuitively want. We naturally want handles mapped to handles consistently. To get consistent slicing orders of systems of loops, first, we can compute the canonical handle and tunnel loops using the method of [22]; second, with these handle and tunnel loops, we can decide the homotopy class of each closed loop in the system of loops, this pair loops in two systems of loops, providing the consistent slicing orders in two systems of loops. In this way, we correspond handles in the source surface with handles in the target surface. For two genus g surfaces, there will be g-factorial consistent mappings, any of them is visually reasonable.

Furthermore, in many applications, users may want more precise controls on the mapping. For example, sometimes handles of the source surface need to be mapped to some specific handles of the target surface. Also, users may require some feature points to be mapped. Both of these can be easily implemented in our framework as follows.

5.6.3 Constraints and User Controls

To assure the handle-correspondence, users only need to pick up a corresponding slicing order of two systems of loops, on the $4g$-gon disk, users can easily set up this order once the systems of loops are computed.

In order to have constraints on the feature points, existing parameterization techniques for topological disk surfaces with constraint points, for example, MAPS [78] can be applied for the initial map. Also, many existing surface mapping frameworks [12, 73, 78, 85, 99, 106] allow feature point correspondence, and they can be applied as the initial map. In our work, since we use Carner et al.'s method [12] to generate the initial map, we also apply their method for the initial feature registration.

Starting from the initial mapping with feature points matched correspondingly, we can perform the optimization without relaxing the feature points and prevent any neighboring vertices' movements that violate the validity of the triangular mesh during the relaxation.

The insertion of landmarks definitely may cause larger distortion in the neighborhood, because now the relaxation cannot be performed freely and the resulting mapping is not globally optimized. The detailed discussion about feature correspondence is beyond the focus of this work. We will explore along this direction in our future work.

5.6.4 Global Convergence and Performance

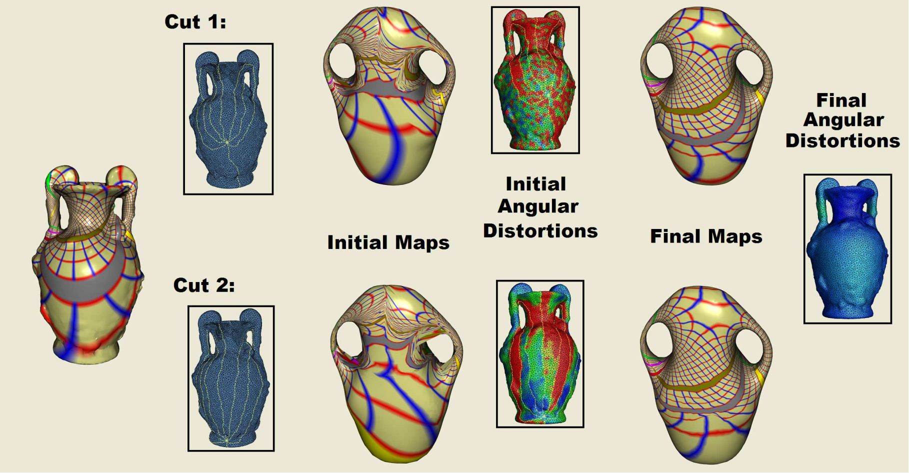
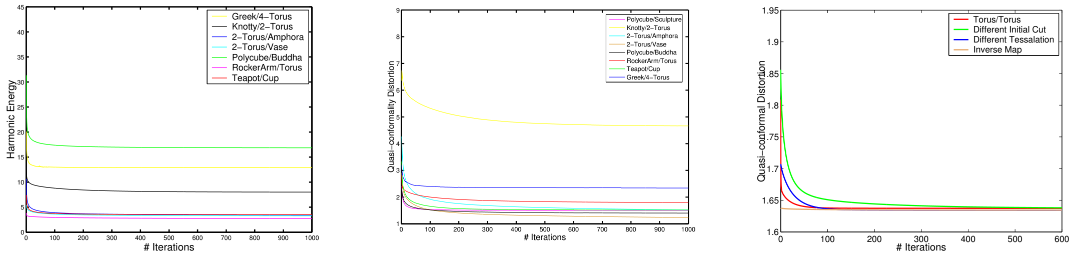


Figure 33: Global Uniqueness of the Optimized Map in the same Homotopy Class.

Our surface mapping optimization converges robustly. Under the same homotopy class, different initial cuttings/mappings reach the same global optimized result. A rigorous proof is given in the appendix. We also perform experiments and visualize this in Figure 33: from left to right, the first column shows the original Amphora model and its texture; the second column are two different initial cut paths. In the third column, we transfer the Amphora's textures onto the target vase model using the corresponding initial maps. Their angular distortion distributions (average D') are color-coded in the fourth column. Transferred textures on the Vase model using the final maps are illustrated in the fifth column. Their final maps are almost the same and have the final distortion color-coded in the rightmost column.



(a) (b) (c)

Figure 34: Mapping Performance. (a) Harmonic energy during iterations. (b) and (c) Global quasi-conformal distortion (average of D') during iterations.

We plot more experimental performances on computations of our mappings in Figure 34: (a) and (b) show the harmonic energy and quasi-conformal distortion convergence during the iteration, respectively. In (c), we perform experiments on genus-1 tori to further quantatively test the robustness and validity of our mapping. The mapping from T_1 to T_2 discussed in the previous Section 5.6.1 with a different initial cutting converges to the same result (green). The mapping from a torus T_1' (different resolution with T_1) to T_2 is plotted in the blue curve. The inverse mapping ($T_2 \rightarrow T_1$, which has the same quasi-conformality bound in optimum) is plotted in brown.

5.6.5 Connectivity Refinement

Since we only use the connectivity of the source mesh S_1, geometry loss may happen in some areas due to under sampling, most likely in high curvature (e.g. sharp feature) areas on S_2. In order to capture such geometric details, we simply apply an adaptive remeshing algorithm similar to [73]. We locally modify the connectivity of the mesh using edge splits, guided by the following two simple error terms which capture the geometric proximity between S_2' and S_2: $E_{length}(e_{ij}) = |\phi(v_i) - \phi(v_j)|$, and $E_{norm}(e_{ij}) = [1 - N(v_i) \cdot N(v_j)]/2$, where $\phi : S_1 \rightarrow S_2$ and $N(v)$ is the normal of vertex v. The first term measures the length of an edge on S_2': longer edges are more likely to miss geometric details and we

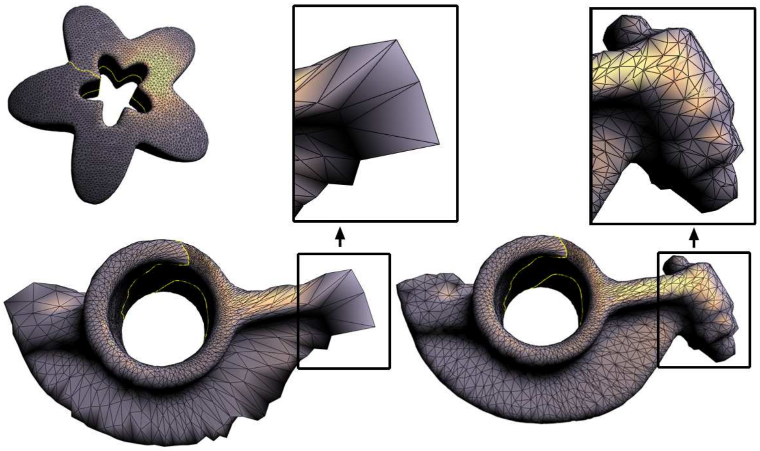


Figure 35: Connectivity Refinement. Left column: the initial mapping from Star model to Rocker-Arm model; Right column: the refined connectivity.

prefer splitting them early. The second term measures the normal deviation of the two vertices of the edge: a greater value implies that the edge crosses a more curved region or a region with sharp features. We iteratively split edges with large combined error. The new vertex generated by the edge split is then mapped back to the surface of the target mesh via the parameterized chart that covers this edge.

In Figure 35, we can see that the model created by mapping the ‘Star’ to the ‘Rocker Arm’, after being refined for 10 iterations, approximates the geometry of the target mesh much better: the left column is the initial mapping while the right column shows the refined connectivity. The number of vertices only increases by a fraction of 11.04%. Our simple error metric is easy to implement as we do not have to maintain the inverse map from S_2 to S_1 in this case.

5.7 Experimental Results

We need an effective way to clearly visualize a mapping between two surfaces because showing region correspondence as well as the distortion are challenging. We use a texture with the color band marks embedded in coordinate lines to aid in this visualization. The texture is first mapped onto the source model, each vertex

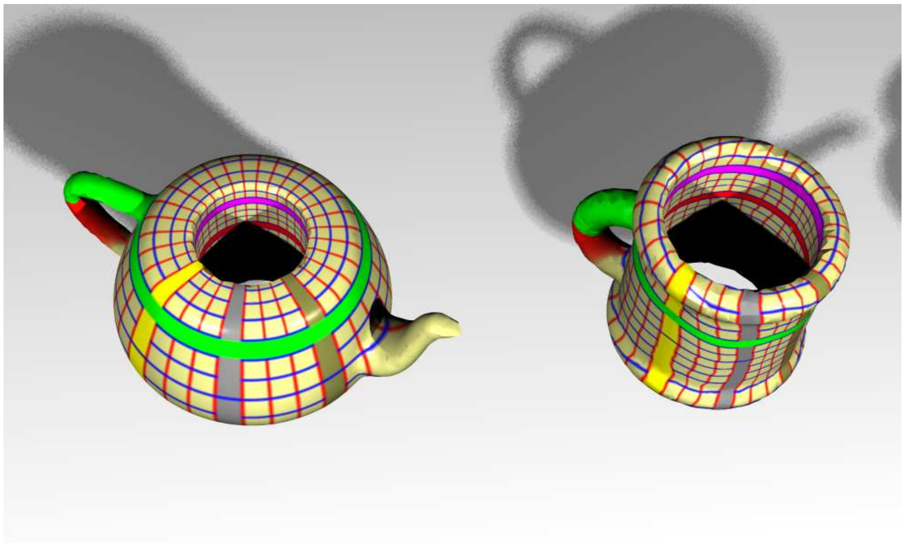


Figure 36: Visualization of Surface Mapping between a Teapot Model and a Cup Model.

on the source surface has its “UV” coordinates. When vertices are mapped to the target surface, their “UV” coordinates are carried. In this way, texture mapping on the source surface is transferred onto the target surface, the color bands on the target surface visualize the region correspondence, and the perpendicularity of the checker board or coordinate lines shows the angular distortion. Figure 36 visualizes mapping effect from genus-2 a teapot model to a cup model. This idea is used for surfaces texture transfer in Chapter 7.

5.7.1 Algorithm Performance

Our optimization is an iterative algorithm; the total number of iteration steps is controlled by a user-defined threshold. In Figure 34, we set the threshold of quasi-conformality to be $1e-6$; in real applications, we can use lower precisions. We perform our algorithm on a MS Windows XP PC with dual Intel Xeon 2.6GHz CPUs, 2GB RAM. The one-iteration running time for most real examples we presented in this chapter are shown in table 2.

Models(S_1/S_2)	Genus	Number of Vertices	Running Time (Seconds)
2-Torus/Vase	2	3.5k/5k	0.31
Amphora/Vase	2	10k/5k	1.4
RockerArm/Torus	1	15k/14.4k	6.88
Teapot/Cup	2	7.5k/10k	0.95
Polycube/Sculpture	3	3.5k/7k	0.49
Greek/4-Torus	4	14.8k/10k	5.07
Polycube/Buddha	6	18k/13.3k	10.23

Table 2: Running Time for Each Iteration of Globally Optimal Surface Mapping.

5.8 Chapter Summary

Comparison with Existing Work. Compared with other state-of-the-art techniques, our approach has several key improvements.

First, our surface mapping framework is based on rigorous mathematical foundation and analysis, unlike most current methods that only guarantee to reach local optima, our method **globally minimizes the stretching** and converges to a **unique** result.

Second, since the existing methods follow the general principle of slicing the surface open into subregions, the initial segmentation directly determines the mapping result. In contrast, our method, because of its global uniqueness, is not controlled by the quality of initial mappings; therefore, it is much more general than other existing techniques.

Third, existing work primarily focuses on low genus surfaces and few take the homotopy types into account. Carner et al. [12] also targeted high genus surfaces, and they studied the mapping with different homotopy classes. However, topology information is the primary information they used for mapping computation and therefore their stretching energy is not optimized. In our current work, the comparison between initial and final mapping shows a great improvement between the initial mapping generated by their method and our globally optimized result. This can be easily visualized through our optimization procedure in the accompanying video.

This chapter has documented our new method for computing a globally optimal map between surfaces of non-trivial topology. Based on the mathematical advances in computing the uniformization metric using intrinsic geometric structure, we can globally perform heat diffusion to alleviate the stretching and the average angle distortion of the map as much as possible. As we discussed above, our algorithm has many key advantages over existing work.

Our mapping algorithm can also serve as a ubiquitous tool for a wider range of applications such as shape registration, morphing, matching, comparison, cross-surface parameterization for spline construction or efficient physically based simulation over regular domains. These applications will be shown in Chapter 7.

Chapter 6

Harmonic Volumetric Mapping using Method of Fundamental Solution

In previous chapters, we extensively studied the limitation of current surface mapping computation techniques, and presented both local and global approaches for surface mapping which contributes to surface mapping computation. In this chapter, we generalize our shape mapping framework from surface data to volumetric data.

Such a generalization is natural. Many scanned data in our surrounding physical world have abundant interior information which could be valuable in many applications. For example, MRI data for clinical analysis are volumetric ones. Therefore, surface mapping which only considers data by their boundary shell is certainly not enough, we often need to compute mappings between volumetric data.

As we introduced in Chapter 2, compared to surface mapping, volumetric mapping is less studied due to its technical challenge, especially the computational complexity. Most existing work lacks efficiency and robustness. In this chapter, we introduce a new meshless boundary method called the *Method of Fundamental Solution* (MFS), which is simple, robust, and efficient.

6.1 Introduction

Despite the necessity of surface mapping techniques, interior volume data carries abundant information including material, density, texture, etc. (beyond pure geometric information). Therefore, not only the thin-shell of the object but the entire solid model should be taken into account in many cases of solid modeling, shape analysis, and physically-based computation. For example, most of the physically-based deformation techniques are volume-driven. Volumetric mapping between objects instead of surface mapping, serves as an enabling and more accurate tool for this task. In spite of this strong need, due to its technical challenges and computational complexity, much less work has been actually carried out in volumetric mapping compared with the surface case. In this work, we aim to pursue a robust, efficient, and accurate algorithm to compute the harmonic volumetric mapping between two solid objects. We make use of the boundary method, in which the behavior of the interior region of the volumetric object is determined only by its surface boundary, thus naturally reducing this volumetric problem to that of its boundary surface scale. The harmonicity of the mapping is guaranteed by the method of fundamental solution.

Harmonicity of a mapping characterizes smoothness, which is a natural phenomenon that depicts the minimized physical stretching energy configuration that arises from the difference between two shapes. In the surface case, harmonic mapping tries to achieve this by vanishing on the source surface the Laplace-Beltrami operator. Intuitively speaking, finding a harmonic mapping between two surfaces with a fixed boundary correspondence is like computing the final deformation of a rubber membrane. The membrane has the source surface as its relaxed shape configuration, and is wrapped onto the target shape with certain fixed boundary constraints. The final mapping that leads to the physically-natural deformation should minimize the harmonic energy and is what the algorithm aims to achieve. Similarly, for harmonic volumetric mapping, we fix the boundary mapping, which is now a surface mapping between shells of the two given solid objects. Then we seek a smooth interior region mapping by enforcing the $3D$ Laplacian everywhere to be zero. This is equivalent to computing the final stable configuration of a solid rubber subject to its boundary shape constraints.

Unlike the surface case, the variational procedure that minimizes a predefined energy needs to adjust a much larger number of points, which usually results in an intolerable computation complexity. According to the maximum principle of harmonic functions, the interior value of a smooth field is determined by its boundary setting. Therefore, we can use a boundary method called the *method of fundamental solution* (MFS) to solve this problem, hence reducing the volumetric solid problem to just the boundary surface scale. To our best knowledge, this is the first work in the computer graphics area that has employed MFS to solve the volumetric mapping problem.

Our specific contributions in volumetric mapping computation are two-fold:

1. We develop a simple and efficient algorithm that can robustly and automatically compute the harmonic volumetric mapping from one volumetric object to another.
2. To the best of our knowledge, this is the first attempt to bring the method of fundamental solution into the graphics and solid modeling community. The technique is an efficient meshless boundary method with great potential. Earlier work provides some theoretical analysis from the point of view of mathematics and mechanical engineering, but it lacks experimental validation. We conduct experiments on the problem of computing harmonic volumetric mapping using this method; and we provide some valuable suggestions for using this method in the modeling field.

We introduce the theory and algorithm of our method in Section 6.2, followed by Section 6.3 with some implementation details and property discussions. Finally, we show our experimental results in Section 6.4 and summarize this chapter in Section 6.5.

6.2 Theory and Algorithm

We pursue a volumetric map $\vec{f}$ from a given solid object M_1 to another object M_2, this is equivalent to building up a smooth one-to-one correspondence between M_1 and M_2. The boundary constraint is a surface mapping $\vec{f}'$ from the boundary surface of M_1, denoted as ∂M_1, to the boundary surface of M_2, ∂M_2.

We focus on objects embedded in $\mathbb{R}^3$. So the mapping $\vec{f}(\mathbf{p}) = \mathbf{q}$ ($\mathbf{p} \in M_1$, and $\mathbf{q} \in M_2$) can be decomposed into three separate components for three axes directions, i.e., $\vec{f} = (f^0, f^1, f^2)$. In each direction, f^i maps the point $\mathbf{p}$ to $\mathbf{q}$'s corresponding component q_i. This problem is then reduced to the computation of three separate $f^i (i = 0, 1, 2)$, with the given boundary mapping constraints $\vec{f'} = (f'^0, f'^1, f'^2)$.

6.2.1 An Intuitive Explanation of Our Idea

We first introduce our idea in an intuitive way through the electrostatics point of view. In each direction, our target is the harmonic function f^i. The harmonicity, or smoothness, can be simulated using an electric field. Suppose we have lots of electronic particles, if we can place them in $\mathbb{R}^3$ as we like, and set arbitrary charge amount on each of them, then we can flexibly control the electric field we get. An important fact is that the potential of electric fields is guaranteed to be harmonic. Therefore, once the electric field generated by this particle system simulates the boundary condition, in other words, the potential on each boundary point $\mathbf{p} \in \partial M_1$ satisfies the boundary constraint $f'^i(\mathbf{p})$, then we can use the potential of this particle system to simulate the mapping component f^i in the interior region.

Electric fields provide a correct simulation for harmonicity because its potential satisfies the vanishing Laplacian operator everywhere, except for the positions of those particles, where the potential is infinite. Therefore, we call these particles *singularity points* or *source points*; and in order to make the potential valid everywhere in the interior region of M_1, we should place them outside of M_1.

Since we know the harmonicity is guaranteed, we are only left to enforce the boundary conditions f'^i. We fix positions of all particles, and the charge amount carried on each particle provides the freedom we have for enforcing the boundary constraints. This fitting process, as we will show in the coming Section 6.2.2, leads to a linear system. We place many estimation points on the boundary ∂M_1, and compute the charge amount distribution which can result in the desired potential given by f'^i. By solving a linear system, we obtain the best fit charge amount distribution, and get the simulation of f^i.

Intuitively, if we have dense enough particles placed outside of M_1, the smooth boundary condition can always be well approximated, only with exceptions in some extremely discontinuous boundary regions. For more details, we refer readers to [32].

This section gives an intuitive explanation and pipeline of our idea. In the coming section, we will rigorously formulate our algorithm, and address its theoretical foundation.

6.2.2 Formulations

We rigorously formulate our volumetric mapping problem as follows:

Given a one-to-one mapping $\vec{f}'$ between boundary surfaces ∂M_1 and ∂M_2: $\vec{f}'(\mathbf{p}) = \mathbf{q}$, $\mathbf{p} \in \partial M_1, \mathbf{q} \in \partial M_2$, our goal is to compute a mapping $\vec{f} : M_1 \to M_2$ such that

$$\begin{cases} \Delta\vec{f}(\mathbf{p}) = 0 & \mathbf{p} \in M_1, \\ \vec{f}(\mathbf{p}) = \vec{f}'(\mathbf{p}) & \mathbf{p} \in \partial M_1. \end{cases}$$

where the operator Δ is defined continuously in $3D$ as

$$\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2},$$

and $\Delta\vec{f} = 0$ for $\vec{f} = (f^0, f^1, f^2)$ is equivalent to $\Delta f^i = 0$ for all $i = 0, 1, 2$.

Harmonicity and Kernel Function. Since Δ is a linear self-adjoint differential operator, we can compute its Green Function. We denote $g^i(\mathbf{x}) = \Delta f^i(\mathbf{x})$, and denote Δ^{-1} as the inverse of the operator Δ, so that $\Delta\Delta^{-1} = I$ where I is the identity operator; then we can write the solution as $f^i(\mathbf{x}) = \Delta^{-1} g^i(\mathbf{x})$.

Note that $f^i(\mathbf{x}) = \Delta^{-1} g^i(\mathbf{x}) = \Delta^{-1} \int \delta(\mathbf{x} - \mathbf{x}') g^i(\mathbf{x}') d\mathbf{x}'$, where δ is the Dirac function. If we make a *kernel function* $K(\mathbf{x}, \mathbf{x}')$ that satisfies $\Delta K(\mathbf{x}, \mathbf{x}') = \delta(\mathbf{x} - \mathbf{x}')$. We can rewrite the solution to f^i following the above equation in terms of the Kernel function as

$$f^i(\mathbf{x}) = \int K(\mathbf{x}, \mathbf{x}') g^i(\mathbf{x}') d\mathbf{x}'.$$

Such a kernel function K is known as the Green's function associated with the 3D Laplacian operator Δ, and has the formula: $K(\mathbf{x}, \mathbf{x}') = \frac{1}{4\pi} \frac{1}{|\mathbf{x} - \mathbf{x}'|}$, where $|\mathbf{x} - \mathbf{x}'|$ denotes the distance between the points $\mathbf{x}$ and $\mathbf{x}'$.

Since f^i in the interior region is fully determined by the boundary values, we discretely solve it using *Method of Fundamental Solution* (MFS) [26] with the above kernel. The linear nature of the Laplacian operator indicates that a boundary-based method such as MFS is most suitable since the interior harmonicity is now represented in an exact manner; we only need to enforce the given boundary condition function $\vec{f'}$, i.e., f'_i for each f^i. The MFS approximation equation to evaluate f^i on an interior or boundary point $\mathbf{p}$ is

$$f^i(\vec{w^i}, \vec{\mathbf{Q}}; \mathbf{p}) = \sum_{n=1}^{N_s} w_n^i \cdot K(\mathbf{p}, \mathbf{Q_n}), \mathbf{p} \in M_1. \tag{9}$$

In this above equation, $\vec{\mathbf{Q}}$ is a $3N_s$-dimensional vector concatenating positions of all N_s three-dimensional source points. $\vec{w^i} = (w_1^i, w_2^i, \cdots, w_{N_s}^i)^T$ is the N_s-dimensional vector representing the charge amount distribution on these source points. It is firstly unknown and is what we want to solve.

Note that source points $\mathbf{Q_n} \in \mathbb{R}^3, n = 1, \cdots, N_s$ should lie outside of M_1, in other words, they are located on the boundary $\partial\widetilde{M_1}$ of a region $\widetilde{M_1}$ containing M_1 (i.e. $M_1 \subset \widetilde{M_1}$). Once no source points are inside M_1, an arbitrary charge distribution $\vec{w}$ can guarantee in an exact manner the vanishing Laplacian operator on f^i in the interior region of M_1, only violating the boundary conditions.

Boundary Fitting. To enforce the boundary conditions, we pursue a special set of charge amounts $\vec{w}$, such that the images of points on ∂M_1 satisfy the boundary map $\vec{f'}$. This boundary fitting process is performed as follows, we sample a set of estimation points on the M_1's boundary surface ∂M_1. These points are called *constraint points* or *collocation points*. Unlike the aforementioned source points which are electronic particles and will be used for estimating electric potentials in interior region of the solid object, collocation points are different types of points; the configuration of source points does not affect the configuration of collocation points and vice versa.

Boundary conditions are represented by enforcing collocation points to be mapped to the boundary of the target model ∂M_2. Their images are given with $\vec{f'}$. For example, for N_c number of collocation points $\mathbf{P_1}, \mathbf{P_2}, \cdots, \mathbf{P_{N_c}}$: in each axis direction, we denote their image vector as $\vec{b^i} = \{b^i{}_1, b^i{}_2, \cdots, b^i{}_{N_c}\}^T = \{f'^i(\mathbf{P_1}), f'^i(\mathbf{P_2}), \cdots, f'^i(\mathbf{P_{N_c}})\}^T$. According to Equation 9, this vector $\vec{b^i}$ can be represented by $\mathbf{A}\vec{w^i}$, where $\mathbf{A}$ is called the *coefficient matrix*, whose element $A_{uv} = K(\mathbf{P_u}, \mathbf{Q_v})$ (P_u is a collocation point while Q_v is a source point). These images should satisfy the boundary condition, i.e., equal to $\vec{b^i}$. Therefore, the above fitting process reduces to a linear system on each axis direction:

$$\mathbf{A}\vec{w^i} = \vec{b^i}.$$

We solve three linear systems in three different axis directions separately. The resulting vectors (w^0, w^1, w^2) corresponding to the three different charge distributions are used to compose our final harmonic volumetric mapping.

6.2.3 Algorithm

With the discussion above, we can formulate our algorithm. The input are two given solid objects M_1, M_2 and their boundary surface mapping $\vec{f}' : \partial M_1 \rightarrow \partial M_2$. The output is a harmonic volumetric mapping $\vec{f} : M_1 \rightarrow M_2$ s.t. on boundary $\vec{f}(\mathbf{p}) = \vec{f}'(\mathbf{p}), \mathbf{p} \in \partial M_1$ and in the interior region: $\frac{\partial^2 \vec{f}}{\partial x^2} + \frac{\partial^2 \vec{f}}{\partial y^2} + \frac{\partial^2 \vec{f}}{\partial z^2} = 0$. $\vec{f}$ is decomposed to (f^0, f^1, f^2), and we solve each f^i separately as follows.

Harmonic Volumetric Mapping.

In: $M_1, M_2, \vec{f}'$.

Out: f^i for $i = 0, 1, 2$.

1. Place the source points and the collocation points. (Section 6.3.1 and Section 6.3.7).
2. Compute the coefficient matrix $\mathbf{A}$. Its element $\mathbf{A}_{uv}$ takes the value of the kernel function K on the collocation point $\mathbf{P_u}$ and the source point $\mathbf{Q_v}$.
3. Decompose the coefficient matrix using Singular Value Decomposition. (Section 6.3.2 and Section 6.3.6).
4. Solve this linear system under the given boundary mapping constraints f'^i using the decomposition result from Step 3, and get f^i represented by $\vec{w^i}$ in the form of Equation 9.

The resulting volumetric mapping is harmonic, guaranteed by the kernel function. It minimizes the harmonic energy, which will be discussed in Section 6.3.3. We assume the boundary surface mapping $\vec{f}'$ is given as an input, and in Section 6.3.4, we briefly discuss how to obtain this surface mapping with existing techniques.

6.3 Implementation and Discussion

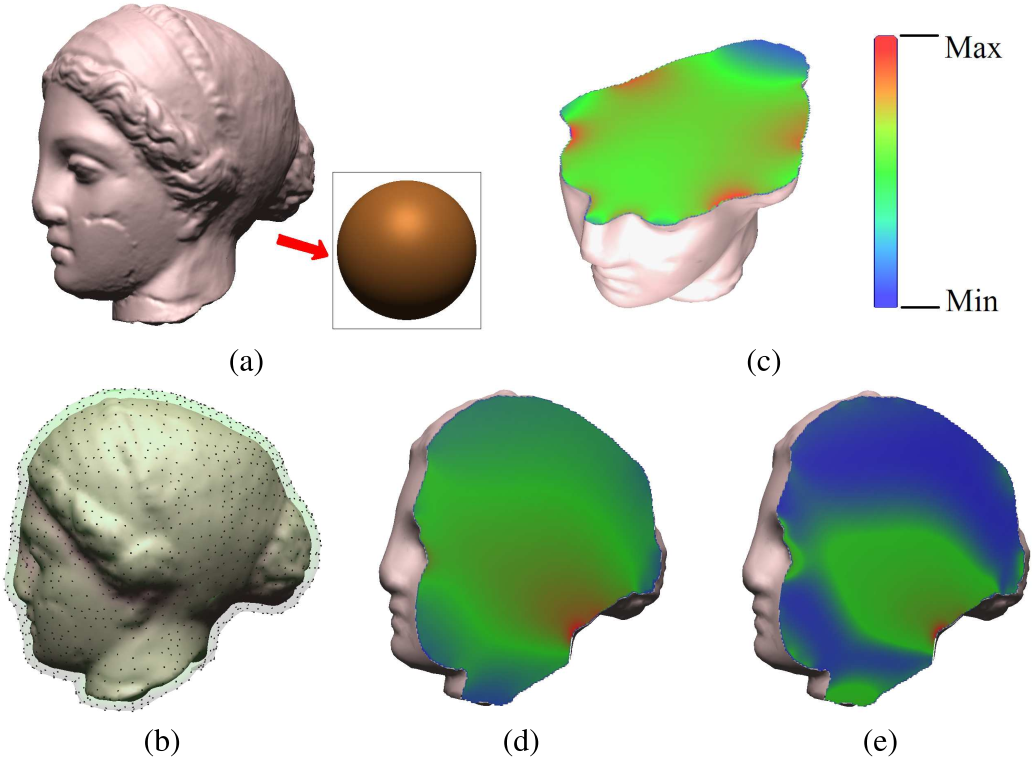


Figure 37: Volumetric harmonic mapping from the solid Igea model to the solid sphere model. (a) shows the source and target objects. As shown in (b), source points are placed on an offset surface. The harmonic energy distribution of the mapping is color-coded and illustrated on two different cross-sections ((c) and (d)), and the deformation energy distribution is illustrated over a cross-section ((e)).

6.3.1 Source Points and Collocation Points Placement

In the first step of our algorithm, we place source points uniformly on an offset surface $\partial\widetilde{M_1}$ outside the boundary surface ∂M_1, as shown in Figure 37(b). The following procedure is a robust way to create such a sampling.

We place collocation points by sampling boundary points $\mathbf{p} \in \partial M_1$ uniformly. The reason that we conduct the source and collocation points placement in this way will be discussed in Section 6.3.7.

Offset Surface Generating Algorithm.

In: M_1

Out: A uniform sampling on the offset surface $\partial\widetilde{M_1}$.

1. Compute the implicit distance field in $\mathbb{R}^3$ with respect to the given object boundary surface ∂M_1 using technique introduced in [76]. We get a distance evaluation function $dist(\mathbf{p})$ so that the signed distance from any point $\mathbf{p} \in \mathbb{R}^3$ to ∂M_1 can be computed efficiently.
2. Build an offset surface $\partial\widetilde{M_1}$ using Bloomenthal's polygonization method [9]. The Bloomenthal's polygonization method takes an implicit distance evaluation function $dist2(\mathbf{p})$ defined in $\mathbb{R}^3$ as the input. Therefore, to build the offset surface $\partial\widetilde{M_1}$ with distance d to ∂M_1, we set such input function $dist2(\mathbf{p}) = dist(\mathbf{p}) + d$.
3. Uniformly sample n points on mesh $\partial\widetilde{M_1}$. We use the uniform sampling technique introduced in [93].

6.3.2 Solving the Linear System

As discussed in Section 6.2.2, we want to solve the linear system $\mathbf{A}\vec{w^i} = \vec{b^i}$. The element in the coefficient matrix $\mathbf{A}$ is the value of the kernel function on each collocation point, which is almost never zero, making the matrix quite dense. The matrix may be ill-conditioned [100], in which case, regular linear system solvers such as Gaussian elimination or LU decompositions usually fail to produce a stable solution. We use *Singular Value Decomposition* (SVD) because it approaches accurate and stable results even when the coefficient matrix is highly ill-conditioned. Another advantage of using SVD is that once we have decomposed the matrix, we can reuse the result for the rapid re-computation of new mappings whenever boundary conditions change. This efficiency also arises from the boundary method, and a detailed discussion about this aspect will be given in Section 6.3.6. More advantages of using SVD in MFS are discussed in [100].

6.3.3 Energy of Volumetric Mapping

Harmonic Energy. Harmonic Energy measures the smoothness of the mapping. It is measured by the integration of the square of the gradient over the entire interior region domain M_1. Both the source and the target models should be normalized to unit size in volume before computing the harmonic energy. The total harmonic energy of this mapping is

$$\int_{M_1} < \nabla\vec{f}, \nabla\vec{f} > d\mathbf{x}. \tag{10}$$

We built a volumetric grid structure and compute the gradient of the mapping on each grid point, then use the following formula to approximate the energy:

$$\sum_{\mathbf{p_{u,v,w}} \in M_1} ||\nabla\vec{f}(\mathbf{p_{u,v,w}})||^2 \cdot vol(\mathbf{p_{u,v,w}}), \tag{11}$$

where $vol(\mathbf{p_{u,v,w}})$ is the interior volume on grid point $\mathbf{p_{u,v,w}}$, and $vol(\mathbf{p_{u,v,w}})$ equals the volume of the intersection of M_1 and the small grid cube $Cube_{\mathbf{p_{u,v,w}}}$ centered at $\mathbf{p_{u,v,w}}$. Here, the edge length of $Cube_{\mathbf{p_{u,v,w}}}$ is the distance between two adjacent grids. We can use the volume of $Cube_{\mathbf{p_{u,v,w}}}$ to approximate $vol(\mathbf{p_{u,v,w}})$. With the increase of the grid sampling density, the value of Equation (11) is asymptotic to Equation (10). We use the simple volume grid data structure because it is easy to implement and efficient in tracing function values on neighboring grid points: the gradient of the harmonic mapping on each grid point can be represented by three vectors: $\nabla\vec{f} = (\nabla f^0, \nabla f^1, \nabla f^2)$, so $||\nabla\vec{f}(\mathbf{p_{u,v,w}})||^2 = ||\nabla f^0||^2 + ||\nabla f^1||^2 + ||\nabla f^2||^2$. In the example of Figure 37, the harmonic energy distribution of the volumetric mapping is colorized in (c) and (d) over two different cross-sections. The color-coding scheme in this chapter is shown in the bar in (c): red represents the maximum while blue represents the minimum.

Deformation Energy. Once a correspondence between two solid objects is created, deformation around each interior voxel point can be estimated easily. This provides us a formal mechanism to compute the energy required to deform one object to another. In Chapter 8 we will use this energy to measure the difference between two shapes.

To compute the deformation energy, we start from the classical strain and stress tensor analysis. Green's strain tensor ε is used to quantify the local strain undergoing a 3-dimensional deformation. If an interior point $\mathbf{p}$ is mapped to $\mathbf{q}$, then the 3×3 tensor ε has its elements ε_{ij} represented by

$$\varepsilon_{ij} = \frac{\partial \mathbf{q}}{\partial p_i} \cdot \frac{\partial \mathbf{q}}{\partial p_j} - \delta_{ij},$$

where $0 \leq i, j \leq 2$ are indices in axis directions, δ_{ij} is the Kronecker delta:

$$\delta_{ij} = \begin{cases} 1: & i = j \\ 0: & i \neq j \end{cases} .$$

According to differential geometry, this strain tensor is invariant under rigid transformation and vanishes under identity mapping. The stress tensor represents the information of the internal forces under the deformation. A simplified linear form of elastic stress with the assumption of isotropy is defined as

$$\sigma_{ij} = \sum_{k=1}^{3} \lambda \varepsilon_{kk} \delta_{ij} + 2\mu \varepsilon_{ij},$$

where λ and μ are two Lamé constants of material, respectively, representing rigidity and resistance to volume dilation change. For example, in most of our experiments, we make use of the parameter of rubber, i.e., $\lambda = 0.0335, \mu = 0.0224$. Finally, the elastic potential density η on this point $\mathbf{p}$ is measured by

$$\eta(\mathbf{p}) = \frac{1}{2} \sum_{i=1}^{3} \sum_{i=1}^{3} \sigma_{ij} \varepsilon_{ij},$$

representing the internal elastic energy under the shape change. Similar to Equation (11), the total deformation energy of this volumetric mapping is computed by

$$\sum_{\mathbf{p_{u,v,w}} \in M_1} \eta(\mathbf{p_{u,v,w}}) \cdot vol(\mathbf{p_{u,v,w}}). \qquad (12)$$

Figure 37(e) color-codes the deformation energy distribution of the volumetric mapping from the solid Igea model to the solid sphere.

6.3.4 Boundary Conditions

The boundary condition of our harmonic volumetric mapping is a surface mapping between ∂M_1 and ∂M_2. We assume that it is provided as an input. Existing surface mapping techniques [62, 73, 78, 85, 99, 106, 130] can be used to create the boundary surface mapping. On one hand, the efficiency of the surface mapping is important. Given the boundary map, our following algorithm is fast and fully automatic; therefore, to make the entire volumetric mapping pipeline efficient and automatic, we hope this surface mapping creation is simple and automatic as well. On the other hand, surface mapping with relatively low distortion is preferred. As we know, the quality of a harmonic mapping depends on the boundary condition. How to generate a suitable boundary surface mapping and how the quality of this mapping affects the volume mapping are worthy of more comprehensive research studies. Since the detailed discussion goes beyond the scope of this work, we will only briefly explain how we get a relatively good initial surface mapping. We consider mapping between solid objects with the same topology, that is, the objects have pairs of corresponding boundary surfaces. Consider the mapping between each surface pair: (1) If the boundary surfaces are closed genus-0 surfaces, the conformal surface mapping suffices. This mapping can be computed/combined through a common sphere domain, similar to techniques in [41] and [62]. (2) If the surfaces are of higher genus, we prefer a globally smooth mapping. In this work, we compute their quasi-conformal mapping as the initial boundary condition using techniques introduced in Chapter 5([81]).

To demonstrate the different volumetric mappings results from different boundary surface mappings, we perform experiments and show results in Fig 38. The harmonic volumetric mappings from the solid Teapot model to the solid Cup model have different harmonicity under two different boundary surface mappings (as shown in (a)). We render the mesh connectivity for the points on the target boundary surfaces to better visualize the differences ((c) and (e)). The second boundary mapping (e) is smoother than the first one (c); it leads to a volumetric mapping (f) with smaller harmonic energy. The harmonic energy distributions for two volumetric mappings are visualized on the Teapot model from the same cross-section. The color-coding scheme for (d) and (f) is depicted in (b).

In the future work, we plan to examine the technical issue of how the surface mapping and the volumetric mapping are related in a quantitative way, and how one mapping guides the computation of the other. By adjusting the surface boundary mapping condition accordingly, we will pursue the free boundary volumetric mapping with minimum harmonic energy.

6.3.5 Comparison with Previous Work

We compare our mapping results with the method introduced in [42]. In their work, the discretized harmonic energy is defined on the tetrahedral mesh to guide their variational procedure. Once we compute our mapping, we can evaluate the mapping on any interior point using Equation (9). We tetrahedralize our volume data (in our work we produce the tetrahedralization using [112]), then compare our results with Gu et al.'s work in [42]. As shown in Figure 39, the volumetric mapping from the solid Igea model (a) to a solid sphere can be visualized by transferring the tetrahedral mesh of the Igea to the solid sphere. Our resulting tetrahedral mesh on sphere (b) appears to be smoother than the mesh produced in [42] (c). This smoothness is visualized from the distributions of the discretized harmonic energy [42] of volumetric mappings, which are color-coded in (b) and (c) using a uniform scheme (d).

Another important advantage of our algorithm is the meshless property. The discretization accuracy and the computational cost of [42] depend heavily on the tetrahedralization quality of the source object: dense tetrahedralization necessarily results in high computational complexity; and irregular tetrahedralization leads to large numerical error in approximating discretized harmonic energy. In contrast, our algorithm is independent of the connectivity, and thus is more flexible and can be adapted to any volumetric data set with spatial-varying resolution.

6.3.6 Computational Efficiency

The computation cost of our algorithm is equivalent to solving three linear systems. More importantly, since we decompose the coefficient matrix A from the MFS using Singular Value Decomposition (Section 6.3.2), only one decomposition is necessary. Furthermore, it is very efficient to recompute the volumetric mapping under different boundary conditions.

Given a new boundary configuration $\vec{b'}$, the corresponding $\vec{w'}$ for the new volumetric mapping can be computed directly from $A^{-1}\vec{b'}$. With the decomposition results, $A^{-1} = VW^{-1}U^T$, where the matrix W^{-1} is a diagonal matrix that can be computed directly from W. Therefore, under a new boundary condition, the decomposition matrix results can be reused, only a multiplication operation between

A^{-1} and $\vec{b}'$ is required.

This shows one more advantage of the boundary method over variational methods which apply iterations on the entire volume whenever the boundary condition is given. Under a new boundary condition, variational methods can not avoid a time-consuming re-computation. In our applications shown later in Chapter 8, we take full advantage of our computational efficiency to compute a large number of sequential volumetric mappings in a temporal deformation sequence, by decomposing A only once.

6.3.7 Source, Collocation Points, and Mapping Efficiency

One important issue that we have to address is how many source points we need to use and where to place them. Using our electric field model, imagine that we want to refine our control of the electric field behavior, the more particles (i.e., source points) we have, naturally the more refined result we should be able to get with increasing computational complexity. On the other hand, numerically, their positions also matter. If all source points are placed in one position, there would be no way that we can achieve more satisfactory results with more source points. The positions of these source points actually determine the behavior of the coefficient matrix **A**, which can be highly ill-conditioned [72]. The condition number of the matrix generally increases as the distance from $\widetilde{M_1}$ to M_1 increases, though the accuracy of the MFS approximation increases under this situation [33]. That is to say, distant source points give a smoother approximation, but unavoidably introduce larger numerical error. Theoretically optimal results of source positions are unknown at present; current literatures either suggest placing source points uniformly on a sphere within three times the diameter of M_1 [33] [11] or on an offset surface of M_1 [119]. Real-world computations in the mechanical engineering field usually choose the source and collocation points in a trial-and-error manner or with the help of human experiences. Inspired by the above pioneering work, we use experimental results to find a suitable setting rule for our mapping problem, and guide the source and collocation points' placement in order to bridge the gap between theoretical results and practical common senses.

We conduct experiments in the following three aspects to find a suitable configuration for our volumetric mapping problem:
(1) the shape of the surface $\widetilde{M_1}$ (source points are sampled on an offset surface or a sphere);
(2) the distance from M_1 to $\widetilde{M_1}$;
(3) the number of the source points and collocation points.

The experimental results are shown in charts plotted in Figure 41. In this figure, Chart (a) plots the boundary constraint error when source points are placed on a sphere (see Figure 40(a)), while Chart (b) shows the case when source points are on an offset surface (see Figure 40(b)). In Chart (a), the x-axis is the radius of the sphere, denoted as *R-Ratio*, represented by the ratio of the sphere radius over the object size, and y-axis shows the boundary constraint error, denoted as *C-Error*. C-Error is computed using: $\sum_{\mathbf{p}} ||f'(\mathbf{p}) - f(\mathbf{p})||^2$ for all collocation points $\mathbf{p}$. C-Error measures the total fitting error of our volumetric mapping to the given boundary constraints. Therefore, we use its value to measure the quality of our mapping. Chart (b) shows the case that source points are placed on the offset surface; the x-axis is the distance from $\partial\widetilde{M_1}$ to ∂M_1; its value, denoted as *O-Distance*, is the ratio of the distance over the source model size. Their corresponding C-Errors are plotted in y-axis. Chart (c) shows the harmonic energy values (y-axis) under the different offset surface settings (x-axis).

Our statistical data demonstrates that: (1) The closer to the model boundary source points are placed, the smaller the boundary constraint error can be achieved. (2) Placing source points on the sphere is not as good as on an offset surface. Because we require that the object be totally inside the interior of the sphere, the radius of the sphere needs to be large enough and the average distance will be much larger compared with the offset surface placement. (3) If source points are placed on an offset surface that is too close to the model, the approximation for the fundamental solution is becoming unstable, which is shown from the values of their harmonic energies (Chart (c)). Therefore, in our experiments and applications, we usually place source points on an offset surface with 0.1 O-Distance.

Chart (d) further shows how the numbers of source points and constraint points affect the boundary constraint errors. We define two ratios *cRatio* and *sRatio*, respectively. The cRatio is defined as the number of collocation points over the number of boundary points. The sRatio is defined as the number of source points over the number of boundary points. The x-axis is the sRatio, and the y-axis shows the boundary constraint error. Different curves show the cases using different cRatio. We can clearly see from this chart: the larger these ratios are, the smaller the boundary constraint error will be. On the other hand, fewer source points create an over-constrained system which will be solved in a much shorter time. In our experiments, we usually set cRatio larger than 0.8 but sRatio around 0.6 for an efficient but well-fitted result for large models.

Unlike fixing the source/collocation points as discussed above, the positions of source points and collocation points can also be considered as unknowns in an optimization procedure, in which case they have to be computed along with the unknown weights during the optimization procedure. This necessarily complicates the entire solver and makes the computation procedure highly non-linear.

Near a boundary region whose target shape is seriously wrinkled, the harmonic mapping may map interior points to the outside of the target object if the source/collocation points nearby are not dense enough. Such a situation can be effectively remedied by increasing the density of source/collocation points around this region adaptively.

6.4 Experimental Results

We first show some experimental results of harmonic volumetric mappings in Figure 42, Figure 43, and Figure 44. In Figure 42, a solid Pierrot model (a) is mapped to a solid sphere (b). The mapping result can be visualized using (c) and (d). In (c), the distance field of the interior region of the solid sphere is color-coded using the scheme shown in Figure 37(c). Here in Figure 42(d), each volume point $\mathbf{p}$ in solid Pierrot model (a) is mapped to an interior point $\mathbf{q}$ in solid sphere model (b). We transfer the color of $\mathbf{q}$ to the position of $\mathbf{p}$. This color-coded distance field on source model transferred from the target model provides an intuitive way to visualize the volumetric mapping result. We call this visualization method *Color-coded Distance Field Transfer*. Another mapping example from the solid Buddha model (e) to the solid sphere (b) is computed and visualized in the same way as shown in (f). One more example, mapping from the solid Max-Planck model (g) to the solid sphere is visualized similarly in (h). We also use another method, tetrahedral mesh, to visualize volumetric maps. In (i), a tetrahedralization of the solid Max-Planck model (g) is illustrated in one cross-section. Under the volumetric mapping, each vertex of the tetrahedral mesh is mapped to a new position inside the solid sphere. Its corresponding tetrahedral mesh cross-section is shown in (j).

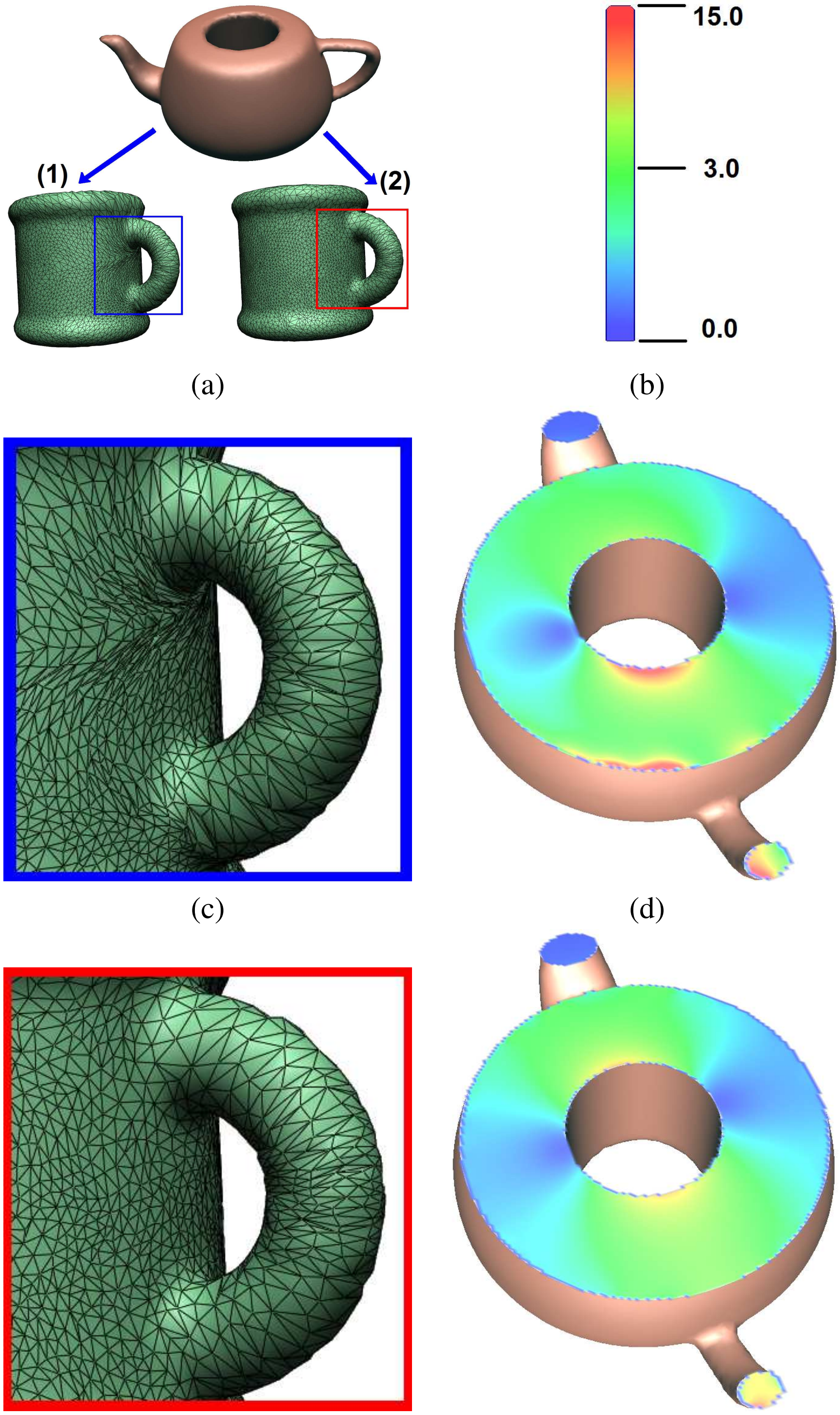
In Figure 43, we visualize the volumetric mapping from a solid polycube model (a) to a solid Buddha model (e). We color-code the distance field of the interior region of Buddha and show it from three cross-sections in (b), (c) and (d); (f), (g) and (h) correspondingly show the *Transferred Color-coded Distance Field*. Figure 44 shows another high genus volumetric mapping example from a Sculpture model (a) to the polycube model (b). (c) color-codes the distance field of the polycube while (d) shows the *Transferred Color-coded Distance Field*.

6.5 Chapter Summary

Based on *the method of fundamental solution* (MFS), we design a simple, robust, and fully automatic meshless algorithm to compute harmonic volumetric maps. To the best of our knowledge, it is the first attempt to bring this method into graphics and geometric modeling community. We conduct experiments to evaluate the performance of the method of fundamental solution on the harmonic volumetric mapping problem in this chapter; accordingly, we suggest the practical rules and develop the effective algorithm on the MFS settings.

Building a correspondence between solid models and canonical/regular objects provides a natural mechanism to facilitate scientific computations and graphical simulations. If we exploit the regular structure of mapped volumetric domains (such as polycubes) and utilize graphics hardware acceleration, physically-based simulations (such as simulating volumetric solid deformations or fluids in deformable bodies) can be efficiently performed.

As discussed in Section 6.3.4, our current harmonic volumetric map depends on the boundary surface mapping. In Figure 38, we show that the volumetric mapping and its boundary surface mapping are closely related to each other. The harmonic energy of the volumetric map keeps decreasing with boundary surface mapping getting more smooth. In the future work, we plan to use the harmonic volumetric mapping to guide the variational process of surface mapping towards the global energy optimization (both for boundaries and solid interiors). Another possible extension is not to fix the positions of source points and collocation points. We can treat them as unknown variables in the MFS procedure. Although this results in a nonlinear optimization process, it may also lead to a free-boundary volumetric mapping procedure for better mapping results.



(a) (b)

(c) (d)

(e) (f)

Figure 38: Different boundary conditions lead to different volumetric mapping results even for the same target object. Volumetric mappings from the solid Teapot model to the solid Cup model (a) under two different boundary conditions ((c) and (e)) have different harmonicity as shown in (d) and (f) (energy distributions depicted on the Teapot model, respectively). (c) and (e) highlight different surface mappings with magnified views.

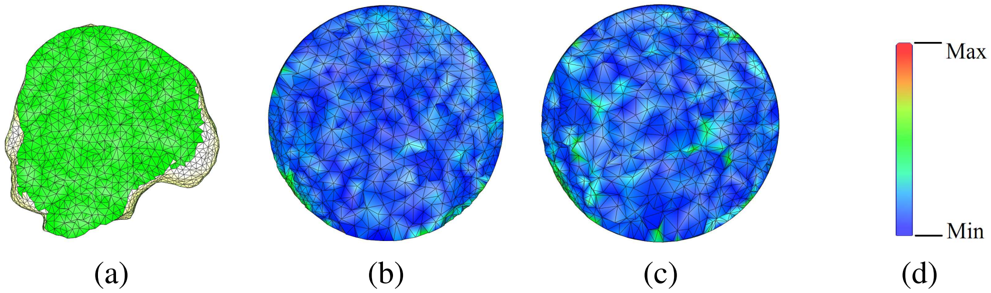


Figure 39: Comparison with previous work. The initial tetrahedralization of the solid Igea model is shown in (a). In (b), the tetrahedral mesh on the solid sphere is transferred from the Igea model using our volumetric mapping algorithm. The result computed using Wang et al.'s variational technique is shown in (c). The harmonic energy distributions of two volumetric mappings are color-coded on resultant tetrahedral mesh using a uniform color-coding scheme as shown in (d).

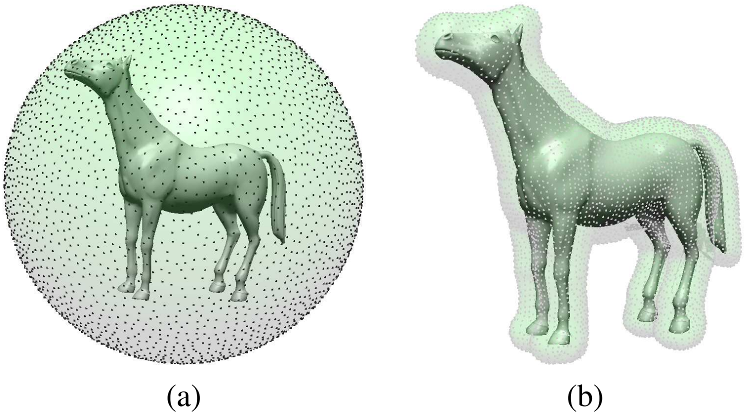
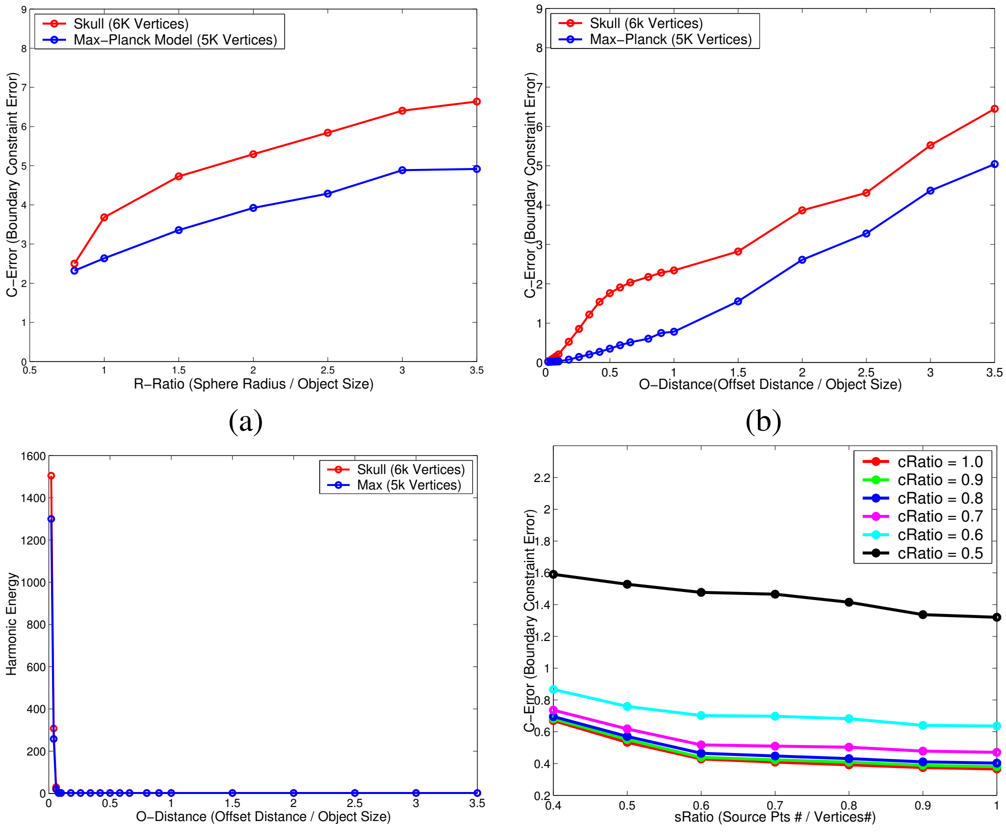


Figure 40: Placement of Source Points. Source points are sampled either on a bounding sphere (a) or on an offset surface of the given model (b).



(a) (b)

(c) (d)

Figure 41: Volumetric mappings under different source/collocation point configurations. (a) shows the boundary constraint error (*C-Error*) under different *R-Ratio* when source points are placed on spheres. (b) and (c) plot the constraint error (b) and harmonic energy (c) respectively under different *O-Distance* when source points are placed on offset surfaces. In (d), constraint error under different numbers of source points and collocation points are compared. The x-axis is the *sRatio*. The y-axis shows the constraint error. Different curves show the cases under different *cRatio*.

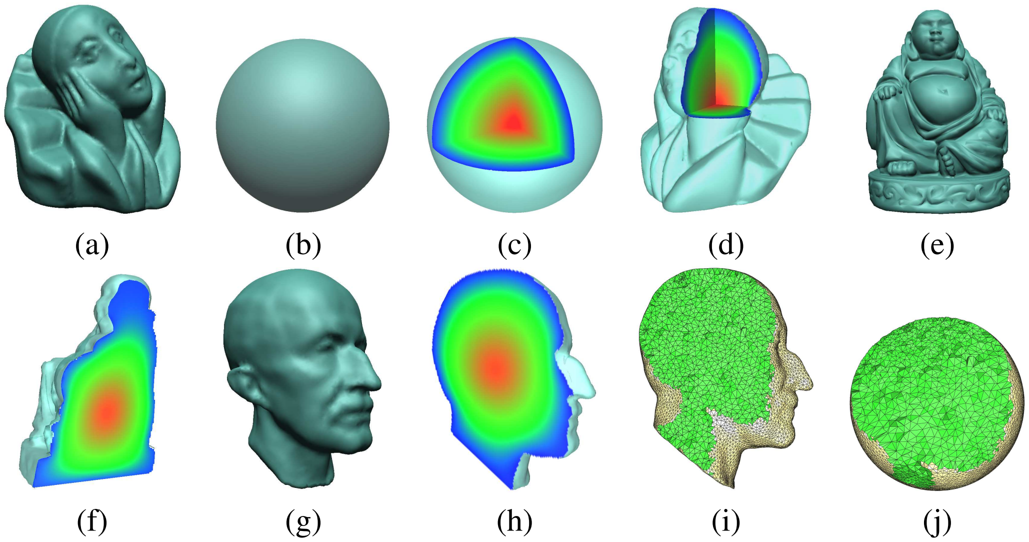


Figure 42: Volumetric mappings to the canonical sphere. The Pierrot model (a) is mapped to the solid sphere (b); (c) shows the color-coded distance field in the sphere. (d) visualizes the volumetric mapping: each point p in the original model of (a) is mapped to a point q inside the solid sphere; the target position q's color (as shown in (c)) is transferred and depicted on the corresponding p position (as shown in (d)). Similar examples of mapping from genus-0 Buddha model (e) and the Max-Planck model (g) to the sphere (b) are visualized in (f) and (h) by this same *color-coded distance field transfer* method. The tetrahedral mesh on the Max-Planck model (i) is mapped onto the solid sphere (j). Their corresponding cross-sections are visualized.

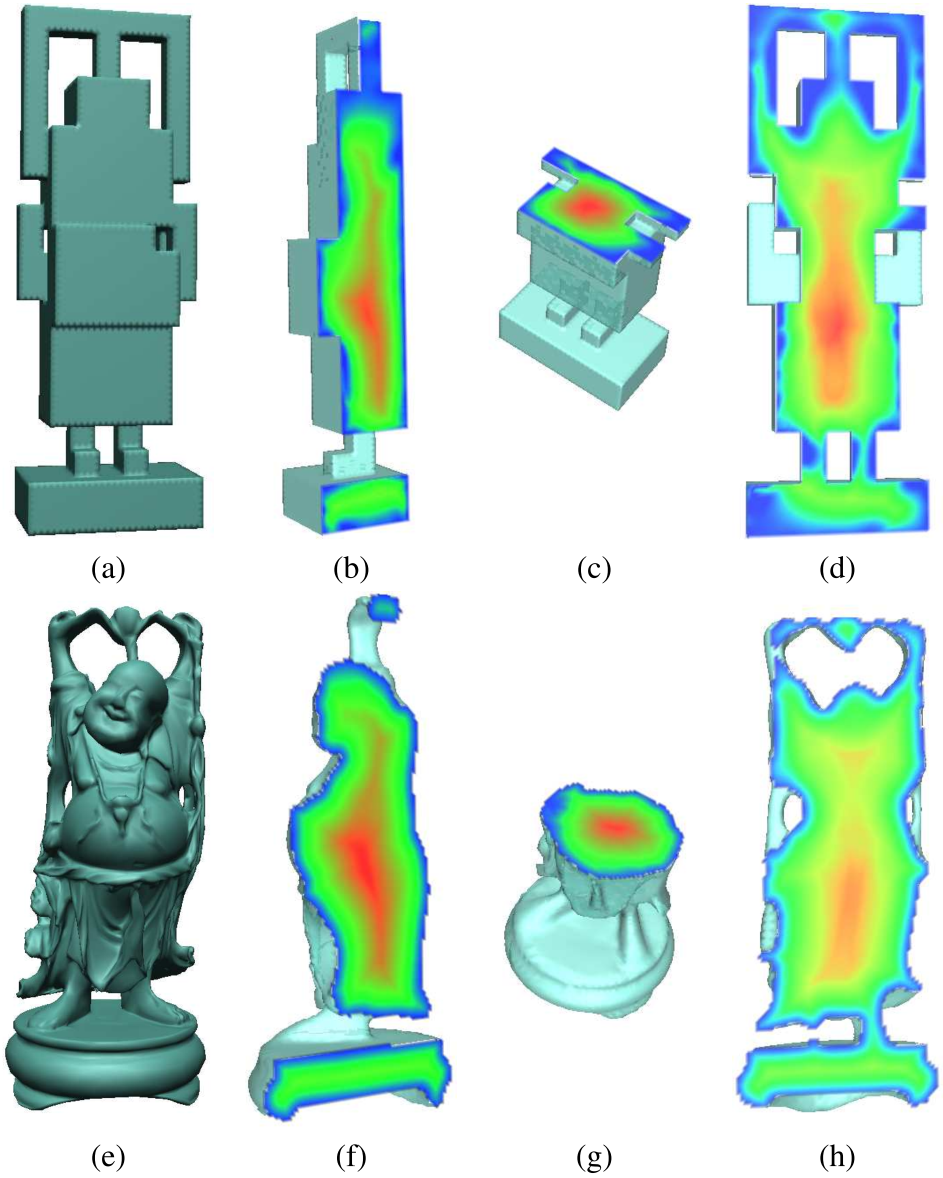


Figure 43: Harmonic volumetric mapping from a solid polycube model (a) to the solid Buddha model (e). (f), (g) and (h) show the color-coded distance field of the Buddha, from three cross-sections respectively. This color-coded distance field is transferred from the Buddha to the PolyCube model as shown in (b), (c) and (d) correspondingly.

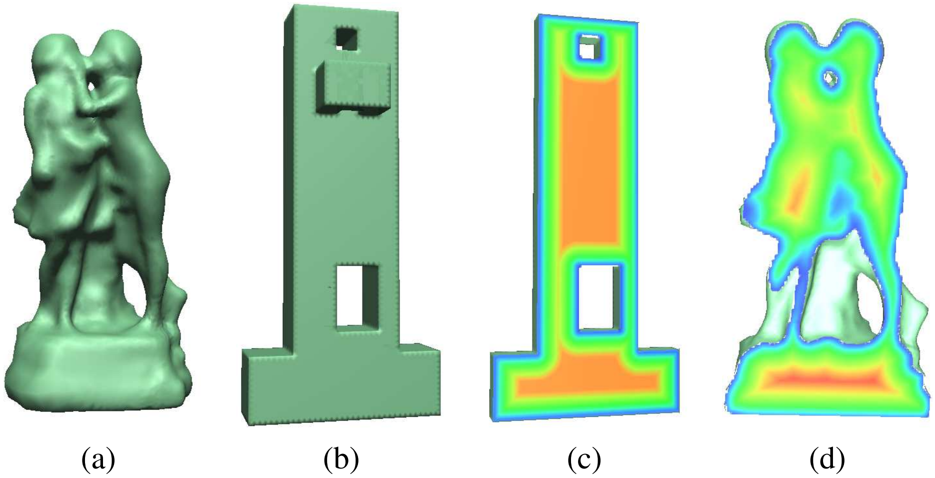


Figure 44: Harmonic volumetric mapping from the solid Sculpture model (a) to a solid polycube model (b). (c) color-codes the distance field of the polycube. In (d), the transferred color-coded distance field is visualized on the Sculpture model.

Chapter 7

Surface Mapping Applications

7.1 Texture Transfer

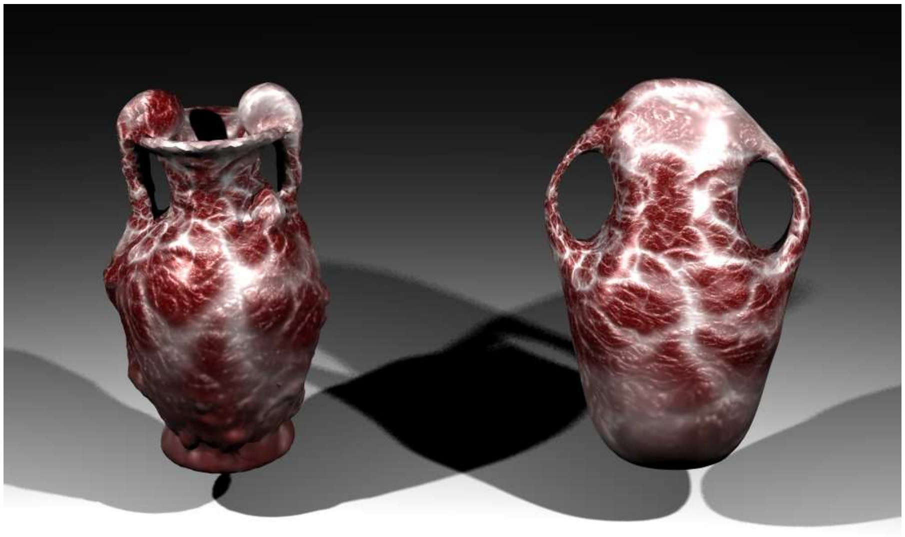


Figure 45: Texture Transfer using the Global Optimized Surface Mapping.

As we shown in 5.7, we use the texture transferring method to effectively visualize the mapping behavior and its distortion. Such texture as well as material transfer is straightforward as an application of our mapping. We show an example in Figure 45, which transfers the texture from the amphora model to the vase model. Since our globally optimized mapping has minimized distortions and global smoothness, such a transplant is physically natural, which potentially provides a powerful tool for reusing or transferring information such as material properties, BRDF, etc. between models towards information integration.

7.2 Shape Morphing

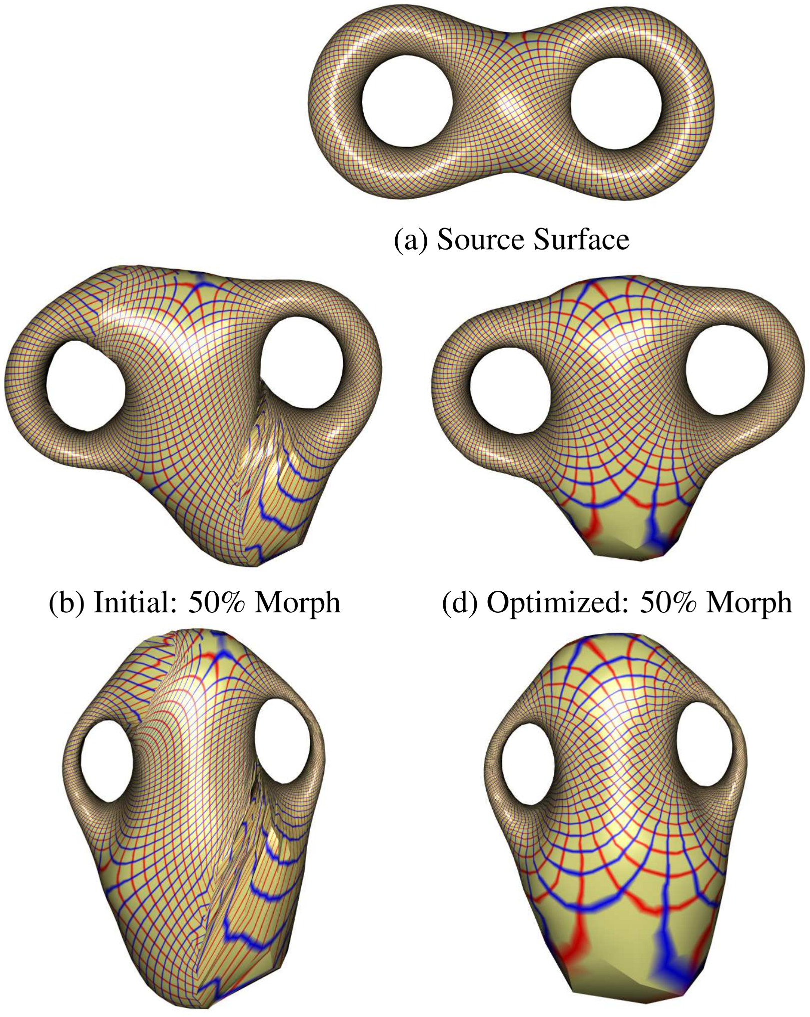
Morphing, as an application had been addressed and shown in section 4.6. It is also another intuitive way to visualize mapping and to evaluate its distortion since the behavior of the morph can provide an intuitive visual judgement of the mapping quality. Mapping with lower distortion usually leads to better morphing effect. Figure 46 shows an example. The initial mapping is created by the technique of [12]. Based on the initial map, we can conduct linear interpolation and generate the morph as shown in the left column. The generated sequence is obviously not attractive. We then optimize the surface map, and regenerate the morph. As shown in the right column, the new morph sequence demonstrates symmetric deformation and is visually much more smooth and pleasing.

Rigorously speaking, the morphing sequence generated by mapping with lower distortions means that the deformation sequence is closer to the 'geodesic' in the space of shapes, minimizes unnecessary distortion during the interpolation of shapes, and thus provides better visualization results.

In graphics applications, shape morphing is widely studied as a direct application for surface mapping. Users usually want to have control of the morphing via feature or constraint points. To achieve this goal, as we previously showed in Section 4.6 and indicated in Section 5.6.3, we can use the existing techniques for feature alignments during the initial mapping process; then we should keep this correspondence during the optimization afterward.

7.3 Cross Surface Parameterization

This chapter introduce our application of surface mapping in cross surface parameterization, or more specifically, in generating canonical mappings from surfaces to simplified domains. Our method conveniently creates canonical mappings from arbitrary surfaces to simplified domains with globally optimized distortions.



(a) Source Surface

(b) Initial: 50% Morph

(d) Optimized: 50% Morph

(c) Initial: Target

(e) Optimized: Target

Figure 46: Optimized Mapping for more Natural Morph. The source surface is shown in (a). If the initial map is used, the Morph generated is depicted in the left column: (b) shows the 50% morph, (c) shows the map on the target surface. When the surface map is optimized using our algorithm, the result is shown in the right column ((d) and (e)).

The canonical domain can be polycubes [120], so that graphics processing such as parameterization with lower distortion, polycube spline generation, etc. can be applied based on our mapping. The domain can also be some canonical *N*-hole tori [36], so that topologically equivalent shapes can be processed or analyzed on this smooth common domain.

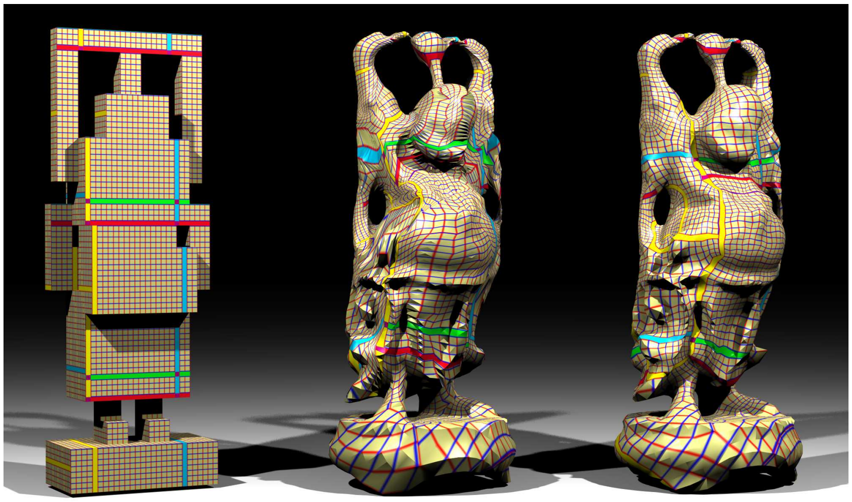


Figure 47: Mapping from a Polycube to the Happy Buddha Model. From left to right, we visualize the texture on the Polycube, the transferred texture on the Buddha by the initial mapping, and the transferred texture by the final mapping.

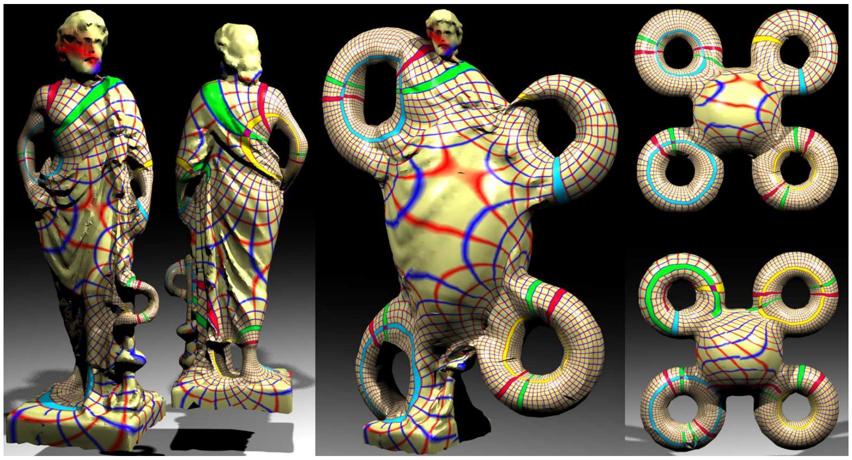


Figure 48: Optimized Surface Mapping from the Greek Model to the 4-Torus. The left column shows the front and back of the Greek model with its texture; the right column shows the front and back of the target surface (4-Torus), respectively, with texture transferred by our mapping; the middle column shows the 50% morph from the Greek to the 4-Torus under our mapping.

In Figure 47, we visualize the polycube map for the genus-6 Buddha model. Our method successfully deforms arbitrarily built initial maps with severe distortions to a global optimum. In Figure 48, we show our mapping from the genus-4

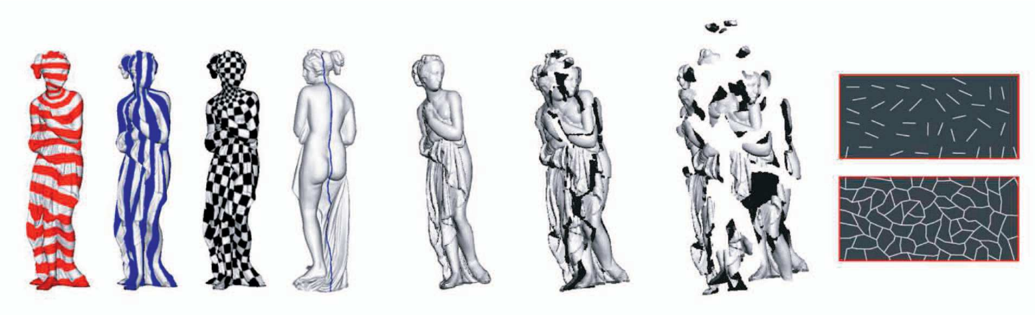


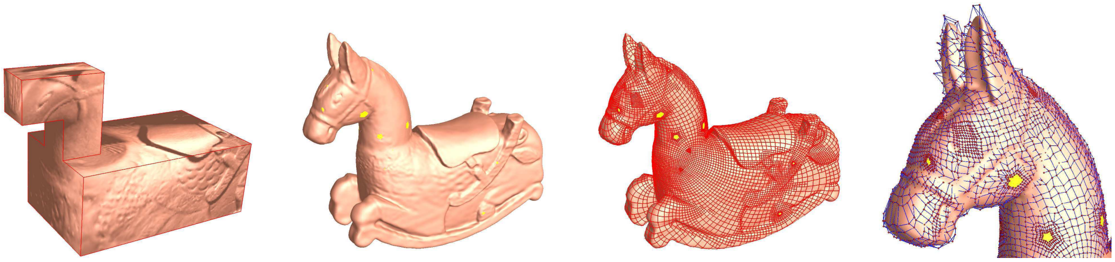
Figure 49: Conformal surface mapping and meshless simulation of the explosion of Iphigenie.

Greek model to a canonical 4-torus.

Our method has an important advantage over direct projection methods for computing polycube map such as [120] in that our method is intrinsic. Therefore it is more robust, invariant with models' spatial positions and sizes. Furthermore, when shapes are with complicated topology and geometry, or the source surface is greatly different than the target surface (for example, Greek and Torus as shown above), direct projection method is highly error-prone, but our method can robustly handle it.

7.3.1 Physically Based Simulation based on Point Cloud Mapping

Surface mapping also benefits physically based simulation. In [47], we conformally map point cloud surface data onto some regular planar domain, and conduct thin-shell simulation such as physically based deformation and cracking on the domain directly. Partial differential equations can be solved more efficiently on the planar domain. Figure 49 shows an example, we conformally map the Iphigenie surface model onto a planar rectangle. The iso-u, iso-v, and the u-v planar parameter are color-coded and visualized in the left three figures, respectively. The fourth figure shows the traced integral curve connecting two boundaries in the parameterization step. The right most figure shows the crack pattern of the parametric domain in the first (top) and final (bottom) step of the simulation. Further technical details for this work is beyond the scope of this dissertation, the author recommends the



(a) Polycube map (b) Polycube T-spline (c) T-junctions on polycube spline (d) Close-up of control points

Figure 50: Polycube spline for the Isidore Horse model.

interested readers to our paper [47] for more deduction and implementation issues.

7.3.2 Spline Construction based on Polycube Maps

A special canonical domain shape is called polycube. Mapping a surface with complicated geometry onto polycubes was first addressed by Tarini et al. in [120]. It has many applications in geometric processing and modeling research. With our globally optimized surface mapping, polycube mapping with lower distortion than [120] can be computed, and we use the polycube as a domain to construct T-spline. Such a least distorted polycube mapping leads to a precise "one-piece" shape representation without any cutting and gluing. Figure 50 shows an example of the polycube spline for the Isidore Horse model. The conformal polycube map, visualized in (a), serves as the parametric domain for the spline. The T-splines obtained via affine structure induced by the polycube map are shown in (b) and (c). The red curves on the spline surface in (c) highlight the T-junctions. The close-up of the spline model overlaid with the control points is illustrated in (d). This polycube T-spline contains 12158 control points. The original model contains 150K vertices. The root-mean-square error is 0.07% of the diagonal of the model. Technical details for polycube map and spline will not be addressed in this document, but can be found in our paper [127].

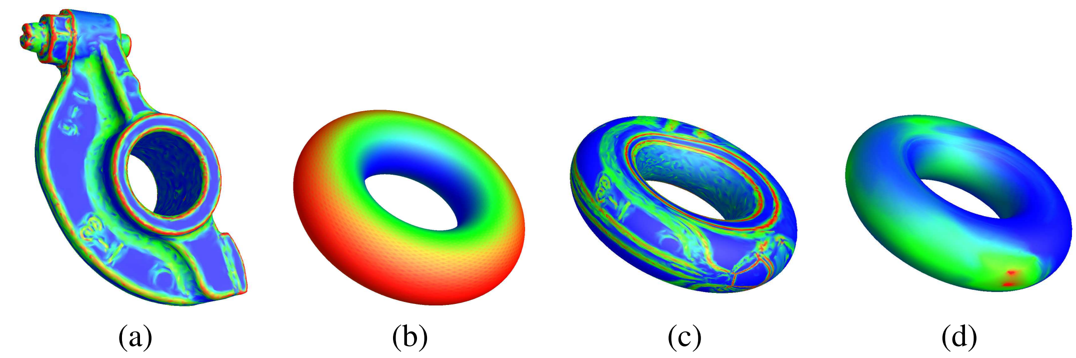
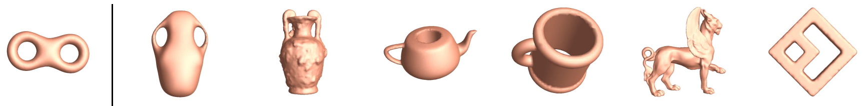


Figure 51: Surface Registration for Matching (Torus vs Rocker Arm model). (a) Mean curvature distributions of Rocker Arm (red represents the maximum while blue represents the minimum); (2) Mean curvature distributions of Torus; (c) Mean curvature difference distributions, visualized on Torus; (d) Conformal stretching factor, visualized on Torus.



1	2	3	4	5	6	7
2	0	3.59	22.72	20.81	59.70	19.43
3		0	21.99	21.29	59.38	18.72
4			0	10.98	39.66	19.65
5				0	44.45	16.91
6					0	32.89
7						0

Figure 52: Shape Comparison using Conformal Representation. The first rows show all shapes to be compared. The second row and the first left column are their indices. The table has the symmetry property, and the numbers measure the distance between models in a pairwise manner.

7.4 Shape Matching and Comparison

Our optimal surface mapping creates global, low angular-distortion correspondence between two models. With such a non-rigid registration, we can easily match two shapes and clearly visualize their difference distributions for potential subsequent analysis purposes.

Conformal Representation. A natural way to characterize the matching between two surfaces is called *conformal representation* [41]. According to [41], when a surface is mapped onto a target surface, if the resulting conformal representation is fixed, the original source surface is rigidly determined. The conformal representation contains two terms: mean curvature H, and *conformal factor* λ. The conformal factor λ of a point p under a mapping f represents the local area change, i.e., the stretching of the map. Discretely, if we denote the area of the one ring neighbor of p as $A(p)$, and the area of one ring neighbor of $f(p)$ on the target surface as $A(f(p))$. $\lambda(p)$ can be approximated by the ratio of $A(f(p))$ over $A(p)$. In our work, although our surface map is not fully angular-distortion-free (according to Riemannian geometry, between most high genus models, these kinds of conformal maps do not exist), our global optimization aims to best relieve angle distortions. Thus the (H,λ) defined on our map is a well approximated and meaningful representation.

Shape Matching. In Figure 51, we visualize our surface matching between a torus and a Rocker Arm model using the above conformal representation. (a) and (b) color-code the mean curvature distributions of Rocker Arm and Torus, respectively. We color-code the mean curvature difference in (c) and the stretching factor distribution in (d). The color-coding of two terms of conformal representation shows us where and how much the two surfaces are intrinsically different in a visually meaningful way. Since the globally integrated matching energy is smaller when the mapping is with lower stretching/distortions, our optimized surface mapping provides a great registration for the above mechanism. On the other hand, the registration by our mapping, with global smoothness and low distortion properties, can be used as a preprocessing step for various other matching techniques. It serves as a general shape registration and visualization tool.

Shape Comparison and Retrieval. Given many shapes in a database, we can match and compare them via canonical domains. This provides an efficient and geometrically meaningful way to measure their differences. Here we perform an experiment on a database containing 6 different genus-2 geometric shapes: Vase, Amphora, Teapot, Cup, Feline, and Cube. We use a two-hole torus as the canonical domain for all these genus two surfaces. We first compute mappings between these surfaces and the 2-torus domain, and then in a pairwise fashion compare these surfaces via the domain using the matching energy defined by the conformal representation:

$$E(S_1,S_2) = \int_{p\in T} ||\lambda_1(p) - \lambda_2(p)||^2 + \beta||H_1(p) - H_2(p)||^2 dp,$$

where S_1 and S_2 are two shapes being compared, T is the canonical torus domain, λ is the conformal factor, and H is the mean curvature. In Figure 52, we can see the models in the first row. The matching energies, used as their distance, are shown in the table. Since the symmetry of the distance is obviously preserved, we only show the upper-right part of the table.

Chapter 8

Volumetric Mapping Applications

8.1 Information Transfer

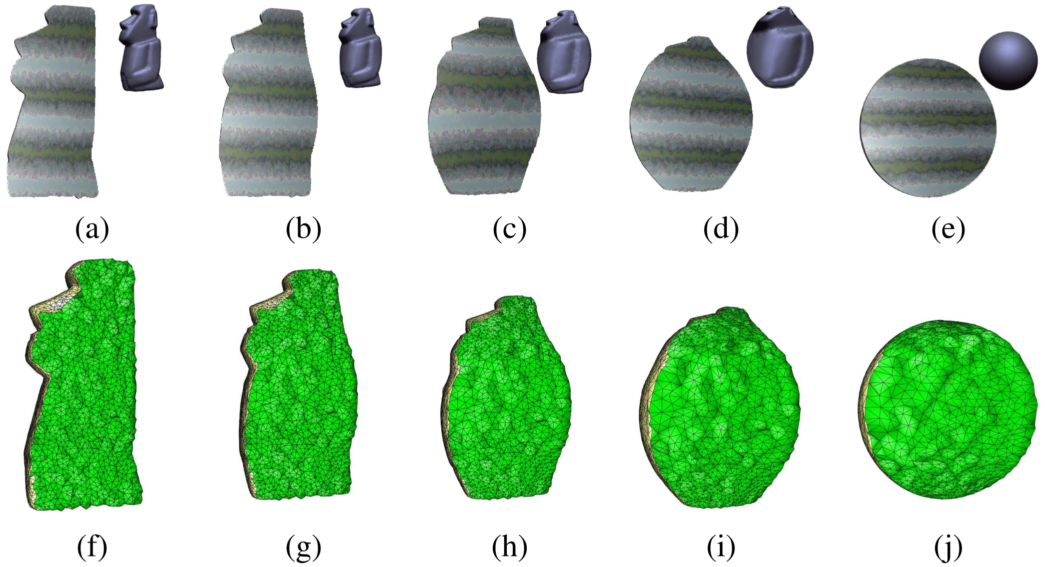


Figure 53: Volumetric harmonic mapping for information reuse. The material on the solid Moai model is preserved when it deforms during the animation ((a)-(e)). (f)-(j) show this consistency (via one corresponding cross-section) on the original tetrahedral mesh and the mapped meshes during its morphing procedure.

Once the correspondence between two volume models has been established, we can easily transfer information from one object to the other. The previous *Color-coded Distance Field Transfer* method already demonstrates this. The information being transferred can be all kinds of volumetric functions, and not just limited to color; it can be material, solid texture, density, and even more complicated ones such as gradient or strain/stress tensors. Therefore, it has potential applications in a larger scope. Figure 53 shows another example. When the Moai model deforms, the material information on the original model is transferred and preserved by the deformed model during the deformation. In the second row, we also show the corresponding tetrahedral mesh from the same corresponding cross-section to visualize this transfer. We believe harmonic volumetric mapping will provide automatic interior region registration on real temporal or deforming data in the future.

8.2 Shape Matching and Analysis

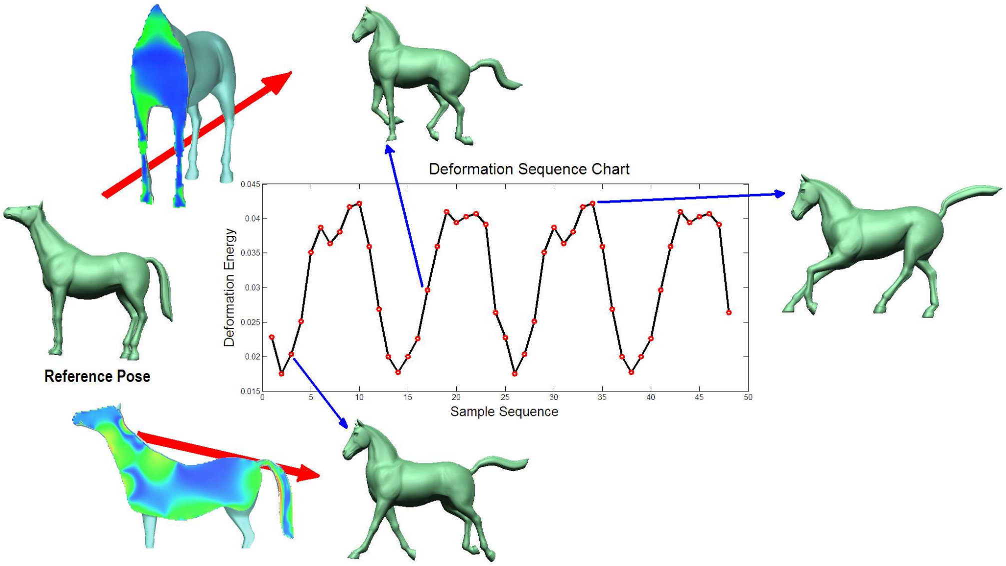


Figure 54: Energy analysis of deformation sequences. The horse model is deformed in a sequence. The deformation energies are calculated (red circles). The distribution of the deformation energy required for each model in the sequence can be illustrated on the reference model.

A direct application of mapping is registration. Based on a good registration, we can easily measure the difference between two objects in a quantitative way and perform both qualitative and quantitative analysis. We use an example to demonstrate the usage of volumetric mappings on shape matching and analysis. In this experiment, we analyze a horse-gallop deformation sequence. We use the vertex correspondence provided in the deformation sequence as the boundary surface mapping. Then we compute the volumetric harmonic mappings from the referenced, static-standing horse model to all the deformed poses in a sequence. With the mapping, we can compute their deformation energies. This energy naturally measures the distance from the deformed shape to the reference model. Since we have mappings between objects, we can get not only a numerical distance value, but also the precise error distribution between two shapes. This distribution can be clearly visualized under this deformation, where stretching and bending concentrate over the shape. Note that, as we discussed in Section 6.3.6, this procedure, with the computation of a large number of volumetric mappings, is performed efficiently by reusing one decomposition result.

The deformation energies of the horse gallop sequence are shown in Figure 54. We can easily see from the energy chart that there are four running cycles in the data-set of the deformation sequence. And with the deformation energy, we naturally measure how different each model is from the reference model. The distributions of the deformation energy between the reference model and the deformed model are color-coded and illustrated. Given a sampled model in the deformation sequence, which regions have high deformation energy concentration can be clearly visualized from the color-coded distribution of the deformation energy, as we depicted on the original model with cross-sections. Without this correspondence, this kind of visualization and analysis is impossible.

8.3 Tetrahedral Remeshing

Regular tetrahedral mesh structure is highly desirable for finite element analysis and physically-based deformations or simulations. This is because regular tetrahedralization provides great precision and efficiency for geometry processing and physically-based computation [121]. With our volumetric mapping, we can easily transfer the tetrahedralization of an object to another object. We call this *tetrahedral remeshing*. As shown in Figure 55, we use the regular tetrahedral mesh of a solid

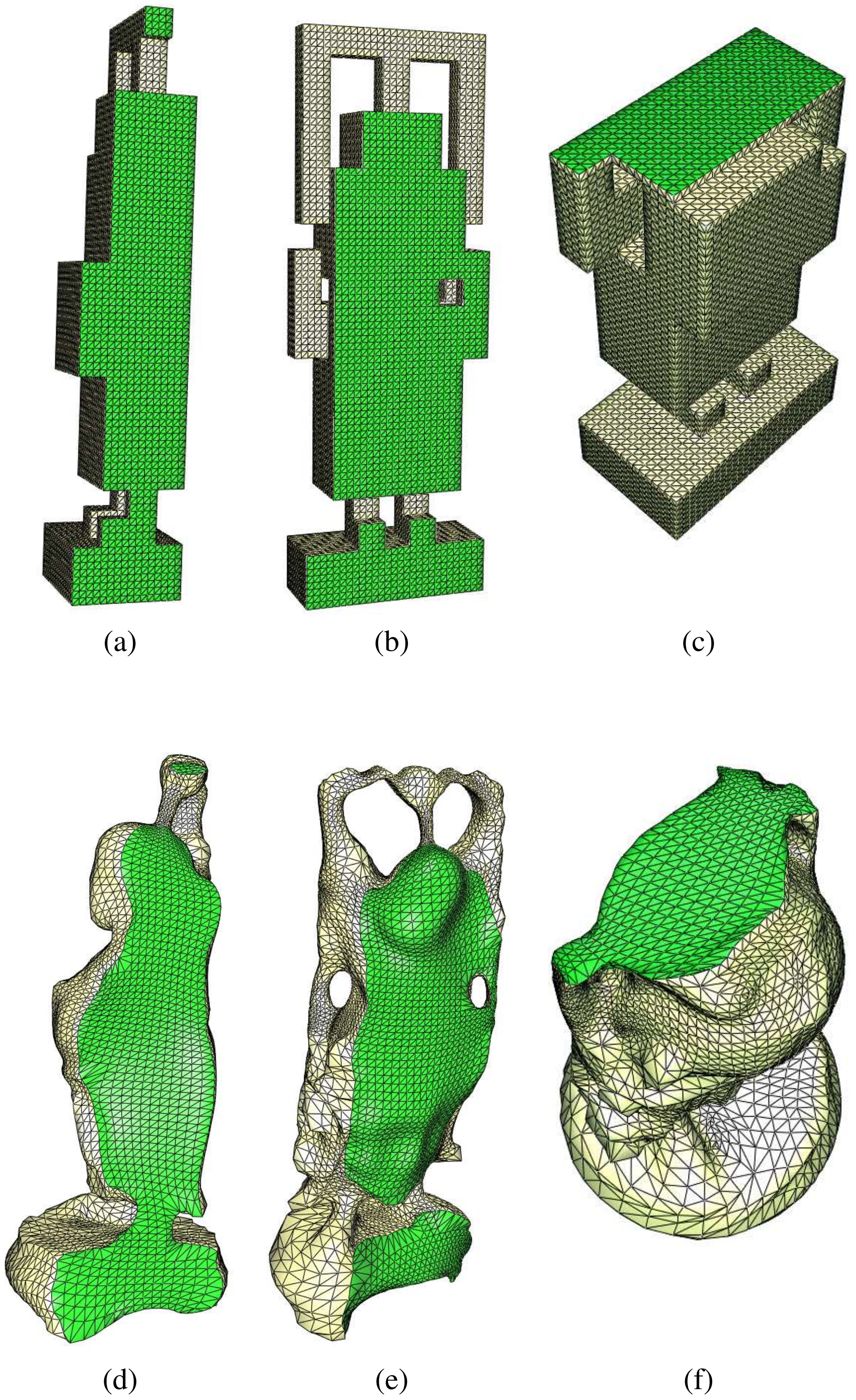


Figure 55: Harmonic Volumetric Mapping for Tetrahedral Remeshing. (a), (b) and (c) show the tetrahedral mesh of the polycube model in Figure 43(a) from three cross-sections. It is utilized to *remesh* the solid Buddha model (Figure 43(e)); and the results are visualized with corresponding cross-sections in (d), (e) and (f).

polycube model (Figure 43(a)) to remesh the solid Buddha model (Figure 43(e)). (a), (b) and (c) show the tetrahedral mesh on the polycube from three cross-sections; (d), (e) and (f) show the remeshed solid Buddha model. Tetrahedralization for regular shapes like polycubes can be easily created as shown in this example. So using our mapping, we can generate regular tetrahedral structure for complicated objects.

8.4 Volume Texture Synthesis

We can also synthesize volumetric texture using our method. As shown in Figure 56, given a 2D texture image, we get the surface texture mapping, then the texture applied on the surface can be smoothly propagated to the interior regions of solid objects. To synthesize the interior texture, we only need to make a change on the boundary condition; instead of using the target boundary points positions, we use the texture (u, v) coordinates. Figure 56(a) shows an solid Igea model; and we map a $2D$ image texture onto its surface as shown in (c). This texture is smoothly extrapolated into the interior region using our method. (e) illustrates the synthesized solid texture. (b), (d) and (f) show another example on the Pensatore model. From the given 2D image, we can synthesize the volumetric texture to decorate the solid interior for graphics applications.

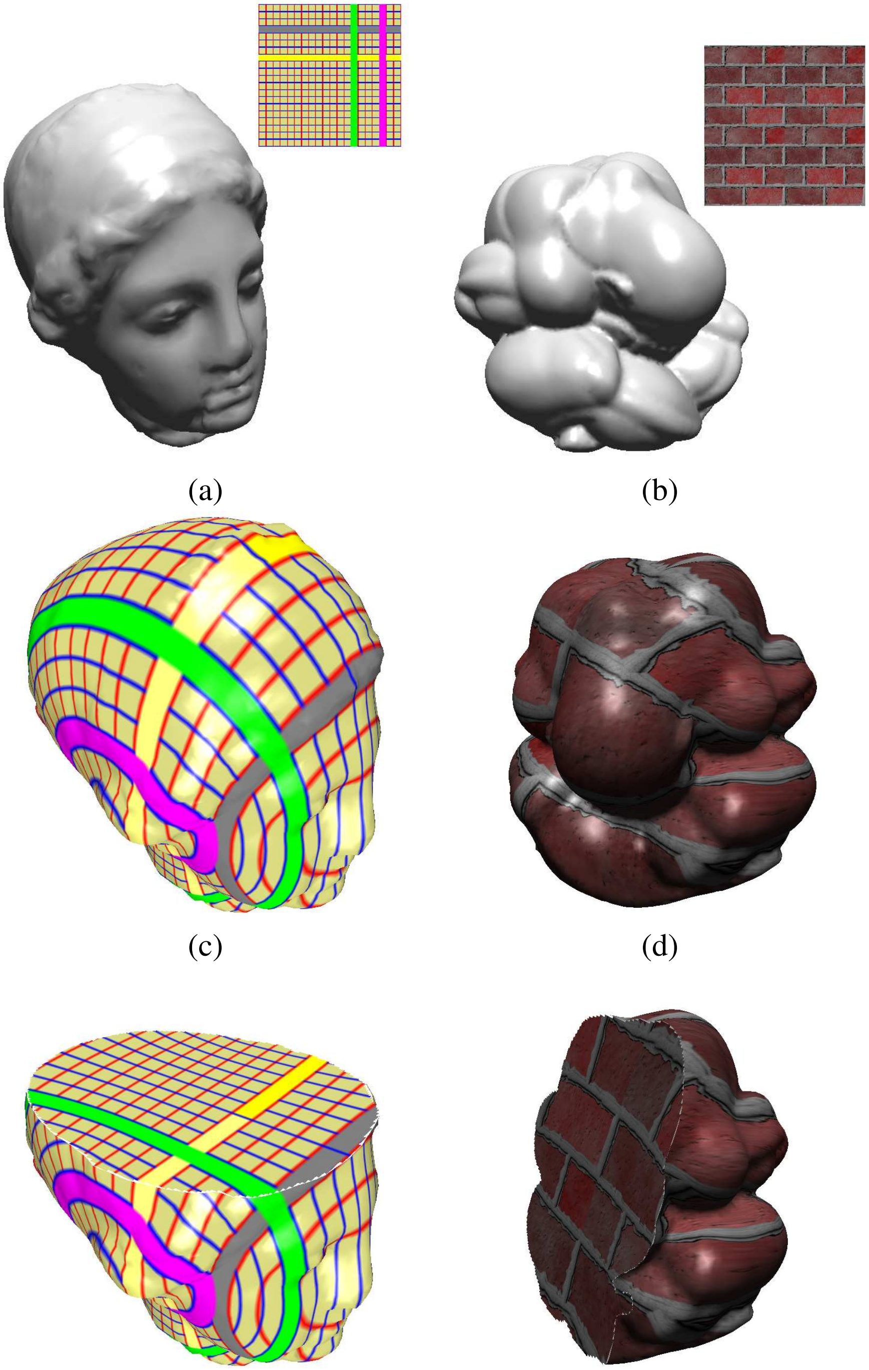


Figure 56: Solid texture synthesis. In the first column, the solid Igea model and a corresponding $2D$ image texture are shown in (a). The surface texture is firstly mapped to the Igea as illustrated in (c). We synthesize the interior solid texture and illustrate a cross-section view in (e). Similarly, in the second column, the solid Pensatore model and its $2D$ texture are shown in (b), the surface texture mapping and the synthesized solid textures are visualized in (d) and (f).

Chapter 9

Conclusion and Future Work

In this dissertation, we have document our recent research results, ongoing research and future research directions of our shape matching framework. Our framework contributes to the study on **curves comparison, surface matching, and volume data mapping**. Theoretically, it provides fundamental progress in understanding, analysis, and computational algorithms on shape mapping problems. From the perspectives of its applications, we also demonstrate the great potential of shape mapping by utilizing our mapping result into many valuable applications.

9.1 Contribution Summary

The contributions of our shape mapping framework are summarized in this section.

By studying curves on surfaces:

1.1 We design an elegant signature to describe curves embedded on surfaces. This signature provides a classification of all simple closed curves, measuring how they segment the surface.

1.2 We apply the signature to surface segmentation, and use it for segmentation matching and segmentation transfer.

1.3 Based on this, we present a semi-automatic surface matching framework, which is currently applied on genus zero surfaces but we believe can be generalized to surfaces with higher genus.

This above framework is not fully automatic and seems inadequate for surface mapping purpose. By further studying surface segmentation:

2.1 We present a surface decomposition scheme, called canonical pants decomposition, applicable for surfaces with arbitrary topology types. With our decomposition method, different surfaces with complicated but same topology can be decomposed into consistent sets of sub-regions for further piecewise mapping, with great automation.

2.2 We design a surface mapping framework, based on the canonical decomposition scheme, that can flexibly handle the topological surgery (mapping between surfaces with different topologies) and feature alignment, through the least and intuitive user involvement.

Piecewise surface mapping has several advantages (intuition, efficiency, and feature alignment), but it lacks global smoothness and global distortion control. Therefore, globally optimal surface mapping is worthy of research, and

3.1 We present a rigorous global computation algorithm that computes the optimal mapping between surfaces with nontrivial topology.

3.2 We prove both theoretically and experimentally the global optimality and uniqueness of our mapping result as well as the global convergence of our algorithm.

By generalizing our surface mapping computation to volumetric data:

4.1 We introduce a meshless method (called method of fundamental solution) into computer graphics community. The method is simple and robust, and greatly improves the efficiency of the volumetric mapping computation.

4.2 We demonstrate that our method works for solid objects in $\mathbb{R}^3$ with boundary surfaces of arbitrary topology types, and can be generalized to data with feature points or layer alignments.

9.2 Limitations of Our Work and Future Improvement

There are also limitations of our current methods, and we plan to work on them in the future.

1. The signatures for curves on surfaces currently only work on genus-0 surfaces, however, we believe it can be generalized to higher genus cases, and we will pursue that in the future.
2. Our global optimization of surface mapping currently relies on gradient descent iterations, which takes a long time to converge to the global optimal. In the future, we will work on other more efficient iterative numerical methods, or use GPU or parallel computing techniques to improve its efficiency.
3. Our harmonic volumetric mapping results rely on the boundary surface mappings, and we have not yet integrated the feature points/layer alignment. In the future, we will studied how the boundary surface map affects the volumetric mapping result and try to compute the harmonic volumetric mapping with the best harmonicity. We will also integrate feature alignment for different application requirements.
4. Our current shape mapping framework only considers purely geometric information. Taking into account other information such as material, domain knowledge etc is meaningful. In the future, we will explore along this direction, especially for volumetric mapping.

9.3 Future Work

Besides the above directions that aims to improve our current computation algorithms, there are many more immediate and valuable research topics based on our current framework. We plan to explore along two directions. One is to find more applications of our shape mapping framework; while another is to further explore shape mapping computation methodologies.

9.3.1 Exploring Mapping Applications

As we have seen in Chapter 7 and Chapter 8, shape mapping has already shown its great potential in many graphics and modeling applications. We can easily see many immediate and direct applications on shape mapping.

Medical Data Registration and Data Fusion. We would like to utilize mapping as a registration tool with real medical data for clinical analysis. In our previous work [51], we use spherical mapping to register the deforming brain cortex surfaces, and visualize the displacement of the evolving cortex surface. Since now we have more advanced techniques for mapping surfaces with complicated topology, and techniques for mapping volumetric data. We would like to generalize the medical data set to others such as lungs or hearts (could be either surface or volumetric data).

Virtual Surgical Simulation. Virtual surgical simulation can be very useful for surgeon training and surgery rehearsal. Physically based simulation usually provides real effects but costs lots of computational efforts. As our previous work [47], this simulation can be converted onto some regular domain with mapping (parameterization) techniques, for great efficiency improvement. We plan to follow the same philosophy, and compute the surgical simulation on canonical domains. In a long run, this could be further combined with haptic environments and technologies to lead to a virtual reality system aiming for surgeon training.

Deformation Transfer. An interesting application in the modeling field is to effectively reuse the deforming sequence [114]. This deformation sequence of a given model can be obtained by motion capture devices, many research works have been conducted on effectively transplanting this sequence onto another object. With volumetric mapping, we can transplant the interior deformation gradient tensors, and integrate them to compute the deformed sequence on the second object.

Shape Retrieval. The number of virtual digital models is exploding nowadays on the internet, and shape retrieval is becoming a hot topic in graphics and modeling fields. With the shape mapping as registration, objects can be compared and retrieved from database. In order to utilize our mapping algorithm for real retrieval applications, we may need to integrate learning techniques for possible semantics effects. Also, to improve the registration, comparison, and retrieval efficiency, we plan to find/generate regular shapes to classify shapes in a database into groups, and further apply some clustering algorithms.

Geometric Data Compression. Techniques such as geometry image [37, 54] can be used for mesh compression. Via cross surface parameterization, we can also re-sample (and get rid of mesh connectivity) surfaces on canonical domains without cutting. Without tearing surfaces apart, compression and reconstruction could be more effective.

9.3.2 Improving Mapping Computation Methodologies

In terms of computation methodology, I would also like to investigate both theory and algorithmic aspects. I would like to further improve the effectiveness and efficiency of our mapping computation.

Volumetric Pants Decomposition. The idea of pants decomposition is also generalizable to volumetric data in the sense of solid "pants". Decomposing volumetric data [15] help simplify the complexity of the computation; also, we should be able to better control the local behavior of volumetric mapping, and improve the efficiency and accuracy of the computation.

Free-boundary Harmonic Volumetric Mapping. The harmonic volumetric mapping is determined by its boundary condition. Whether the globally optimized surface mapping leads to the volumetric mapping with smallest harmonic energy is not clear (and seems unnecessary). We would like to study the relationship between the volumetric map and its boundary map, and try to compute the volumetric map with best harmonicity, which leads to the most natural deformation energy computation between two objects.

Equiareal Mapping. In medical imaging, surface mapping with preserved area distortion is useful for clinical analysis. However, a pure area-preserving map is hard to control and lacks mature computation techniques because it has infinite dimensions. We are working towards this goal by mapping surfaces with the combination of angle and area preservation.

Shape Mapping Relied on More Than Geometry. Our current mapping framework only considers pure geometry. It may be necessary to integrate more information such as domain knowledge, or other techniques such as machine learning. We will explore along this direction as well.

9.4 Concluding Remarks

These directions for future work, and the many other open problems that exist, are sure to encourage interesting and exciting research in shape mapping for years to come. As technical difficulties are overcome, and existing computational algorithms are improved, the applications of shape mapping will increase in variety and number. We are pleased to have taken the first step in uncovering the heretofore untapped potential of shape mapping by presenting our framework to the graphics and visual computing. It is our hope that this integrated approach and demonstrated mapping applications will foster continued interest and research in this area. We look forward to the continued exploration of shape mapping and predict a successful future for it.

Bibliography

[1] M. Alexa. Merging polyhedral shapes with scattered features. In *Proc. of the Intl Conf. on Shape Modeling and Applications*, pages 202–210, 1999.

[2] B. Allen, B. Curless, and Z. Popović. The space of human body shapes: reconstruction and parameterization from range scans. In *SIGGRAPH '03*, pages 587–594, New York, NY, USA, 2003. ACM Press.

[3] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. 3D shape histograms for similarity search and classification in spatial databases. In *SSD '99: Proc. 6th Intl Symp. on Advances in Spatial Databases*, pages 207–226, London, UK, 1999. Springer-Verlag.

[4] A. Asirvatham, E. Praun, and H. Hoppe. Consistent spherical parameterization. In *International Conference on Computational Science (2)*, pages 265–272, 2005.

[5] P. Banerjee. *The Boundary Element Methods in Engineering*. McGraw-Hill, New York, 1994.

[6] R. Basri, L. Costa, D. Geiger, and D. Jacobs. Determining the similarity of deformable shapes. *Vision Research*, 38:2365–2385, 1998.

[7] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS, Proc. Neural information processing systems conf.*, pages 831–837, 2000.

[8] T. Belytschko, Y. Y. Lu, and L. Gu. Element free Galerkin methods. *International Journal for Numerical Methods in Engineering*, 37(2):229–256, 2005.

[9] J. Bloomenthal. An implicit surface polygonizer. *Graphics Gems IV*, pages 324–349, 1994.

[10] A. I. Bobenko and B. A. Springborn. Variational principles for circle patterns and Koebe's theorem. *Trans. Amer. Math. Soc.*, 356:659–689, 2004.

[11] A. Bogomolny. Fundamental solutions method for elliptic boundary value problems. *SIAM Journal on Numerical Analysis*, 22:644–669, 1985.

[12] C. Carner, M. Jin, X. Gu, and H. Qin. Topology-driven surface mappings with robust feature alignment. In *IEEE Visualization*, pages 543–550, 2005.

[13] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *J. Comput. Phys.*, 155(2):468–498, 1999.

[14] B. Chow and F. Luo. Combinatorial Ricci flows on surfaces. *J. Differ. Geom.*, 63(1):97–129, 2003.

[15] J. Chuang, N. Ahuja, C. Lin, C. Tsai, and C. Chen. A potential-based generalized cylinder representation. *Computers and Graphics*, 28:907–918, 2004.

[16] I. Cohen, N. Ayache, and P. Sulger. Tracking points on deformable objects using curvature information. In *ECCV*, pages 458–466, 1992.

[17] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, 2001.

[18] D. DeCarlo and J. Gallier. Topological evolution of surfaces. In *Proc. Graphics interface*, pages 194–203, 1996.

[19] M. Desbrun and M. P. Cani. Animating soft substances with implicit surfaces. In *Proc. SIGGRAPH '95*, pages 287–290, 1995.

[20] M. Desbrun and M. P. Cani. Smoothed particles: A new paradigm for animating highly deformable bodies. In *Proc. EG Workshop on Animation and Simulation*, pages 61–76, 1996.

[21] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes. *Comput. Graph. Forum*, 21(3):209–218, 2002.

[22] T. K. Dey, K. Li, and J. Sun. On computing handle and tunnel loops. In *International Conf. on Cyberworlds*, pages 357–366, 2007.

[23] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Comput. Aided Geom. Des.*, 22(5):392–423, 2005.

[24] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH*, pages 173–182, 1995.

[25] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *ACM-SIAM Symp. on Discrete algorithms*, pages 1038–1046, 2005.

[26] G. Fairweather and A. Karageorghis. The method of fundamental solution for elliptic boundary value problems. *Advances in Computational Mathematics*, 9(1-2):69–95, September 1998.

[27] M. Floater. Parametrization and smooth approximation of surface triangulations. *Computer Aided Geometric Design*, 14(3):231–250, 1997.

[28] M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–27, 2003.

[29] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, Mathematics and Visualization, pages 157–186. 2005.

[30] M. Frenkel and R. Basri. Curve matching using the fast marching method. In *EMMCVPR*, pages 35–51, 2003.

[31] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. Modeling by example. *ACM Trans. Graph.*, 23(3):652–663, 2004.

[32] P. Garabedian. *Partial Differential Equations*. American Mathematical Society, 1998.

[33] M. A. Golberg and C. S. Chen. The method of fundamental solutions for potential, helmholtz and diffusion problems. *Boundary integral methods: Numerical and mathematical aspects (A99-30801 07-64)*, 1:103–176, 1999.

[34] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3D meshes. *ACM Trans. Graph.*, 22(3):358–363, 2003.

[35] A. Gregory, A. State, M. Lin, D. Manocha, and M. Livingston. Feature-based surface decomposition for correspondence and morphing between polyhedra. In *Proc. of the Computer Animation*, pages 64–71, 1998.

[36] C. Grimm and J. Hughes. Parameterizing n-holed tori. In *Mathematics of Surfaces X*, pages 14–29, September 2003.

[37] X. Gu, S. J. Gortler, and H. Hoppe. Geometry images. *ACM Trans. Graph.*, 21(3):355–361, 2002.

[38] X. Gu, Y. He, M. Jin, F. Luo, H. Qin, and S.-T. Yau. Manifold splines with single extraordinary point. In *SPM '07: Proc. of ACM symp. on Solid and physical modeling*, pages 61–72, 2007.

[39] X. Gu, Y. He, and H. Qin. Manifold splines. In *SPM '05: Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 27–38, New York, NY, USA, 2005. ACM.

[40] X. Gu and B. Vemuri. Matching 3D shapes using 2D conformal representations. In *MICCAI (1)*, pages 771–780, 2004.

[41] X. Gu, Y. Wang, T. Chan, P. Thompson, and S. T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Trans. Med. Imaging*, 23(8):949–958, 2004.

[42] X. Gu, Y. Wang, and S. T. Yau. Volumetric harmonic map. *Communications in Information and Systems*, 3(3):191–202, 2003.

[43] X. Gu and S. Yau. Computing conformal structure of surfaces. *Communications in Information and Systems*, 2(2):121–146, 2002.

[44] X. Gu and S.-T. Yau. Global conformal surface parameterization. In *Proc. Symp. Geometry Processing*, pages 127–137, 2003.

[45] X. Gu and S.-T. Yau. Surface classification using conformal structures. In *ICCV*, pages 701–708, 2003.

[46] X. Gu and S. T. Yau. *Computational Conformal Geometry*. International Press, 2007.

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

[48] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Trans. Vis. Comput. Graph.*, 6(2):181–189, 2000.

[49] R. S. Hamilton. The ricci flow on surfaces. *Contemp. Math.*, 71:237–262, 1988.

[50] A. Hatcher, P. Lochak, and L. Schneps. On the Teichmüller tower of mapping class groups. *J. Reine Angew. Math.*, 521:1–24, 2000.

[51] Y. He, X. Li, X. Gu, and H. Qin. Brain image analysis using spherical splines. *Proc. EMMCVPR2005, Lecture notes in computer science*, 3757:633–644, 2005.

[52] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *SIGGRAPH*, pages 203–212, 2001.

[53] E. Hildreth. *The Measurement of Visual Motion*. MIT Press, Cambridge, 1984.

[54] H. Hoppe and E. Praun. Shape compression using spherical geometry images. *Mathematics and Visualization*, pages 27–46, 2005.

[55] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three-dimensional shape searching: state-of-the-art review and future trends. *Computer-Aided Design*, 37(5):509–530, 2005.

[56] D. L. James and D. K. Pai. Artdefo: accurate real time deformable objects. In *Proc. SIGGRAPH '99*, pages 65–72, 1999.

[57] M. Jin, F. Luo, and X. Gu. Computing surface hyperbolic structure and real projective structure. In *SPM '06: Proc. ACM symposium on Solid and physical modeling*, pages 105–116, 2006.

[58] R. L. Johnston and R. Mathon. The computation of electric dipole fields in conducting media. *International J. Numer. Methods. Engrg.*, 14(12):1739–1760, 1979.

[59] P. Joshi, M. Meyer, T. DeRose, B. Green, and T. Sanocki. Harmonic coordinates for character articulation. In *SIGGRAPH '07*, pages 71–81, 2007.

[60] J. Jost. *Compact Riemann Surfaces*. Springer-Verlag, 2002.

[61] T. Ju, S. Schaefer, and J. D. Warren. Mean value coordinates for closed triangular meshes. *SIGGRAPH*, 24(3):561–566, 2005.

[62] T. Kanai, H. Suzuki, and F. Kimura. Three-dimensional geometric metamorphosis based on harmonic maps. *The Visual Comput.*, 14(4):166–176, 1998.

[63] A. Karageorghis and G. Fairweather. The method of fundamental solutions for the numerical solution of the biharmonic equation. *J. Computer Physics*, 69:434 – 459, 1987.

[64] M. Kazhdan. *Shape Representations and Algorithms for 3D Model Retrieval*. PhD thesis, Princeton University, June 2004.

[65] M. Kazhdan, B. Chazelle, D. Dobkin, A. Finkelstein, and T. Funkhouser. A reflective symmetry descriptor. In *ECCV (2)*, pages 642–656, 2002.

[66] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz. A reflective symmetry descriptor for 3D models. *Algorithmica*, 38(1):201–225, 2003.

[67] M. Kazhdan and T. Funkhouser. Harmonic 3D shape matching. In *SIGGRAPH 2002 Technical Sketch*, pages 191–191, 2002.

[68] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape descriptors. In *SGP '03: Proc. 2003 Eurographics/ACM SIGGRAPH symp. on Geometry processing*, pages 156–164, 2003.

[69] J. Kent, W. Carlson, and R. Parent. Shape transformation for polyhedral objects. In *SIGGRAPH '92*, pages 47–54. ACM Press, 1992.

[70] L. Kharevych, B. Springborn, and P. Schröder. Discrete conformal mappings via circle patterns. *ACM Trans. Graph.*, 25(2):412–438, 2006.

[71] A. Khodakovsky, N. Litke, and P. Schröder. Globally smooth parameterizations with low distortion. *ACM Trans. Graph.*, 22(3):350–357, 2003.

[72] T. Kitagawa. On the numerical stability of the method of fundamental solutions applied to the dirichlet problem. *Japan Journal of Applied Mathematics*, 35:507–518, 1988.

[73] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3D models. *ACM Trans. Graph.*, 23(3):861–869, 2004.

[74] R. Kress and A. Mohsen. On the simulation source technique for exterior problems in acoustics. *Math. Methods Appl. Sci.*, 8:585 – 597, 1986.

[75] V. Kupradze and M. Aleksidze. The method of functional equations for the approximate solution of certain boundary value problems. *USSR Comp. Math. and Math. Phys.*, 4-4:82–126, 1964.

[76] E. Larsen, S. Gottschalk, M. C. Lin, and D. Manocha. Fast distance queries with rectangular swept sphere volumes. In *Proc. IEEE International Conference on Robotics and Automation*, pages 3719–3726, 2000.

[77] F. Lazarus and A. Verroust. Three-dimensional metamorphosis: a survey. *The Visual Computer*, 14(8/9):373–389, 1998.

[78] A. Lee, D. Dobkin, W. Sweldens, and P. Schröder. Multiresolution mesh morphing. In *Proc. SIGGRAPH*, pages 343–350, 1999.

[79] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *SIGGRAPH*, pages 362–371, 2002.

[80] X. Li. Parameterization based shape comparison. *Technical Report, Stony Brook University*, 2005.

[81] X. Li, Y. Bao, X. Guo, M. Jin, X. Gu, and H. Qin. Globally optimal surface mapping for surfaces with arbitrary topology. *IEEE Trans. on Visualization and Computer Graphics*, 14(4):805–819, 2008.

[82] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin. Harmonic volumetric mapping for solid modeling applications. In *Proc. ACM symp. on Solid and physical modeling*, pages 109–120, 2007.

[83] T. Martin, E. Cohen, and R. Kirby. Volumetric parameterization and trivariate b-spline fitting using harmonic functions. In *Proc. ACM Solid and Physical Modeling*, pages 269–280, 2008.

[84] R. Mathon and R. L. Johnston. The approximate solution of elliptic boundary-value problems by fundamental solutions. *SIAM J. Numer. Anal.*, 14(4):638–650, 1977.

[85] T. Michikawa, T. Kanai, M. Fujita, and H. Chiyokura. Multiresolution interpolation meshes. In *Proc. Pacific Graphics*, pages 60–69, 2001.

[86] J. W. Milnor. *Morse Theory*. Princeton Univ. Press, 1963.

[87] J. Monaghan. An introduction to SPH. *Computer Physics Communications*, 48:89–96, 1988.

[88] G. Mori, S. Belongie, and J. Malik. Efficient shape matching using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1832–1837, 2005.

[89] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24(3):471–478, 2005.

[90] M. Müller, R. Keiser, A. Nealen, M. Pauly, M. Gross, and M. Alexa. Point based animation of elastic, plastic and melting objects. In *Proc. ACM SIGGRAPH/EG Symp. Computer Animation*, pages 141–151, 2004.

[91] M. Novotni and R. Klein. 3D Zernike descriptors for content based shape retrieval. In *ACM Symposium on Solid Modeling and Applications*, pages 216–225, 2003.

[92] R. Ohbuchi, T. Otagiri, M. Ibato, and T. Takei. Shape-similarity search of three-dimensional models using parameterized statistics. In *Pacific Conference on Computer Graphics and Applications*, pages 265–275, 2002.

[93] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *SMI '01: Proc. International Conference on Shape Modeling & Applications*, pages 154–166, 2001.

[94] M. Pauly, R. Keiser, B. Adams, P. Dutre, M. Gross, and L. J. Guibas. Meshless animation of fracturing solids. *ACM Trans. Graph.*, 24(3):957–964, 2005.

[95] A. Pentland and S. Sclaroff. Closed-form solutions for physically based shape modeling and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(7):715–729, 1991.

[96] U. Pinkall and K. Polthier. Computing discrete minimal surfaces and their conjugates. In *Experimental Mathematics*, volume 2, pages 15–36, 1993.

[97] A. Poullikkas, A. Karageorghis, and G. Georgiou. The method of fundamental solutions for three-dimensional elastostatics problems. *Computers and Structures*, 80:365 – 370, 2002.

[98] E. Praun and H. Hoppe. Spherical parametrization and remeshing. In *SIGGRAPH '03*, pages 340–349, 2003.

[99] E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterizations. In *Proc. SIGGRAPH*, pages 179–184, 2001.

[100] P. A. Ramachandran. Method of fundamental solutions: singular value decomposition analysis. *Comunications in Numerical Methods in Engineering*, 18(11):789–801, 2002.

[101] N. Ray, W. Li, B. Lévy, A. Sheffer, and P. Alliez. Periodic global parameterization. *ACM Trans. Graph.*, 25(4):1460–1485, 2006.

[102] P. V. Sander, J. Snyder, S. J. Gortler, and H. Hoppe. Texture mapping progressive meshes. In *SIGGRAPH '01*, pages 409–416, 2001.

[103] D. Saupe and D. V. Vranić. 3D model retrieval with spherical harmonics and moments. In *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, pages 392–397, London, UK, 2001. Springer-Verlag.

[104] R. Schoen and S.-T. Yau. On univalent harmonic maps between surfaces. *Inventiones Mathematicae*, 44(3):265–278, 1978.

[105] R. Schoen and S. T. Yau. *Lectures on harmonic maps*. International Press, Cambridge, MA, USA, 1997.

[106] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. *SIGGRAPH.*, 23(3):870–877, 2004.

[107] T. Sebastian, P. Klein, and B. Kimia. On aligning curves. *PAMI*, 25(1):116–125, 2003.

[108] E. Sharon and D. Mumford. 2D-shape analysis using conformal mapping. In *CVPR (2)*, pages 350–357, 2004.

[109] A. Sheffer and E. de Sturler. Parameterization of faceted surfaces for meshing using angle-based flattening. *Eng. Comput. (Lond.)*, 17(3):326–337, 2001.

[110] A. Sheffer, C. Gotsman, and N. Dyn. Robust spherical parameterization of triangular meshes. *Computing*, 72(1-2):185–193, 2004.

[111] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.

[112] H. Si. http://tetgen.berlios.de/ version 1.41. 2006.

[113] K. Stephenson. *Introduction to circle packing*. Cambridge University Press, 2005.

[114] R. W. Sumner and J. Popovic. Deformation transfer for triangle meshes. *ACM Trans. Graph.*, 23(3):399–405, 2004.

[115] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *Shape Modeling International*, pages 130–139, 2003.

[116] V. Surazhsky, P. Alliez, and C. Gotsman. Isotropic remeshing of surfaces: a local parameterization approach. In *Proceedings of 12th International Meshing Roundtable*, 2003.

[117] H. D. Tagare. Shape-based nonrigid correspondence with application to heart motion analysis. In *IEEE Trans. Medical Imaging*, volume 18, pages 570–578, 1999.

[118] J. Tangelder and R. Veltkamp. A survey of content based 3D shape retrieval methods. In *SMI '04: Proc. of the Shape Modeling International 2004*, pages 145–156, 2004.

[119] R. Tanklevich, G. Fairweather, A. karageorghis, and Y. S. Smyrlis. Potential field based geometric modeling using the method of fundamental solutions. *Technical Report*.

[120] M. Tarini, K. Hormann, P. Cignoni, and C. Montani. Polycube-maps. *ACM Trans. Graph.*, 23(3):853–860, 2004.

[121] J. Teran, N. Molino, R. Fedkiw, and R. Bridson. Adaptive physics based tetrahedral mesh generation using level sets. *Engineering with Computers*, 21(1):2–18, 2005.

[122] W. Thurston. *Geometry and topology of 3-manifolds*. Princeton lecture notes, 1976.

[123] W. Tutte. How to draw a graph. In *Proc. London Math. Soc.*, pages 743–767, 1963.

[124] J. Vorsatz, C. Rössl, and H.-P. Seidel. Dynamic remeshing and applications. In *SM '03: Proc. ACM symposium on Solid modeling and applications*, pages 167–175, 2003.

[125] D. Vranić, D. Saupe, and J. Richter. Tools for 3D-object retrieval: Karhunen-loeve transform and spherical harmonics. In *IEEE MMSP 2001*, pages 293–298, 2001.

[126] D. V. Vranić. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *Proc. ICIP.(3)*, pages 757–760, 2003.

[127] H. Wang, Y. He, X. Li, X. Gu, and H. Qin. Polycube splines. In *SPM '07: Proceedings of the 2007 ACM symposium on Solid and physical modeling*, pages 241–251, 2007.

[128] H. Wang, M. Jin, Y. He, X. Gu, and H. Qin. User-controllable polycube map for manifold spline construction. In *Proc. ACM Symp. on Solid and Physical Modeling*, pages 397–404, 2008.

[129] L. Younes. Computable elastic distances between shapes. *SIAM J. Appl. Math.*, 58(2):565–586, 1998.

[130] R. Zayer, C. Rössl, Z. Karni, and H.-P. Seidel. Harmonic guidance for surface deformation. *Computer Graphics Forum*, 24(3):601–609, 2005.

[131] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *CVPR*, pages 2524–2530, 1999.

[132] M. Zöckler, D. Stalling, and H.-C. Hege. Fast and intuitive generation of geometric shape transitions. *Visual Computer*, 16(5):241–253, 2000.

Appendix

Convergence, One-to-one, and Uniqueness of Our Globally Optimal Surface Mapping

In this appendix, we show our surface global optimization algorithm converges to a globally unique one-to-one map with the minimal harmonic energy under the uniformization metric. The pipeline is: (1) we demonstrate the **existence** of the harmonic map between given surfaces S_1 and S_2 with same topology; (2) we show if the final map we get is harmonic, then it is one-to-one, and **globally unique**; (3) we will show our algorithm does **converge** to such a unique mapping with minimal harmonic energy under uniformization metric.

Existence

Given two high genus surfaces S_1 and S_2 with same non-trivial topology. The existence of the harmonic map is guaranteed by the following theorem

Theorem 4 *Suppose that S_1 and S_2 are compact surfaces without boundary and that $h : S_1 \to S_2$ is a diffeomorphism. Then there exists a harmonic diffeomorphism $f : S_1 \to S_2$ isotopic to h. Furthermore, f is of least energy among all diffeomorphisms isotopic to h.*

Detailed proof can be found in [60], page 176. Since our initial map is constructed as a diffeomorphism between S_1 and S_2, the existence of harmonic map is guaranteed.

One-to-one and uniqueness

We show if the final map is harmonic, then it is a **diffeomorphism** (one-to-one and differentiable) and has the global uniqueness. We prove in the third step that we do reach a harmonic map. The following theorem guarantees the harmonic map calculated in our algorithm is a diffeomorphism.

Theorem 5 *Let $f : S_1 \rightarrow S_2$ be a harmonic map between closed oriented surfaces of the same genus with degree equals ± 1. And $K_{S_2} \leq 0$, then f is a diffeomorphism.*

Detailed proof can be found in [60], page 187, or [105], page 15. In our algorithm, the initial map is constructed by matching the fundamental polygons of S_1 and S_2. Therefore, each point on S_2 has a unique pre-image on S_1, hence, the degree of the initial map is 1. The Gaussian curvature of the target surface is 0 (for genus-1 surfaces) or -1 (for high genus surfaces), therefore if f is harmonic, then f is one-to-one and differentiable.

The following theorem postulates the uniqueness of the map.

Corollary 6 *Let u_1, u_2 be harmonic maps $M \rightarrow N$ of degree one between compact surfaces without boundaries, with genus greater than one, where $K_N \equiv -1$. If u_1 and u_2 are homotopic to each other, then $u_1 = u_2$.*

The detailed proof can be found in [46], page 144 and [105], page 16. In our algorithm, the homotopy class of the map is determined by the way to match the fundamental polygons. The map is harmonic, the curvature on the target surface is -1, therefore, the harmonic map is unique.

For genus-one surfaces, their uniformization metric is flat, which can be lifted to its universal covering space. The universal cover can be embedded on the plane isometrically. The fundamental polygons are parallelograms. A harmonic map between two genus-one surfaces with their flat uniformization metrics induces a map between their universal covering spaces, which is an affine transformation from the plane to itself. The affine transformation maps the fundamental polygon of the source surface to that of the target surface. Therefore, harmonic maps in a homotopy class only differ by a translation. Each one is the equally optimal result.

Convergence

We prove our algorithm converge to a harmonic map. Harmonic energy of a surface map is non-negative, namely, it has lower bound. Our relaxation process reduces harmonic energy monotonically; therefore, it converges to a critical point of the harmonic energy, which by definition is a harmonic map. As the aforementioned theorems show, there is no local minimum, and this critical point is globally unique. Therefore, our method converges to the global unique harmonic map, and it is one-to-one and differentiable.

For genus-one surfaces, this convergence proof also applies, and all the minima are globally equal and globally optimal. Our minimization process will converge to one of them.